32-bit RISC Microcontroller
TX03 Series

TMPM370FYDFG
TMPM370FYFG

Revision 1.1

2023-07

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

©2026
Toshiba Electronic Devices & Storage Corporation

Contents

Contents 2

List of figures 15

List of tables 19

1. Features 26

1.1. Products Lists Categorized by Functions 30

1.2. Block Diagram 31

1.3. Pin Layout (top view) 32

1.3.1. QFP100 32

1.3.2. LQFP100 33

1.4. Pin Name and Functions 34

1.4.1. Functional Pin and Ports Assignment (Pin Number) 34

1.5. Power Supply Type and Supply Pins 39

2. Processor Core 40

2.1. Information on the processor core 40

2.2. Configurable Options 41

2.3. Exceptions / Interrups 41

2.3.1. Number of Interrupt Inputs 41

2.3.2. Number of Priority Level Interrupt Bits 41

2.3.3. SysTick 42

2.3.4. SYSRESETREQ 42

2.3.5. LOCKUP 42

2.3.6. Auxiliary Fault Status register 42

2.4. Events 42

2.5. Power Management 43

2.6. Exclusive access 43

3. Memory Map 44

3.1. Memory Map 44

3.1.1. Memory Map 45

3.2. Details of SFR area 46

4. RESET Operation 47

4.1. State before RESET 47

4.2. State during RESET 47

4.3. State after releasing RESET 47

5. Clock/Mode Control(CG) 48

5.1. Features 48

5.2. Registers 49

5.2.1. Register List 49

5.2.2. CGSYSCR (System control register) 50

5.2.3. CGOSCCR (Oscillation control register) 51

5.2.4. CGSTBYCR (Standby control register) 52

5.2.5. CGPLLSEL (PLL Selection Register) 53

5.3. Clock control 54

5.3.1. Clock Type 54

5.3.2. Initial Values after Reset 54

5.3.3. Clock system Diagram 55

5.3.4. Clock Multiplication Circuit (PLL) 56

5.3.5. Warming-up function 57

5.3.6. System Clock 58

5.3.7. Prescaler Clock Control 59

5.4. Modes and Mode Transitions 59

5.4.1. Mode Transitions 59

5.5. Operation Mode 60

5.5.1. NORMAL mode 60

5.6. Low Power Consumption Modes 60

5.6.1. IDLE Mode 60

5.6.2. STOP mode 61

5.6.3. Low power Consumption Mode Setting 62

5.6.4. Operational Status in Each Mode 63

5.6.5. Releasing the Low Power Consumption Mode 64

5.6.6. Warming-up 66

5.6.7. Clock Operation in Mode Transition 67

6. Exceptions 68

6.1. Outline 68
6.1.1. Types 68
6.1.2. Handling Flowchart 69
6.2. RESET Exceptions 76
6.3. SysTick 76
6.4. Interrupts 77
6.4.1. Non-Maskable Interrupt (NMI) 77
6.4.2. Maskable Interrupt 77
6.4.3. Interrupt Request 77
6.4.4. List of Interrupt Sources for Each Product 81
6.4.5. Interrupt Detection Level 84
6.4.6. Handling Method 85
6.5. Exception/Interrupt-Related Registers 92
6.5.1. Register List 92
6.5.2. NVIC Registers 94
6.5.3. CG Registers 121
7. Product Information 133
7.1. Information of Each Peripheral Function 133
7.1.1. Clock/Mode Control (CG) 133
7.1.2. Exception 134
7.1.3. Input/output Port 135
7.1.4. 16-bit Timer/event Counter (TMRB) 136
7.1.5. Serial Channel (SIO/UART) 140
7.1.6. 12-bit analog-to-digital converter (ADC) 142
7.1.7. Motor Control Circuit (PMD) 145
7.1.8. Vector Engine (VE) 149
7.1.9. Encoder Input Circuit (ENC) 150
7.1.10. Voltage Detection circuit (VLTD) 152
7.1.11. Frequency detection circuit (OFD) 153
7.1.12. Watchdog Timer (WDT) 154
7.1.13. Op-amplifiers/Analog Comparators (AMP/CMP) 155
7.1.14. Flash memory 157
7.1.15. Protect/Security Function 159
7.1.16. Debug Interface 160
8. Input/output Port 161
8.1. Outlines 161

8.2. Signal connection list 162

8.3. Registers 167

8.3.1. List of Register 168

8.3.2. List of Port Functions and Settings 169

8.4. Block Diagrams of Ports 181

8.4.1. Type T1 182

8.4.2. Type T2 183

8.4.3. Type T3 184

8.4.4. Type T4 185

8.4.5. Type T5 186

8.4.6. Type T6 187

8.4.7. Type T7 188

8.4.8. Type T8 189

8.4.9. Type T9 190

8.4.10. Type T10 191

8.4.11. Type T11 192

8.4.12. Type T12 193

8.4.13. Type T13 194

8.4.14. Type T14 195

8.4.15. Type T15 196

8.4.16. Type T16 197

8.4.17. Type T17 198

8.4.18. Type T18 199

8.4.19. Type T19 200

8.4.20. Type T20 201

8.4.21. Type T21 202

8.5. Precaution 203

8.5.1. Pin status during a reset 203

8.5.2. Unused pins 203

8.5.3. Notice of using debug interface pins used as general-purpose ports 203

9. 16-bit Timer/Event Counter (TMRB) 204

9.1. Outline 204

9.2. Block Diagram 205

9.3. Registers 206

9.3.1. List of Registers 206

9.3.2. TBxEN (Enable Register) 207

9.3.3. TBxRUN (RUN Register) 208

9.3.4. TBxCR (Control Register) 209
9.3.5. TBxMOD (Mode Register) 210
9.3.6. TBxFFCR (Flip-Flop Control Register) 211
9.3.7. TBxST (Status Register) 212
9.3.8. TBxIM (Interrupt Mask Register) 213
9.3.9. TBxUC (Up Counter Capture Register) 214
9.3.10. TBxRG0 (Timer Register 0) 215
9.3.11. TBxRG1 (Timer Register 1) 215
9.3.12. TBxCP0 (Capture Register 0) 216
9.3.13. TBxCP1 (Capture Register 1) 216

9.4. Operations 217
9.4.1. Prescaler 217
9.4.2. Up Counter (UC) 219
9.4.3. Timer Register (TBxRG0, TBxRG1) 219
9.4.4. Capture Control 220
9.4.5. Capture Register (TBxCP0, TBxCP1) 220
9.4.6. Up Counter Capture Register (TBxUC) 220
9.4.7. Comparator (CP0, CP1) 220
9.4.8. Timer Flip-Flop (TBxFF0) 220
9.4.9. Capture Interrupt (INTCAPx0, INTCAPx1) 220

9.5. Explanation of Each Mode 221
9.5.1. 16-bit Interval Timer Mode 221
9.5.2. 16-bit Event Counter Mode 221
9.5.3. 16-bit Programmable Square Wave Output (PPG) Mode 222
9.5.4. External Trigger Programmable Square Wave (PPG) Output Mode 224
9.5.5. Applications Using the Capture Function 225

10. Serial Channel (SIO/UART) 228

10.1. Outline 228

10.2. Block Diagram 229

10.3. Registers 230
10.3.1. Register List 230
10.3.2. SCxEN (Enable Register) 231
10.3.3. SCxBUF (Buffer Register) 232
10.3.4. SCxCR (Control Register) 233
10.3.5. SCxMOD0 (Mode Control Register 0) 234
10.3.6. SCxBRCR (Baud Rate Generator Control Register) 235
10.3.7. SCxBRADD (Baud Rate Generator Control Register 2) 236

10.3.8. SCxMOD1 (Mode Control Register 1) 237
10.3.9. SCxMOD2 (Mode Control Register 2) 238
10.3.10. SCxRFC (Reception FIFO Configuration Register) 240
10.3.11. SCxTFC (Transmission FIFO Configuration Register) (Note2) 241
10.3.12. SCxRST (Reception FIFO Status Register) 242
10.3.13. SCxTST (Transmission FIFO Status Register) 243
10.3.14. SCxFCNF(FIFO Configuration Register) 244

10.4. Mode 246

10.5. Data Format 247
10.5.1. Data Format List 247
10.5.2. Parity Bit Control 247
10.5.3. STOP Bit Length 248

10.6. Clock Control 249
10.6.1. Prescaler 249
10.6.2. Serial Clock Generation Circuit 252

10.7. Transmission/Reception Buffers and FIFO 260
10.7.1. Block Diagram 260
10.7.2. Transmission/Reception Buffers 260
10.7.3. FIFO 261

10.8. Status Flag 261

10.9. Error Flags 262
10.9.1. OERR Flag 262
10.9.2. PERR Flag 263
10.9.3. FERR Flag 263

10.10. Reception 264
10.10.1. Reception Counter 264
10.10.2. Reception Control 264
10.10.3. Reception Operation 265

10.11. Transmission 268
10.11.1. Transmission Counter 268
10.11.2. Transmission Control 268
10.11.3. Transmission Operation 269

10.12. Handshake Function 272

10.13. Interrupt/Error Generation Timing 274
10.13.1. Reception Interrupt (INTRXx) 274

10.13.2. Transmission Interrupt (INTTXx) 275
10.13.3. Interrupt Generation Timing When Errors Occur 277
10.14. Software Reset 277
10.15. Explanation of Operation for Each Mode 278
10.15.1. Mode 0 (SIO Mode) 278
10.15.2. Mode 1 (7-bit UART Mode) 289
10.15.3. Mode 2 (8-bit UART Mode) 290
10.15.4. Mode3 (9-bit UART mode) 291
11. 12-Bit Analog-to-Digital Converter (ADC) 293
11.1. Functions and features 293
11.2. Block Diagram 294
11.3. List of Registers 295
11.4. Register Descriptions 297
11.4.1. ADxCLK (Clock Setting Register) 297
11.4.2. ADxMOD0 (Mode Setting Register 0) 298
11.4.3. ADxMOD1 (Mode Setting Register 1) 299
11.4.4. ADxMOD2 (Mode Setting Register 2) 300
11.4.5. ADxCMPCR0 (Monitoring Setting Register 0) 301
11.4.6. ADxCMPCR1 (Monitoring Setting Register 1) 302
11.4.7. ADxCMP0 (Conversion Result Compare Register 0) 303
11.4.8. ADxCMP1 (Conversion Result Compare Register 1) 303
11.4.9. ADxREG0 (Conversion Result Register 0) 304
11.4.10. ADxREG1 (Conversion Result Register 1) 305
11.4.11. ADxREG2 (Conversion Result Register 2) 306
11.4.12. ADxREG3 (Conversion Result Register 3) 307
11.4.13. ADxREG4 (Conversion Result Register 4) 308
11.4.14. ADxREG5 (Conversion Result Register 5) 309
11.4.15. ADxREG6 (Conversion Result Register 6) 310
11.4.16. ADxREG7 (Conversion Result Register 7) 311
11.4.17. ADxREG8 (Conversion Result Register 8) 312
11.4.18. ADxREG9 (Conversion Result Register 9) 313
11.4.19. ADxREG10 (Conversion Result Register 10) 314
11.4.20. ADxREG11 (Conversion Result Register 11) 315
11.4.21. PMD Trigger Program Registers 316
11.4.22. ADxTSET03/ADxTSET47/ADxTSET811 (Timer Trigger Program Registers) 337
11.4.23. ADxSSET03/ADxSSET47/ADxSSET811 (Software Trigger Program Registers) 341
11.4.24. ADxASET03/ADxASET47/ADxASET811 (Constant Conversion Program Registers) 345

11.4.25. ADxMOD3 (Mode Setting Register 3) 349

11.5. Operation Descriptions 350
11.5.1. Analog Reference Voltage 350
11.5.2. Starting AD Conversion 350
11.5.3. AD Conversion Monitoring Function 352

11.6. Timing chart of AD conversion 352
11.6.1. Software trigger Conversion 352
11.6.2. Constant Conversion 354
11.6.3. AD Conversion by Trigger 355

11.7. Usage Example 356
11.7.1. Example When Using PMD ch0 (3 shunts) and one AD converter (unit A) 356
11.7.2. Example When Using PMD ch0 (3 shunts) and two AD converters (unit A and B) 357
11.7.3. Example When Using PMD ch0 (3 shunts), ch1 (1 shunt) and two AD converters (unit A and B) 358
11.7.4. Example When Using PMD ch0 (1 shunt) and one AD converter (unit A) 360

12. Motor Control Circuit (PMD) 361

12.1. Outline 361

12.2. Block Diagram 362

12.3. Registers 363
12.3.1. List of Registers 363
12.3.2. PMD Control 364
12.3.3. Pulse Width Modulation circuit 367
12.3.4. Conduction Control Circuit 378
12.3.5. Protection Control Circuit 382
12.3.6. OVV protection control 388
12.3.7. Protection Control When Break Occurs in Debugger 393
12.3.8. Dead time circuit 394
12.3.9. Synchronous Trigger Generation Circuit 396

13. Vector Engine (VE) 404

13.1. Outline 404
13.1.1. Features 404
13.1.2. Main Functions 405

13.2. Block Diagram 405
13.2.1. VE, PMD, and ADC Relationships 406

13.3. Registers 408
13.3.1. List of Registers 408

13.3.2. VE Control Register 411
13.3.3. Common Register 422
13.3.4. Specific Registers 423
13.4. Operations 453
13.4.1. Schedule Management 453
13.4.2. Task Overview 458
13.5. VE and AD Conversion Result Register 470
14. Encoder Input Circuit (ENC) 471
14.1. Outline 471
14.2. Block Diagram 471
14.3. Registers 472
14.3.1. List of Registers 472
14.3.2. ENxTNCR (Encoder Input Control Register) 472
14.3.3. ENxRELOAD (Encoder Counter Reload Register) 476
14.3.4. ENxINT (Encoder Compare Register) 477
14.3.5. ENxCNT (Encoder Counter Register) 478
14.4. Operational Description 480
14.4.1. Encoder Mode 480
14.4.2. Sensor Mode 480
14.4.3. Timer Mode 481
14.5. Function 481
14.5.1. Mode Operation Outline 481
14.5.2. Counter and interrupt generate operation when <CMPEN> = "1" 489
14.5.3. Counter and interrupt generate operation when <CMPEN> = "0" 491
14.5.4. Encoder Rotation Direction 493
14.5.5. Counter Circuit 494
14.5.6. Interrupt 495
15. Power-on Reset circuit (POR) 497
15.1. Block Diagram 497
15.2. Function 498
16. Voltage Detection circuit (VLTD) 501
16.1. Outline 501
16.2. Block Diagram 501

16.3. Registers 502
16.3.1. List of Registers 502
16.3.2. VDCR (Voltage Detection Control Register) 502
16.4. Operations 503
16.4.1. Control 503
16.4.2. Function 503
17. Oscillation Frequency Detector Circuit (OFD) 505
17.1. Configuration 505
17.2. Registers 507
17.2.1. Registers List 507
17.2.2. OFDCR1 (Oscillation frequency detection control register 1) 507
17.2.3. OFDCR2(Oscillation frequency detection control register 2) 508
17.2.4. OFDMNPLLOFF (Lower detection frequency setting register (PLL OFF)) 509
17.2.5. OFDMNPLLON (Higher detection frequency setting register (PLL ON)) 510
17.2.6. OFDMXPLLOFF (Higher detection frequency setting register (PLL OFF)) 511
17.2.7. OFDMXPLLON (Higher detection frequency setting register (PLL ON)) 512
17.3. Function 513
17.3.1. Operation Control of Oscillation Frequency Detection Circuit 513
17.3.2. Setting of Detection Frequency 514
17.3.3. Oscillation Frequency Detection Reset 515
18. Watchdog Timer (WDT) 517
18.1. Outline 517
18.2. Block Diagram 517
18.3. Registers 518
18.3.1. List of Registers 518
18.3.2. WDMOD (Watchdog Timer Mode Register) 519
18.3.3. WDCR (Watchdog Timer Control Register) 520
18.4. Operations 521
18.4.1. Basic Operation 521
18.4.2. Operating Mode and State 521
18.5. Operation when Runaway is Detected 522
18.5.1. INTWDT Generation 522
18.5.2. Watchdog Timer Reset Generation 523
18.6. Control register 524

18.6.1. Watchdog Timer Mode Register (WDMOD) 524
18.6.2. Watchdog Timer Control Register (WDCR) 524
18.6.3. Setting Example 525

19. Op-amplifiers/Analog Comparators (AMP/CMP) 526

19.1. Block Diagram 527

19.2. Registers 528
19.2.1. List of Registers 528
19.2.2. Op-amplifiers 528
19.2.3. Comparators 529

19.3. Operations 530

20. Flash Operations 531

20.1. Flash Memory 531
20.1.1. Features 531
20.1.2. Flash Block Diagram 533

20.2. Operation Mode 534
20.2.1. Reset Operation 535
20.2.2. User Boot Mode (Single Chip Mode) 536
20.2.3. Single Boot Mode 545
20.2.4. Mode Setting 549
20.2.5. Memory Map 550
20.2.6. Interface Specifications 551
20.2.7. Data Transfer Format 552
20.2.8. Constraints of Internal Memories 552
20.2.9. Transfer Format of the Boot Program 552
20.2.10. Operation Description of Boot Program 556
20.2.11. Flowchart of Entire Boot Program 570

20.3. Flash Memory Write/Erase in Onboard Programming 571
20.3.1. Flash Memory 571
20.3.2. Address Bit Configuration during Bus Write Cycle 581

21. Protect/Security Function 586

21.1. Outline 586

21.2. Features 586
21.2.1. Writing/erasing protect of internal ROM (Flash) 586
21.2.2. Security function 586

21.3. Registers 588
21.3.1. Registers List 588
21.3.2. FCSECBIT (Security bit register) 589
21.3.3. FCFLCS (Flash control register) 590
21.4. Setting/Releasing Method 591
21.4.1. Writing/erasing protect of internal ROM (Flash) 591
21.4.2. Security bit 591
22. Debug Interface 592
22.1. Outline 592
22.1.1. SWJ-DP 592
22.1.2. ETM 592
22.2. Pin Functions 593
22.3. Connection With Debug Tool 595
22.3.1. Connection 595
22.3.2. Notice When Debug Interface Pins are Used As General-purpose Port 595
22.3.3. Peripheral Functions Operation During HALT Mode (Stop Execution of Program In Debugging) 595
23. Port Section Equivalent Circuit Schematic 596
23.1. PA0 to 7, PB0 to 7, PC0 to 7, PD0 to 6, PE0 to 7, PF0 to 4, PG0 to 7 596
23.2. PH0 to 7, PI0 to 3, PJ0 to 7, PK0, PK1 597
23.3. PL0, PL1 598
23.4. PF0 598
23.5. X1, X2 599
23.6. RESET 599
23.7. MODE 599
23.8. VREFHA, VREFHB, VREFLA, VREFLB 600
23.9. AMPVDD5, AMPVSS, CREFABC, CREFD 600
23.10. VOUT15, VOUT3 601
24. Electrical Characteristics 602
24.1. Absolute Maximum Ratings 602
24.2. DC Electrical Characteristics (1/2) 603

24.3. DC Electrical Characteristics (2/2) (Current Consumption) 604
24.4. 12-bit analog-to-digital Converter Characteristics 605
24.5. Op-amplifiers Electrical Characteristics 606
24.6. Analog Comparators Electrical Characteristics 606
24.7. AC Electrical Characteristics 607
24.7.1. AC Conditions for measurement 607
24.7.2. Serial Channel (SIO/UART) 608
24.7.3. 16-bit Timer/Event Counter (TMRB) 610
24.7.4. External Interrupt Pin 611
24.7.5. Debug communication 612
24.8. Flash Characteristic 614
24.9. Sample Oscillation Circuit 614
24.10. Ceramic resonator 615
24.11. Notice When Power ON 616
24.11.1. Port L (PL0 and PL1) State When Supply Voltage Rises 616
24.11.2. When the external reset is not used 617
24.11.3. When using an external reset 618
25. Package Dimensions 621
25.1. Type: QFP100-P-1420-0.65Q 621
25.2. Type: LQFP100-P-1414-0.50H 622
26. Revision History 623
RESTRICTIONS ON PRODUCT USE 624

List of figures

Figure 1.1 Block diagram....31
Figure 3.1 Memory Map....45
Figure 5.1 Clock Block Diagram....55
Figure 5.2 PLL setting sequence after reset....56
Figure 5.3 Mode Transition Diagram....59
Figure 5.4 Transition of operation modes: NORMAL → STOP → NORMAL....67
Figure 6.1 Handling Flowchart....69
Figure 6.2 Stack After Completion of Stack Push....73
Figure 6.3 Interrupt Request Route....78
Figure 6.4 Handling Flowchart....85
Figure 8.1 Port Type T1....182
Figure 8.2 Port Type T2....183
Figure 8.3 Port Type T3....184
Figure 8.4 Port Type T4....185
Figure 8.5 Port Type T5....186
Figure 8.6 Port Type T6....187
Figure 8.7 Port Type T7....188
Figure 8.8 Port Type T8....189
Figure 8.9 Port Type T9....190
Figure 8.10 Port Type T10....191
Figure 8.11 Port Type T11....192
Figure 8.12 Port Type T12....193
Figure 8.13 Port Type T13....194
Figure 8.14 Port Type T14....195
Figure 8.15 Port Type T15....196
Figure 8.16 Port Type T16....197
Figure 8.17 Port Type T17....198
Figure 8.18 Port Type T18....199
Figure 8.19 Port Type T19....200
Figure 8.20 Port Type T20....201
Figure 8.21 Port Type T21....202
Figure 9.1 TMRB Block Diagram....205
Figure 9.2 Output Waveforms in 16-bit Programmable Square Wave Output (PPG) Mode....222
Figure 9.3 Register Buffer Operation....222
Figure 9.4 Output Circuit in 16-bit Programmable Square Wave Output (PPG) Mode....223
Figure 9.5 Square Wave Output in External Trigger Programmable Square Wave (PPG) Output Mode....224
Figure 9.6 Square Wave Output from External Signal Using Free-running Counter (with Delay)....225
Figure 9.7 Measuring the "High" Level-Width of TBxIN Pin Input Pulses....227

Figure 10.1 SIO/UART Block Diagram....229
Figure 10.2 Data Format....247
Figure 10.3 Configuration of Transmission/Reception Buffers and FIFO....260
Figure 10.4 Reception Buffer Operation....265
Figure 10.5 Reception FIFO Operation....266
Figure 10.6 Generation of Transmission Clock in UART Mode....268
Figure 10.7 Transmission Buffer Operation....269
Figure 10.8 Transmission FIFO Operation....270
Figure 10.9 Handshake Function Connection....272
Figure 10.10 CTS Signal Timing....273
Figure 10.11 Reception Operation Data Flow and Read Path....274
Figure 10.12 Transmission Operation Data Flow and Write Path....275
Figure 10.13 SIO Mode Reception Operation (When SCLKx Pin Output is Used for Transfer Clock)....279
Figure 10.14 SIO Mode Reception Operation (When SCLKx Pin Input is Used as Transfer Clock)....280
Figure 10.15 SIO Mode Transmission Operation (When SCLKx Pin Output is Used for Transfer Clock)....282
Figure 10.16 SIO Mode Transmission Operation (When SCLKx Pin Input is Used for Transfer Clock)....284
Figure 10.17 SIO Mode Transmission/Reception Operation (When SCLKx Pin Output is Used for Transfer Clock)....286
Figure 10.18 SIO Mode Transmission/Reception Operation (When SCLKx Pin Input is Used for Transfer Clock)....288
Figure 10.19 Mode 1 (7-bit UART Mode)....289
Figure 10.20 Mode 2 (8-bit UART Mode)....290
Figure 10.21 Serial Link with Wake-up Function....291
Figure 10.22 1st Frame Format for Wake-up Function....292
Figure 10.23 2nd Frame Format for Wake-up Function....292
Figure 11.1 AD converter Block Diagram....294
Figure 11.2 PMD Trigger Program Registers....317
Figure 11.3 Timing Chart of AD Conversion....351
Figure 11.4 Software Trigger AD Conversion Timing Chart....353
Figure 11.5 Writing "0" to <ADEN> during Software Trigger AD Conversion....353
Figure 11.6 Constant Conversion Timing Chart....354
Figure 11.7 AD Conversion by PMD Trigger During AD conversion....355
Figure 11.8 AD Conversion by Timer Trigger During AD conversion....355
Figure 11.9 Example Circuit When Using PMD Ch0 (3 Shunts) and One AD Converter (Unit A)....356
Figure 11.10 Example Circuit When Using PMD ch0 (3 shunts) and two AD converters (unit A and B)....357
Figure 11.11 Example Circuit When Using PMD ch0 (3 shunts), ch1 (1 shunt) and two AD converters (unit A and B)....358
Figure 11.12 Example Circuit When Using PMD ch0 (1 shunt) and one AD converter (unit A)....360
Figure 12.1 In Conjunction with PMD, ADC, AMP/CMP and VE....361
Figure 12.2 PMD Block Diagram....362

Figure 12.3 Pulse Width Modulation Circuit 367
Figure 12.4 PWM Waveforms of Sawtooth and Triangle Waveform 368
Figure 12.5 Conduction Control Circuit 378
Figure 12.6 Protection Control Circuit 382
Figure 12.7 Dead Time Circuit 394
Figure 12.8 Synchronous Trigger Generation Circuit 398
Figure 13.1 Block Diagram of Vector Control 404
Figure 13.2 Configuration of VE 405
Figure 13.3 Interaction among VE, PMD and ADC 407
Figure 13.4 Motor Control Operation Flow Example 453
Figure 13.5 Operation Transition in Operation Schedule 456
Figure 14.1 Block diagram of encoder input circuit 471
Figure 15.1 Block Diagram of POR 497
Figure 15.2 Operation Timing of Power-on reset (1/2) 499
Figure 15.3 Operation Timing of Power-on reset (2/2) 499
Figure 16.1 Voltage Detection Circuit 501
Figure 16.2 Voltage detection timing 504
Figure 17.1 Initial Value of Detection Frequency Range (PLL OFF) 506
Figure 17.2 Initial Value of Detection Frequency Range (PLL ON) 506
Figure 17.3 Oscillation Frequency Detector Circuit 506
Figure 17.4 Operation of Oscillation Frequency Detection Circuit in each Mode 514
Figure 17.5 Oscillation Frequency Detection Reset Timing 516
Figure 18.1 Block Diagram of WDT 517
Figure 18.2 Timing of INTWDT Generation 522
Figure 18.3 Timing of Watchdog Timer Reset Generation 523
Figure 19.1 Op-amplifiers and Analog Comparators Connection Diagram 527
Figure 19.2 Op-amplifier and Analog Comparator Configuration 530
Figure 20.1 Flash Block Diagram 533
Figure 20.2 Mode Transition Diagram 535
Figure 20.3 Memory Map 550
Figure 20.4 Serial Operation Mode Determination Data 565
Figure 20.5 Reception flowchart of Serial Operation Mode 567
Figure 20.6 Flowchart of Serial Operation Mode Determination 568
Figure 20.7 Flowchart of Password Area Check 569
Figure 20.8 Flowchart of Entire Boot program 570
Figure 20.9 Block Configuration 572
Figure 20.10 Automatic Program 584
Figure 20.11 Automatic Erase 585
Figure 24.1 Timing Chart When Transfer Clock is Input on SCLKx Pin 609
Figure 24.2 Timing Chart When Transfer Clock Is Output on SCLKx Pin 609

Figure 24.3 Timing Chart of Event Counter 610
Figure 24.4 Timing Chart of Capture 610
Figure 24.5 Timing Chart When Using External Interrupts Other than the following 611
Figure 24.6 Timing chart When Releasing STOP Mode by External Interrupt Pin 611
Figure 24.7 Timing Chart of JTAG/SWD 612
Figure 24.8 Timing Chart of ETM trace 613
Figure 24.9 Example of high-frequency oscillation circuit 614
Figure 24.10 Power-on sequence (without external reset) 617
Figure 24.11 Timing chart when releasing external reset is earlier than releasing POR reset 618
Figure 24.12 Timing chart when releasing external reset is slower than releasing POR reset 619
Figure 24.13 Timing chart when t_{DVDD} is longer than t_{PWUP} 620

List of tables

Table 1.1 Products Lists 30
Table 1.2 Signal connection List: SIO/UART 34
Table 1.3 Signal connection List: TMRB 35
Table 1.4 Signal connection List: ADC/AMP/CMP 36
Table 1.5 Signal connection List: INT/PMD 37
Table 1.6 Signal connection List: ENC/DEBUG/Control Pin/Power Supply 38
Table 1.7 Power supply type and supply pins 39
Table 2.1 Core Revision 40
Table 2.2 Configurable options and their implementations 41
Table 3.1 Details of SFR 46
Table 5.1 Range of high-speed frequency (unit: MHz) 58
Table 5.2 Pin States in the STOP mode 61
Table 5.3 Low-power consumption mode setting 62
Table 5.4 Operational Status in Each Mode 63
Table 5.5 Release Source in Each Mode 64
Table 5.6 Warming-up setting in mode transition 66
Table 6.1 Exception Types and Priority 71
Table 6.2 Priority Grouping Setting 72
Table 6.3 Configuration of Vector Table 74
Table 6.4 List of Interrupt Source (Non-maskable Interrupt) 81
Table 6.5 List of Interrupt Source 81
Table 6.6 Setting Interrupt Mask Register 86
Table 6.7 Setting NVIC registers 87
Table 6.8 Setting for External Interrupt Pin 87
Table 6.9 Setting for Interrupt Set-Pending Register 88
Table 6.10 Setting for CG 88
Table 6.11 Setting Enabling CPU's Interrupt 89
Table 6.12 Address of Interrupt Priority Register 113
Table 6.13 Configuration of Interrupt Priority Register for Interrupt Numbers 0 to 3 114
Table 6.14 Address of System Handler Priority Register 117
Table 6.15 Configuration of System Handler Priority Register 118
Table 7.1 Built-in List 133
Table 7.2 Base Address 133
Table 7.3 Built-in List 134
Table 7.4 Base Address 134
Table 7.5 Built-in List (1/2) 135
Table 7.6 Built-in List (2/2) 135
Table 7.7 Base Address (1/2) 135

Table 7.8 Base Address (2/2) 135
Table 7.9 Built-in List 136
Table 7.10 Base Address (1/2) 136
Table 7.11 Base Address (2/2) 136
Table 7.12 Function pins and ports 137
Table 7.13 Connection with Input/output Signals and Peripheral Functions (1/2) 138
Table 7.14 Connection with Input/output Signals and Peripheral Functions (2/2) 139
Table 7.15 Built-in List 140
Table 7.16 Base Address 140
Table 7.17 Function pins and ports 141
Table 7.18 Connection with Input/output Signals and Peripheral Functions 141
Table 7. 19 Built-in List 142
Table 7.20 Base Address 142
Table 7.21 Function pins and ports (1/2) 143
Table 7.22 Function pins and ports (2/2) 143
Table 7.23 Connection with Input/output Signals and Peripheral Functions 144
Table 7.24 Built-in List 145
Table 7.25 Base Address 145
Table 7.26 Function pins and ports 146
Table 7.27 Function pins and ports 146
Table 7.28 Connection with Input Signals and Peripheral Functions 147
Table 7.29 Connection with Input Signals and Peripheral Functions 147
Table 7.30 Connection with Output Signals and Peripheral Functions 148
Table 7.31 Connection with Output Signals and Peripheral Functions 148
Table 7.32 Built-in List 149
Table 7.33 Base Address 149
Table 7.34 Built-in List 150
Table 7.35 Base Address 150
Table 7.36 Function pins and ports 151
Table 7.37 Built-in List 152
Table 7.38 Base Address 152
Table 7.39 Built-in List 153
Table 7.40 Base Address 153
Table 7.41 Built-in List 154
Table 7.42 Base Address 154
Table 7.43 Function pins and ports 154
Table 7.44 Built-in List 155
Table 7.45 Base Address 155
Table 7.46 AMP Function Pins and Ports 156
Table 7.47 CMP Function Pins and Ports 156

Table 7.48 Built-in List 157
Table 7.49 Base Address 157
Table 7.50 Flash Memory Block Configurations 157
Table 7.51 Macro code with ID-Read 158
Table 7.52 Used Resources in Single Boot Mode 158
Table 7.53 RAM Address Range 158
Table 7.54 Built-in List 159
Table 7.55 Base Address 159
Table 7.56 Debug Interface Pin List 160
Table 8.1 Signal connection list: SIO/UART 162
Table 8.2 Signal connection list: TMRB 163
Table 8.3 Signal connection list: ADC/AMP,CMP 164
Table 8.4 Signal connection list: INT/PMD 165
Table 8.5 Signal connection list: ENC/DEBUG/FLASH 165
Table 8.6 Register List (1/2) 168
Table 8.7 Register List (2/2) 168
Table 8.8 Port A registers setting 170
Table 8.9 Port B registers setting 171
Table 8.10 Port C registers setting 172
Table 8.11 Port D registers setting 173
Table 8.12 Port E registers setting 174
Table 8.13 Port F registers setting 175
Table 8.14 Port G registers setting 176
Table 8.15 Port H registers setting 177
Table 8.16 Port I registers setting 178
Table 8.17 Port J registers setting 179
Table 8.18 Port K registers setting 180
Table 8.19 Port L registers setting 180
Table 9.1 Resolution of Output Clock from Prescaler (fc = 80MHz) 217
Table 10.1 Division Value Setting 236
Table 10.2 Modes and Data Formats 246
Table 10.3 Resolution of Output Clock from Prescaler (fc = 80MHz) (1/2) 250
Table 10.4 Resolution of Output Clock from Prescaler (fc = 80MHz) (2/2) 251
Table 10.5 Settable Division Values 252
Table 10.6 Register Settings for Transfer Clock that Can be Used in SIO Mode 253
Table 10.7 Register Settings for SIOCLK Clock which Can be Used in UART Mode 255
Table 10.8 Example of Baud Rate in UART Mode. (Unit: kbps) 257
Table 10.9 Example Baud Rate in UART Mode (Unit: kbps) 258
Table 10.10 Serial Transfer Mode and Buffer Configuration 260
Table 10.11 Transfer Mode and FIFO Configuration 261

Table 10.12 Error Flags 262
Table 10.13 Buffer Configuration and INTRXx Generation Timing 274
Table 10.14 Conditions for INTRXx Generation when FIFO is Enabled 275
Table 10.15 Buffer Configuration and INTTXx Generation Timing 276
Table 10.16 Conditions for INTTXx Generation when FIFO is Enabled 276
Table 10.17 Timing of Error Interrupt Generation in SIO Mode 277
Table 10.18 Error Interrupt Generation Timing in UART Mode 277
Table 11.1 Program number select 324
Table 11.2 Select the AIN pin 330
Table 11.3 Select the AIN pin 337
Table 11.4 Select the AIN pin 341
Table 11.5 Select the AIN pin 345
Table 11.6 AD conversion time and delay time (SCLK = 40MHz) 351
Table 11.7 Example of AD Converter Unit A Settings 356
Table 11.8 Example of AD Converter Unit A and B Settings 357
Table 11.9 Example of AD Converter Unit A Settings 359
Table 11.10 Example of AD Converter Unit B Settings 359
Table 11.11 Example of AD Converter Unit A Settings 360
Table 12.1 Conduction Control Circuit Output by PMDxMDOUT (When PMDxMDPOT<POLH><POLL> = "00") 379
Table 12.2 Conduction Control Circuit Output by PMDxMDOUT (When PMDxMDPOT<POLH><POLL> = "11") 379
Table 12.3 One Shunt Current Detection Settings 379
Table 12.4 Timing of Updating Subsequent Stage of Trigger Compare Register 396
Table 12.5 Trigger Output 397
Table 13.1 VEOUTCRx<UPWM>, VEOUTCRx<UOC[1:0]> PMD setting: Output control of U-phase (UOx, XOx) 449
Table 13.2 VEOUTCRx<VPWM>, VEOUTCRx<VOC[1:0]> PMD setting: Output control of V-phase (VOx, YOx) 449
Table 13.3 VEOUTCRx<WPWM>, VEOUTCRx<WOC[1:0]> PMD setting: Output control of W-phase (WOx, ZOx) 449
Table 13.4 Tasks to be Executed in Each Schedule 455
Table 13.5 Example Setting in Typical Operation Flow 455
Table 13.6 Task Overview 458
Table 13.7 PI Control of d-axis Current 459
Table 13.8 PI Control of q-axis Current 460
Table 13.9 Phase Interpolation 461
Table 13.10 SIN/COS Computation 462
Table 13.11 dq-αβ Coordinate Axis Conversion 463
Table 13.12 2-3 Phase Conversion (Space Vector Conversion) 464

Table 13.13 3 Phase Voltage Calculation (When VESECTORx<SECTOR[3:0]>= "0") 465
Table 13.14 Output Control 466
Table 13.15 Trigger Generation 467
Table 13.16 Input Processing 468
Table 13.17 3-2-phase Conversion 469
Table 13.18 αβ-dq Coordinate Axis Conversion 469
Table 13.19 Combination of VE Channel and PMD Channel 470
Table 13.20 Combination of VE Channel, ADC Unit and AD Conversion Result Register 470
Table 14.1 Counter control 494
Table 14.2 Interrupt factors 495
Table 16.1 Voltage detection circuit characteristics 504
Table 17.1 Status of OFD in each Operation Mode 513
Table 17.2 Used Frequency and Setting Value 514
Table 18.1 Watchdog timer detection time (fc = 80MHz) 519
Table 20.1 Write/erase time 531
Table 20.2 Rewrite Method 532
Table 20.3 Operation Modes 534
Table 20.4 Operation Mode Setting 534
Table 20.5 Mapping of Internal Flash Memory to RAM 550
Table 20.6 Connection of Pins 551
Table 20.7 Operation Command Data 552
Table 20.8 Memory Constraints for Single Boot 552
Table 20.9 Transfer Format of Boot Program [RAM Transfer] 553
Table 20.10 Transfer Format of Boot Program [Flash Memory Chip Erase and Protect Bit Erase] 555
Table 20.11 Target area 556
Table 20.12 Password Area 559
Table 20.13 ACK Response Data for Serial Operation Determination Data 564
Table 20.14 ACK Response Data for Operation Command Data 564
Table 20.15 ACK Response Data for CHECKSUM Data 564
Table 20.16 ACK Response Data for Flash Memory Chip Erase and Protect Bit Erase Operation 564
Table 20.17 Password Area 568
Table 20.18 Flash Memory Function 571
Table 20.19 Flash Memory Access by Internal CPU 580
Table 20.20 Address Bit Configuration during Bus Write cycle 581
Table 20.21 Block Address 582
Table 20.22 Protect Bit Program Address 582
Table 20.23 Protect Bit Erase Address 582
Table 20.24 ID Addresses (IA) of 4th Bus Write Cycle of ID-Read Command and Data (ID) that can be Read by Subsequent 32-bit Transfer Instruction 583
Table 21.1 Restrictions by the security function 587

Table 22.1 Relation Between Debug Interface Pins and General-purpose Ports 593
Table 22.2 Value of Port; Which is Assigned as Debug Interface Pin; Control Registers After Releasing RESET 594
Table 22.3 Debug Interface And Using Debug Interface pins 595
Table 24.1 Absolute Maximum Ratings 602
Table 26.1 Revision History 623

****

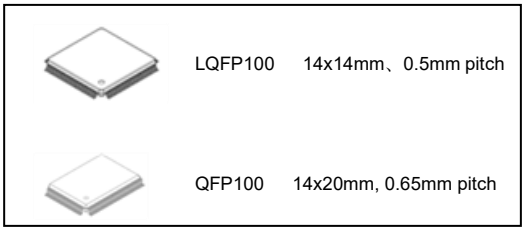
Arm, Cortex and Thumb are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.



All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

CMOS Digital Integrated Circuit Silicon Monolithic

TMPM370FYDFG
TMPM370FYFG



TMPM370FYDFG/TMPM370FYFG are a 32-bit RISC microprocessor series with an Arm® Cortex®-M3 microprocessor core.

Product Name	ROM (FLASH)	RAM	Package
TMPM370FYDFG	256Kbytes	10Kbytes	QFP100-P-1420-0.65Q
TMPM370FYFG	256Kbytes	10Kbytes	LQFP100-P-1414-0.50H

1. Features

(1) Arm Cortex-M3 microprocessor core

(a) Achieves improved code efficiency with Thumb®-2 instructions.

- New 16-bit Thumb instructions for improved program flow
- New 32-bit Thumb instructions for improved performance and code size
- Improved code efficiency with a 32-bit/16-bit mixed Thumb instruction set

(b) Achieves both high performance and low power consumption

[High performance]

- Execute 32-bit multiplication (32 × 32 = 32bit) in one clock
- Execute division in 2 to 12 clocks

[Low power consumption]

- Optimized design using a low power consumption library
- Standby function that stops the operation of the microprocessor core

(c) High-speed interrupt response suitable for real-time control

- Instructions with long execution time can be interrupted by interrupt.
- The hardware handles the push to the stack automatically.

(2) Internal program memory and data memory

- Internal RAM: 10Kbyte
- Internal Flash ROM: 256Kbyte

(3) 16-bit timer/event counter (TMRB): 8 channels

- 16-bit interval timer mode
- 16-bit event counter mode
- Input capture function
- External trigger PPG output
-

(4) Watchdog timer (WDT): 1 channel

- Watchdog timer (WDT) generates a reset or a non-maskable interrupt (NMI).

(5) Power-on reset circuit (POR)

(6) Voltage detection circuit (VLTD)

(7) Oscillation frequency detection circuit (OFD)

(8) Vector engine (VE): 1unit

- Calculation circuit for motor control
- Support to 2 motors

(9) Programmable motor driver circuit (PMD): 2 channels

- 3phase complementary PWM output
- Synchronous trigger generation that links AD converter
- Emergency stop protection function (EMG/Comparator output)

(10) Encoder input circuit (ENC): 2 channels

- Support to incremental encoder (AB signal/ABZ signal)
- Rotation direction detection circuit
- Counter for absolute position detection
- Position compare circuit
- Noise filter
- Support to 3 phase sensor input

(11) Serial channel (SIO/UART): 4 channels

- Either UART mode or SIO mode can be selected (4byte FIFO equipped).

(12) 12-bit analog-to-digital converter (ADC) : 2unit (Analog input: 22 channels)

- Starting function by the internal trigger: TMRB interrupt/PMD trigger
- Constant conversion mode
- AD monitoring 2channels
- Conversion speed 2μs (@AD conversion clock = 40MHz)

(13) OP-Amp (AMP): 4 channels

- 8 gain can be selected.

(14) Comparator (CMP): 3+1 channel(s)

- Protection for motor emergence stop
- 2 input type (OPAMP output/analog input)

(15) Input/output ports (PORT): 76 pins

- I/O pin: 74 pins
- Input pin: 2 pins

(16) Interrupt factors

- Internal factors: 62 factors

  The priority can be set 7 levels. (except the watchdog timer interrupt)
- External factors: 16 factors

  The priority can be set over 7 levels.

(17) Low-power consumption mode

- IDLE, STOP mode

(18) Clock/mode control (CG)

- Internal PLL (8 multiplication)
- Clock gear function : The high-speed clock can be divided into 1, 2, 4, 8 or 16.

(19) Endian

- Little endian

(20) Maximum operating frequency: 80MHz

(21) Operating voltage range

- 4.5 to 5.5 V (with internal regulator)

(22) Temperature range

- -40°C to 85°C (except Flash writing/erasing and debug)
- 0°C to 70°C (while Flash writing/erasing)

(23) Package

- QFP100-P-1420-0.65Q (14mm × 20mm, 0.65mm pitch)
- LQFP100-P-1414-0.50H (14mm × 14mm, 0.5mm pitch)

1.1. Products Lists Categorized by Functions

Table 1.1 Products Lists

Built-in Functions		TMPM370FYDFG
		TMPM370FYDFG
Memory	Flash (KB)	256
	RAM (KB)	10
I/O port	PORT (pin)	76
Interrupt	Internal	62
	External	16
Timer function	TMRB (ch)	8
Serial communication function	SIO/UART (ch)	4
Analog function	12-bit ADC Unit A/B (AIN ch)	22
	AMP (ch)	4
	CMP (ch)	3+1
Motor control peripherals	VE (unit)	1
	PMD (ch)	2
	ENC (ch)	2
System function	VLTD (ch)	1
	WDT (ch)	1
	OFD (ch)	1
	POR	1
Debug interface	Debug	JTAG/SW/TRACE
Package	Package type	QFP100 (14mm x 20mm、0.65mm pitch)
		LQFP100 (14mm x 14mm、0.5mm pitch)

1.2. Block Diagram

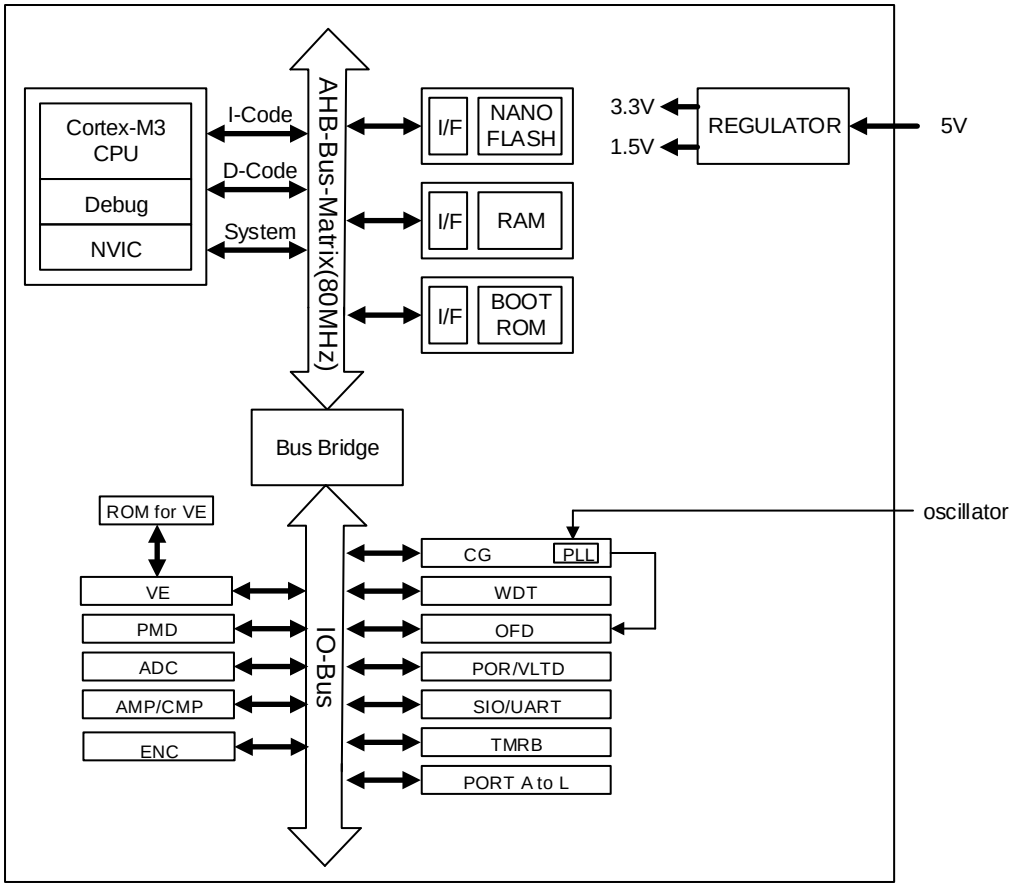
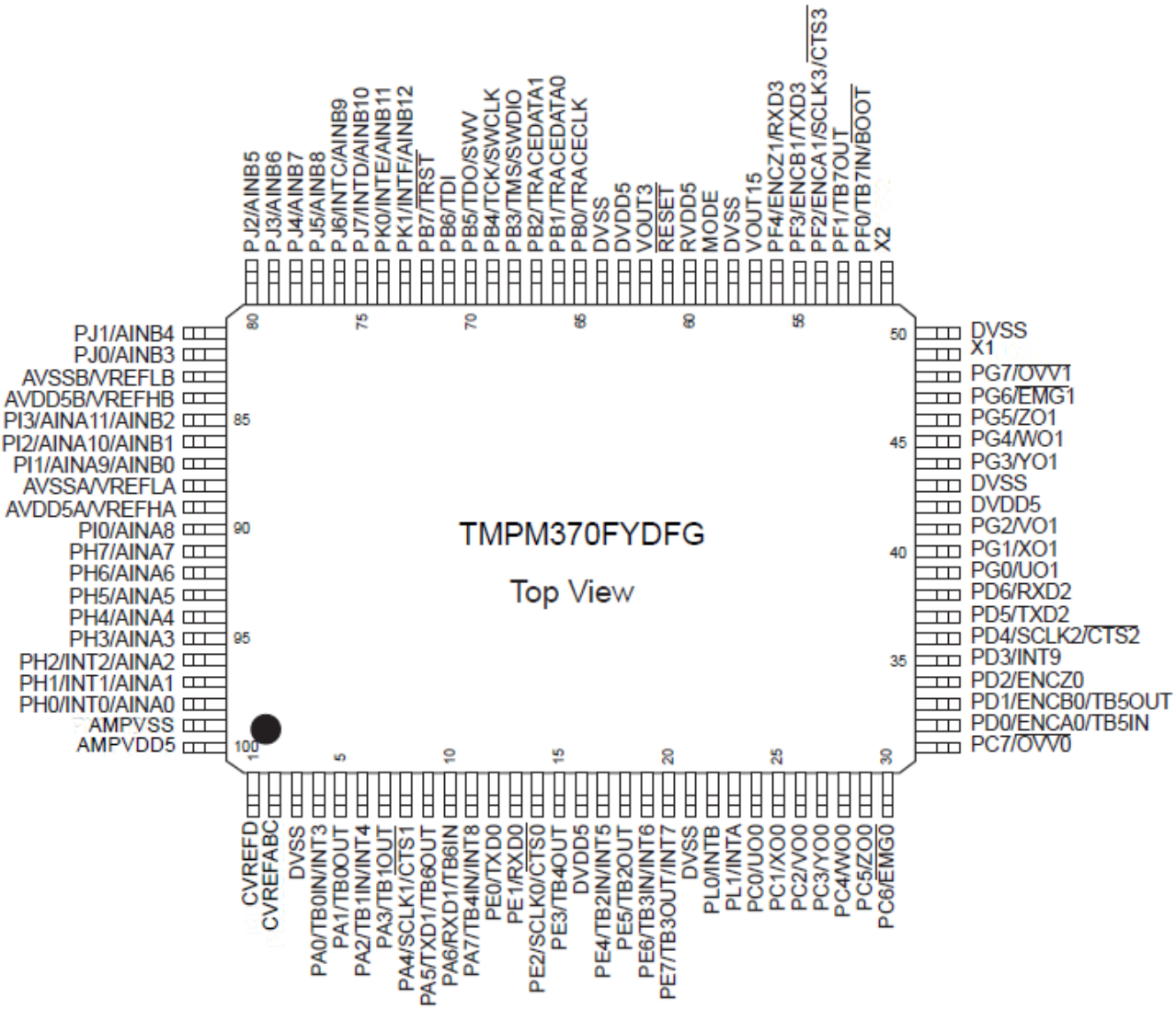


Figure 1.1 Block diagram

1.3. Pin Layout (top view)

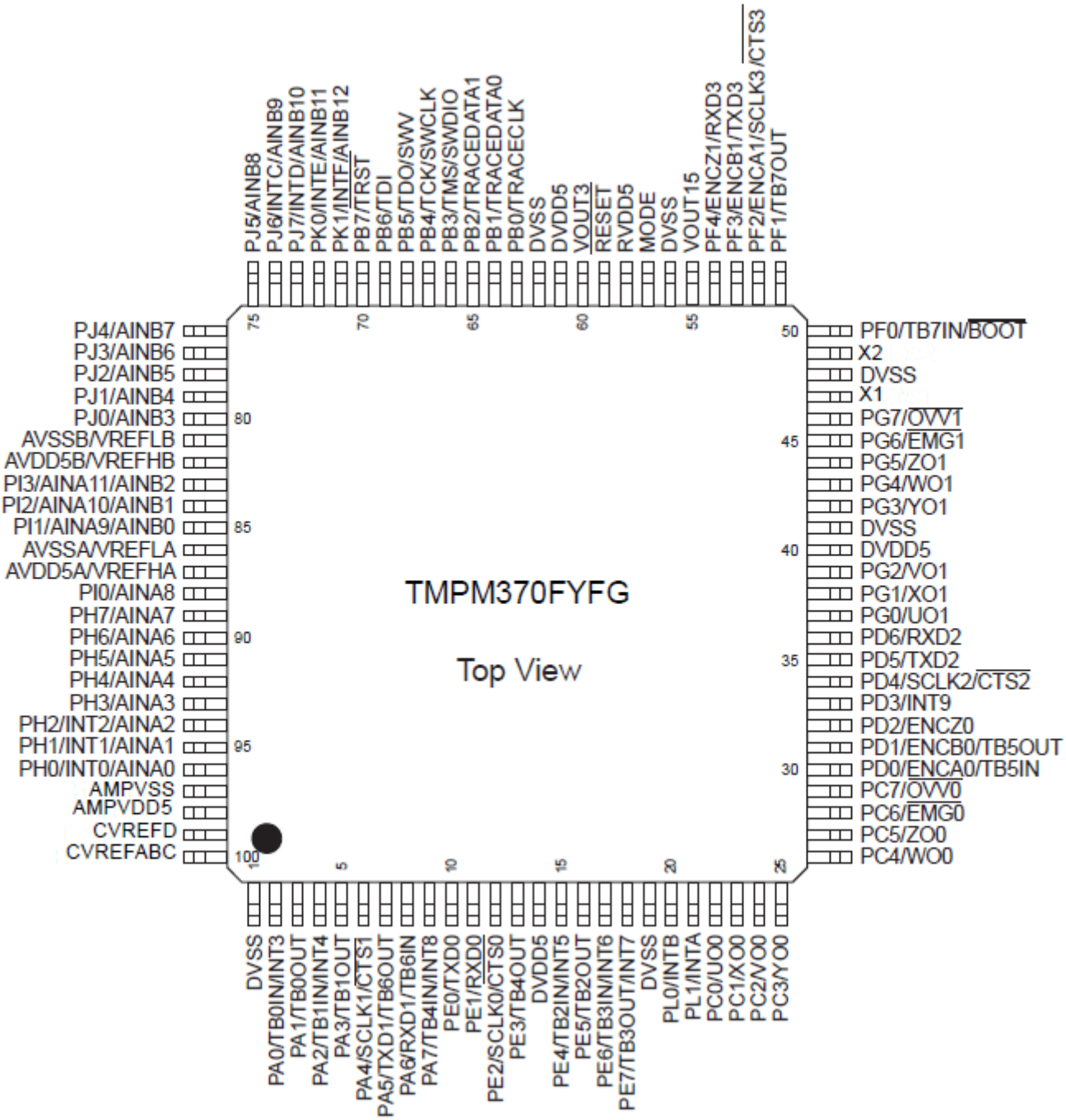
1.3.1. QFP100

The pin layout of TMPM370FYDFG is shown below.



1.3.2. LQFP100

The pin layout of TMPM370FYFG is shown below.



1.4. Pin Name and Functions

1.4.1. Functional Pin and Ports Assignment (Pin Number)

Following table shows a pin number of the port assignment and each product which were seen from the functional pin.

Table 1.2 Signal connection List: SIO/UART

Function	Combination functional pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYDFG (LQFP100)
SIO/UART	RXD0	PE1	13	11
	TXD0	PE0	12	10
	$\overline{\text{CTS0}}$/SCLK0	PE2	14	12
	RXD1	PA6	10	8
	TXD1	PA5	9	7
	$\overline{\text{CTS1}}$/SCLK1	PA4	8	6
	RXD2	PD6	38	36
	TXD2	PD5	37	35
	$\overline{\text{CTS2}}$/SCLK2	PD4	36	34
	RXD3	PF4	56	54
	TXD3	PF3	55	53
	$\overline{\text{CTS3}}$/SCLK3	PF2	54	52

Table 1.3 Signal connection List: TMRB

Function	Combination functional pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYDFG (LQFP100)
TMRB	TB0IN	PA0	4	2
	TB0OUT	PA1	5	3
	TB1IN	PA2	6	4
	TB1OUT	PA3	7	5
	TB2IN	PE4	17	15
	TB2OUT	PE5	18	16
	TB3IN	PE6	19	17
	TB3OUT	PE7	20	18
	TB4IN	PA7	11	9
	TB4OUT	PE3	15	13
	TB5IN	PD0	32	30
	TB5OUT	PD1	33	31
	TB6IN	PA6	10	8
	TB6OUT	PA5	9	7
	TB7IN	PF0	52	50
	TB7OUT	PF1	53	51

Table 1.4 Signal connection List: ADC/AMP/CMP

Function	Combination functional pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYDFG (LQFP100)
ADC	AINA0	PH0	98	96
	AINA1	PH1	97	95
	AINA2	PH2	96	94
	AINA3	PH3	95	93
	AINA4	PH4	94	92
	AINA5	PH5	93	91
	AINA6	PH6	92	90
	AINA7	PH7	91	89
	AINA8	PI0	90	88
	AINA9/AINB0	PI1	87	85
	AINA10/AINB1	PI2	86	84
	AINA11/AINB2	PI3	85	83
	AINB3	PJ0	82	80
	AINB4	PJ1	81	79
	AINB5	PJ2	80	78
	AINB6	PJ3	79	77
	AINB7	PJ4	78	76
	AINB8	PJ5	77	75
	AINB9	PJ6	76	74
	AINB10	PJ7	75	73
	AINB11	PK0	74	72
	AINB12	PK1	73	71
AMP/CMP	AINA9/AINB0	PI1	87	85
	AINA10/AINB1	PI2	86	84
	AINA11/AINB2	PI3	85	83
	AINB3	PJ0	82	80

Table 1.5 Signal connection List: INT/PMD

Function	Combination functional pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYDFG (LQFP100)
INT	INT0	PH0	98	96
	INT1	PH1	97	95
	INT2	PH2	96	94
	INT3	PA0	4	2
	INT4	PA2	6	3
	INT5	PE4	17	15
	INT6	PE6	19	17
	INT7	PE7	20	18
	INT8	PA7	11	9
	INT9	PD3	35	33
	INTA	PL1	23	21
	INTB	PL0	22	20
	INTC	PJ6	76	74
	INTD	PJ7	75	73
	INTE	PK0	74	72
	INTF	PK1	73	71
PMD	UO0	PC0	24	22
	XO0	PC1	25	23
	VO0	PC2	26	24
	YO0	PC3	27	25
	WO0	PC4	28	26
	ZO0	PC5	29	27
	$\overline{\text{EMG0}}$	PC6	30	28
	$\overline{\text{OVV0}}$	PC7	31	29
	UO1	PG0	39	37
	XO1	PG1	40	38
	VO1	PG2	41	39
	YO1	PG3	44	42
	WO1	PG4	45	43
	ZO1	PG5	46	44
	$\overline{\text{EMG1}}$	PG6	47	45
	$\overline{\text{OVV1}}$	PG7	48	46

Table 1.6 Signal connection List: ENC/DEBUG/Control Pin/Power Supply

Function	Combination functional pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYDFG (LQFP100)
ENC	ENCA0	PD0	32	30
	ENCB0	PD1	33	31
	ENCZ0	PD2	34	32
	ENCA1	PF2	54	52
	ENCB1	PF3	55	53
	ENCZ1	PF4	56	54
DEBUG	TMS/SWDIO	PB3	68	66
	TCK/SWCLK	PB4	69	67
	TDO/SWV	PB5	70	68
	TDI	PB6	71	69
	$\overline{\text{TRST}}$	PB7	72	70
	TRACECLK	PB0	65	63
	TRACEDATA0	PB1	66	64
	TRACEDATA1	PB2	67	65
Control Pin	$\overline{\text{BOOT}}$	PF0	52	50
	MODE	-	59	57
	$\overline{\text{RESET}}$	-	61	59
Power supply	DVSS	-	3	1
	DVSS	-	21	19
	DVSS	-	43	41
	DVSS	-	50	48
	DVSS	-	58	56
	DVSS	-	64	62
	DVDD5	-	16	14
	DVDD5	-	42	40
	DVDD5	-	63	61
	RVDD5	-	60	58
	VOUT15	-	57	55
	VOUT3	-	62	60
	AVSSB/VREFLB	-	83	81
	AVDD5B/VREFHB	-	84	82
	AVSSA/VREFLA	-	88	86
	AVDD5A/VREFHA	-	89	87
	CVREFABC	-	2	100
	CVREFD	-	1	99
	AMPVDD5	-	100	98
	AMPVSS	-	99	97

1.5. Power Supply Type and Supply Pins

Table 1.7 Power supply type and supply pins

Power Supply Type	Voltage range	Power supply pins
DVDD5	4.5 to 5.5V	PA, PB, PC, PD, PE, PF, PG, PL $\overline{\text{RESET}}$, MODE
AVDD5A		PH、PI
AVDD5B		PJ, PK
RVDD5		-
VOUT15	1.35 to 1.65V	VOUT15 is a capacitor connection pin for stabilizing the output of the internal regulator. Cannot be supplied to external circuits. VOUT15 must be connected to DVSS through 3.3 to 4.7μF capacitor.
VOUT3	2.7 to 3.6V	VOUT3 is a capacitor connection pin for stabilizing the output of the internal regulator. Cannot be supplied to external circuits. VOUT3 must be connected to DVSS through 3.3 to 4.7μF capacitor.

2. Processor Core

The TX03 series has a high-performance 32-bit processor core (the Arm Cortex-M3 processor core). For information on the operations of this processor core, please refer to the "Arm documentation set of the Arm Cortex-M series processor". This section explains the product-specific information.

2.1. Information on the processor core

The following table shows the revision of the processor core in the .

For details on the CPU core and architecture, refer to the documentation on the Arm’s website.

Table 2.1 Core Revision

Product Name	Core Revision
TMPM370FYDFG	r2p0
TMPM370FYFG	

2.2. Configurable Options

In the Cortex-M3 core, some blocks can be selected to implement. The following table shows the configurations of the TMPM370FYDFG/TMPM370FYFG.

Table 2.2 Configurable options and their implementations

Configurable option	Implementation
FPB	Literal comparator: 2 Instruction comparator: 6
DWT	Comparator: 4
ITM	Available
MPU	Not available
ETM	Available
AHB-AP	Available
AHB trace Macrocell interface	Available
TPIU	Available
WIC	Not available
Debug port	JTAG/serial wire

2.3. Exceptions / Interrups

Exceptions and interruptions are described in the following section.

2.3.1. Number of Interrupt Inputs

The number of interrupts can optionally be defined from 1 to 240 in the Cortex-M3 core.

The number of interrupts for TMPM370FYDFG/TMPM370FYFG is 78. The number of interrupt inputs is reflected in <INTLINESNUM[4:0]>bit of NVIC register. In this product, when read <INTLINESNUM[4:0]>bit, “0x00” is read out.

2.3.2. Number of Priority Level Interrupt Bits

The Cortex-M3 core can optionally configure the number of priority level interrupt bits from 3 bits to 8 bits.

TMPM370FYDFG/TMPM370FYFG have 3 priority level interrupt bits. The number of priority level interrupt bits is used for assigning a priority level in the interrupt priority registers and system handler priority registers.

2.3.3. SysTick

The Cortex-M3 core has a SysTick timer which can generate SysTick exception.

For the detail of SysTick, refer to the "6.3. SysTick" and "6.5.2. NVIC Registers" in "Exception".

2.3.4. SYSRESETREQ

The Cortex-M3 core outputs SYSRESETREQ signal when <SYSRESETREQ>bit of Application Interrupt and Reset Control Register is set.

TMPM370FYDFG/TMPM370FYFG provide the same operation as warm reset when SYSRESETREQ signal is output.

2.3.5. LOCKUP

When irreparable exception generates, the Cortex-M3 core outputs LOCKUP signal to show a serious error included in software.

TMPM370FYDFG/TMPM370FYFG do not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interrupt (NMI) or reset.

2.3.6. Auxiliary Fault Status register

The Cortex-M3 core provides auxiliary fault status registers to supply additional system fault information to software.

However, TMPM370FYDFG/TMPM370FYFG are not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

2.4. Events

The Cortex-M3 core has event output signals and event input signals. An event output signal is output by SEV instruction execution. If an event is input, the core returns from low-power consumption mode caused by WFE instruction.

TMPM370FYDFG/TMPM370FYFG do not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

2.5. Power Management

The Cortex-M3 core provides power management system which uses SLEEPING signal and SLEEPDEEP signal.

SLEEPDEEP signal is output when <SLEEPDEEP>bit of System Control Register is set.

These signals are output in the following circumstances:

- Wait-For-Interrupt (WFI) instruction execution
- Wait-For-Event (WFE) instruction execution
- The timing when interrupt service routine (ISR) exits in case that <SLEEPONEXIT>bit of System Control Register is set.

TMPM370FYDFG/TMPM370FYFG do not use SLEEPDEEP signal so that <SLEEPDEEP>bit must not be set.

And also event signal is not used so that please do not use WFE instruction.

For detail of power management, refer to "5. Clock/Mode Control(CG)"

2.6. Exclusive access

In Cortex-M3 core, supports exclusive access to DCode and system buses.

However, TMPM370FYDFG/TMPM370FYFG do not use this function.

3. Memory Map

3.1. Memory Map

The memory map for TMPM370FYDFG/TMPM370FYFG is based on the Arm Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM370FYDFG/TMPM370FYFG are mapped to the Code, SRAM and peripheral regions of the Cortex-M3 respectively. The special function register (SFR) means the control registers of all input/output ports and peripheral functions. The SRAM and SFR areas are all included in the bit-band region.

The CPU register area is the processor core’s internal register region.

For more information on each region, see the "the Arm documentation set for Cortex-M series processors".

Note that access to regions indicated as "Fault" causes a memory fault if memory faults are enabled, or causes a hard fault if memory faults are disabled. Also, do not access the vendor-specific region.

3.1.1. Memory Map

Figure 3.1 shows the memory map.

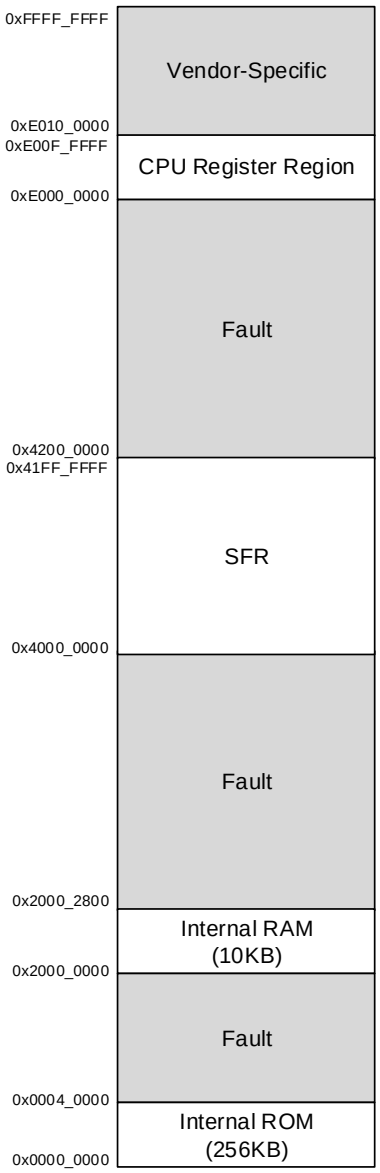


Figure 3.1 Memory Map

3.2. Details of SFR area

Table 3.1 shows the details of the SFR area.

Do not access a reserved area in Table 3.1. See the chapter of each peripheral function for details.

Table 3.1 Details of SFR

Start Address	End Address	Peripheral
0x4000_0000	0x4000_037F	PORT
0x4000_0380	0x4000_FFFF	Reserved
0x4001_0000	0x4001_01FF	TMRB
0x4001_0200	0x4001_03FF	Reserved
0x4001_0400	0x4001_053F	ENC
0x4001_0540	0x4002_007F	Reserved
0x4002_0080	0x4002_017F	SIO/UART
0x4002_0180	0x4002_FFFF	Reserved
0x4003_0000	0x4003_02FF	ADC
0x4003_0300	0x4003_03FF	Reserved
0x4003_0400	0x4003_043F	AMP/CMP
0x4003_0440	0x4003_FFFF	Reserved
0x4004_0000	0x4004_003F	WDT
0x4004_0040	0x4004_01FF	Reserved
0x4004_0200	0x4004_023F	CG
0x4004_0240	0x4004_07FF	Reserved
0x4004_0800	0x4004_083F	OFD
0x4004_0840	0x4004_08FF	Reserved
0x4004_0900	0x4004_093F	VLTD
0x4004_0940	0x4004_FFFF	Reserved
0x4005_0000	0x4005_023F	VE
0x4005_0240	0x4005_03FF	Reserved
0x4005_0400	0x4005_04FF	PMD
0x4005_0500	0x4007_FFFF	Reserved
0x4008_0000	0x41FF_EFFF	Hard fault
0x41FF_F000	0x41FF_F03F	FLASH
0x41FF_F040	0x41FF_FFFF	Reserved

4. RESET Operation

4.1. State before RESET

When the power on, the internal state of is undefined until the power-on reset is generated or the "Low" level inputs to $\overline{\text{RESET}}$ pin. Therefore, The setting in registers and the state of each pin are undefined.

4.2. State during RESET

TMPM370FYDFG/TMPM370FYFG have the POR. The power-on reset signal is generated after the power on, it resets the internal state of TMPM370FYDFG/TMPM370FYFG.

And, when resetting the internal state of, TMPM370FYDFG/TMPM370FYFG $\overline{\text{RESET}}$ pin must be held "Low" level for at least 12 system clocks (1.2μs@10MHz, under oscillating) at the voltage of a power supply within an operating voltage range and the stable state of the high-frequency oscillation.

Note1: The power can be turned-on while $\overline{\text{RESET}}$ pin is input "Low" level.

Note2: The RESET operation may alter the internal RAM data.

4.3. State after releasing RESET

The control registers in Coretex-M3 core and the registers in peripheral functions are initialized after releasing RESET.

Because the PLL circuit stops after releasing RESET, CGOSCCR and CGPLLSEL must be set to use PLL circuit.

The CPU branches to an interrupt service routine of the RESET after handling RESET exception. The address of an interrupt service routine of the RESET is stored at 0x0000_0004.

5. Clock/Mode Control(CG)

5.1. Features

The clock/mode control block enables to select clock gear, prescaler clock and warming-up of the PLL clock multiplication circuit and oscillator.

There is also the low-power consumption mode which can reduce power consumption by mode transitions.

This chapter describes how to control clock operating modes and mode transitions.

The clock/mode control block has the following functions:

- Controls the system clock
- Controls the prescaler clock
- Controls the PLL multiplication circuit
- Controls the warming-up timer

In addition to NORMAL mode, the TMPM370FYDFG/TMPM370FYFG can operate low-power mode to reduce power consumption according to its usage conditions.

5.2. Registers

5.2.1. Register List

The following table shows the CG-related registers and addresses.

Register name		Address (Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
PLL selection register	CGPLLSEL	0x000C
Reserved	–	0x0010

Note: Access to the "Reserved" area is prohibited.

5.2.2. CGSYSCR (System control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	1
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	FPSEL	-	PRCK		
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	GEAR		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:18	-	R	Read as "0".
17:16	-	R/W	Write as "01".
15:13	-	R	Read as "0".
12	FPSEL	R/W	Selects fperiph source clock 0: fgear 1: fc
11	-	R	Read as "0".
10:8	PRCK[2:0]	R/W	Selects Prescaler clock 000: fperiph 001: fperiph / 2 010: fperiph / 4 011: fperiph / 8 100: fperiph / 16 101: fperiph / 32 110: Reserved 111: Reserved Specifies the prescaler clock to peripheral function.
7:3	-	R	Read as "0".
2:0	GEAR[2:0]	R/W	Selects High-speed clock (fc) gear 000: fc 001: Reserved 010: Reserved 011: Reserved 100: fc / 2 101: fc / 4 110: fc / 8 111: fc / 16

5.2.3. CGOSCCR (Oscillation control register)

	31	30	29	28	27	26	25	24
bit symbol	WUODR							
After reset	1	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	WUODR				-	-	-	-
After reset	0	0	0	0	1	1	1	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	XEN1
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:20	WUODR[11:0]	R/W	Specifies count time of the warming-up timer.
19:16	-	R/W	Write as "1110".
15:12	-	R/W	Write as "0".
11:10	-	R	Read as "0".
9	-	R/W	Write as "0".
8	XEN1	R/W	High-speed oscillator(External OSC) 0: Stop 1: Oscillation Specifies operation of the high-speed oscillator (OSC).
7:4	-	R	Read as "0".
3	WUPSEL1	R/W	Clock source for warming-up timer Write as "0".
2	PLLON	R/W	PLL operation 0: Stop 1: Oscillation Specifies operation of the PLL. It stops after reset. Setting the bit is required.
1	WUEF	R	Status of warming-up timer (WUP) (Note) 0: Warming-up completed 1: Warming-up operation This bit shows the status of the warming-up timer.
0	WUEON	W	Operation of warming-up timer (Note) 0: don't care 1: Starting warming-up Set this bit to "1" to start the warming-up timer.

Note: Setting this register is not required when return from low-power consumption mode which is warmed up automatically (Table 5.6).When WUP is started by software, operation mode must be changed after confirming that warming-up is completed.

5.2.4. CGSTBYCR (Standby control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	DRVE
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	RXEN
After reset	0	0	0	0	0	0	0	1
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	STBY		
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31:18	-	R	Read as "0".
17	-	R/W	Write as "0".
16	DRVE	R/W	Pin status in STOP mode 0: Inactive 1: Active
15:10	-	R	Read as "0".
9	-	R/W	Write as "0".
8	RXEN	R/W	High-speed oscillator operation after releasing the STOP mode. Write as "1".
7:3	-	R	Read as "0".
2:0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP 010: Reserved 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved

5.2.5. CGPLLSEL (PLL Selection Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	1	0	1	0	0	0	0	1
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	-	PLLSEL
After reset	0	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:12	-	R/W	Write as "1010".
11	-	R	Read as "0".
10:1	-	R/W	Write as "00_1001_1111".
0	PLLSEL	R/W	Use of PLL 0: f_{OSC} use 1: PLL use Specifies use or disuse of the clock multiplied by the PLL. " f_{OSC}" is automatically set after reset. Setting this bit to "1" is required to use the PLL.

Note: Be sure to disable OFD when enabling PLL. Especially, be careful setting PLL in the initial routine because OFD is available after OFD reset.

Example:

OFDCR1<OFDWEN7:0> = "0xF9" : Sets control register write enable code

OFDCR2<OFDEN7:0> = "0x00" : Disable OFD

CGPLLSEL<PLLSEL> = "1" : Enable PLL

OFDMNPLLON = "xxxx" : Sets lower detection frequency

OFDMXPLLON = "yyyy" : Sets higher detection frequency

OFDCR2<OFDEN7:0> = "0xE4" : Enable OFD

OFDCR1<OFDWEN7:0> = "0x06" : Sets control register write disable code

5.3. Clock control

5.3.1. Clock Type

Each clock is defined as follows :

f_{OSC}	: Clock input from external high-speed oscillator (X1 and X2)
f_{PLL}	: Clock octupled by PLL (x8)
fc	: Clock specified by CGPLLSEL<PLLSEL> (high-speed clock)
fgear	: Clock specified by CGSYSCR<GEAR[2:0]>
fsys	: The same clock as fgear (system clock)
fperiph	: Clock specified by CGSYSCR<FPSEL>
ΦT0	: Clock specified by CGSYSCR<PRCK[2:0]> (Prescaler clock)

The high-speed clock fc and the prescaler clock ΦT0 are dividable as follows.

High-speed clock	: fc, fc / 2, fc / 4, fc / 8, fc / 16
Prescaler clock	: fperiph, fperiph / 2, fperiph / 4, fperiph / 8, fperiph / 16, fperiph / 32

5.3.2. Initial Values after Reset

Reset operation initializes the clock configuration as follows.

High-speed oscillator (OSC)	: Stop (X1, X2)
PLL (Clock multiplication circuit)	: Stop
High-speed clock gear	: fc (no frequency dividing)

Reset operation causes all the clock configurations to be the same as f_{OSC}.

fc = f_{OSC}

fsys = fc (= f_{OSC})

fperiph = fc (= f_{OSC})

ΦT0 = fperiph (= f_{OSC})

5.3.3. Clock system Diagram

Figure 5.1 shows the clock system diagram.

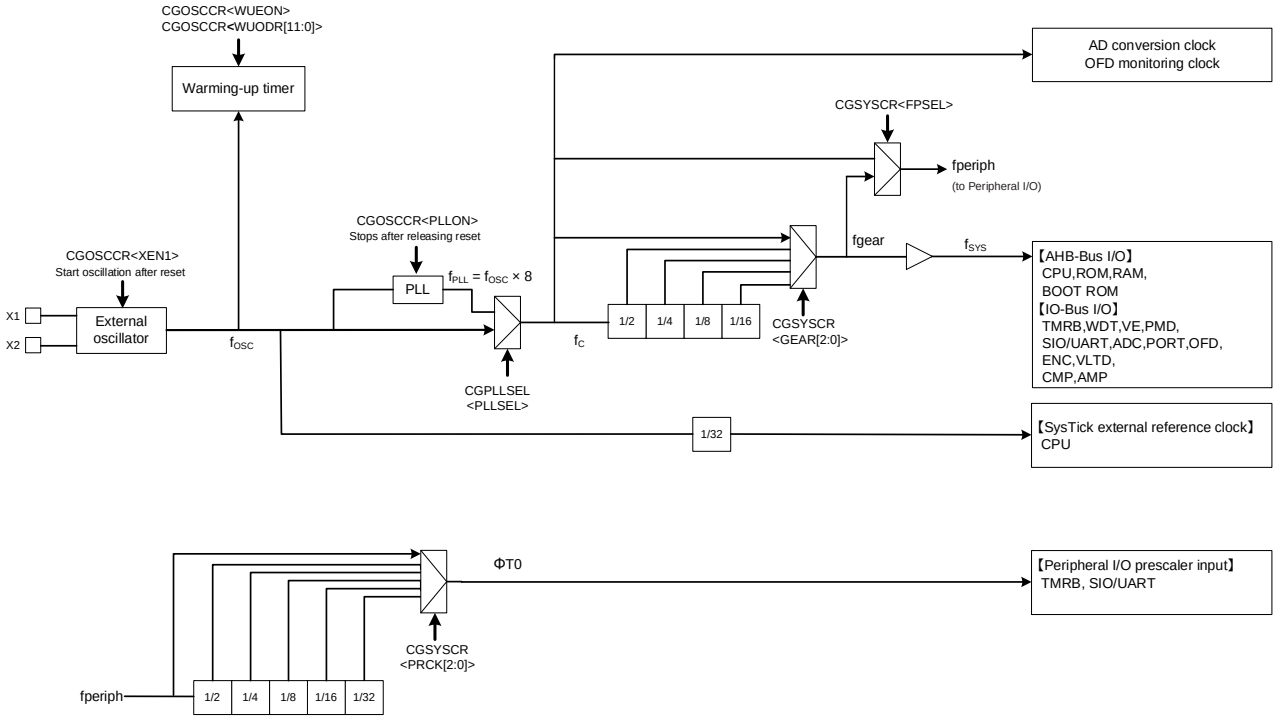


Figure 5.1 Clock Block Diagram

The input clocks to the selectors shown with an arrow are set as default after reset.

5.3.4. Clock Multiplication Circuit (PLL)

This circuit outputs the f_{PLL} clock that is octuple of the high-speed oscillator output clock (f_{OSC}) As a result, the oscillation frequency of the oscillator is low and the internal clock can be high-speed.

The PLL is disabled after reset. To enable the PLL, set the CGOSCCR<PLLON> to "1" and then set the CGPLLSEL<PLLSEL> to "1". Then f_{PLL} clock output is octuple of the high-speed oscillator (f_{OSC}).

The PLL requires a certain amount of time to be stabilized, which should be secured using the warming-up function or other methods.

Note: When the PLL starts operation, it is necessary to wait approximately 200μs or more for the PLL to be stabilized.

5.3.4.1. The sequence of PLL setting

The following shows PLL setting sequence after reset.

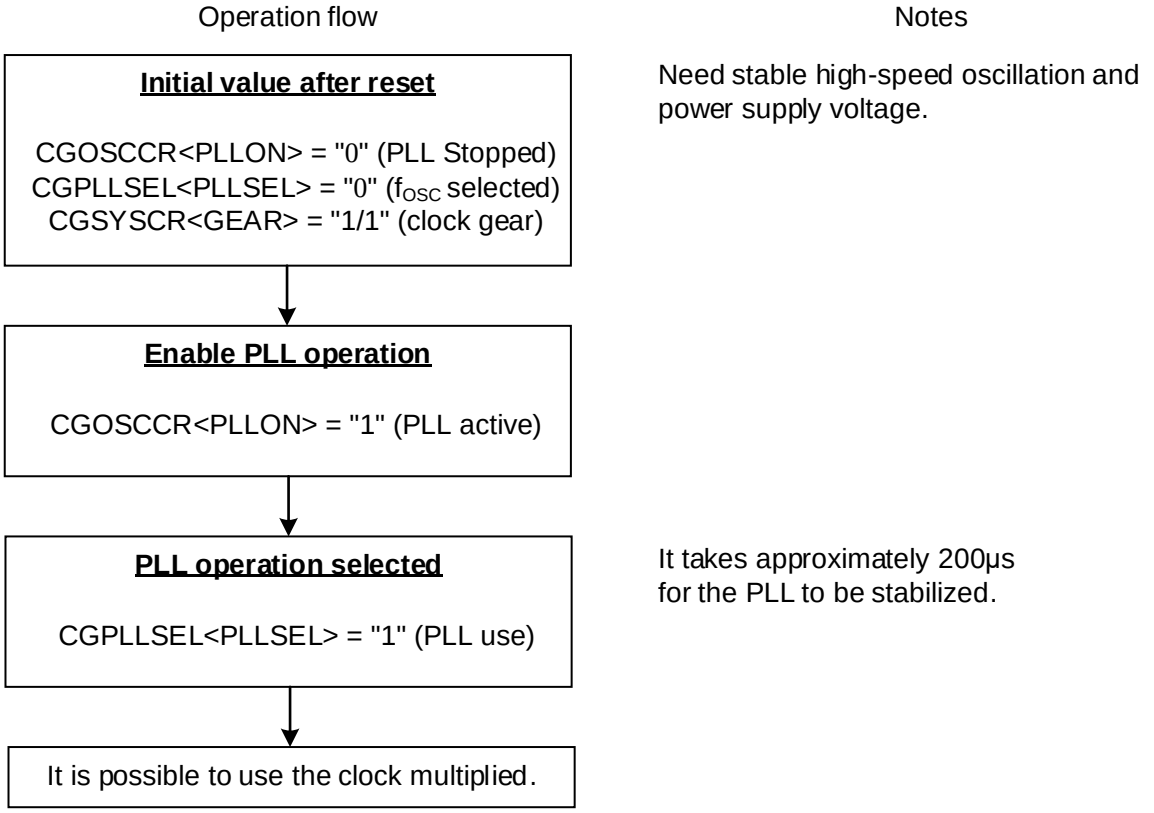


Figure 5.2 PLL setting sequence after reset

Note: When you stop PLL, please confirm that it is the register CGPLLSEL<PLLSEL> = "0" after setting the CGPLLSEL<PLLSEL> = "0". Then, please set CGOSCCR<PLLON> = "0" (PLL stopped).

5.3.5. Warming-up function

The warming-up function secures the stability time for the oscillator and the PLL with the warming-up timer. The warming-up function is used when returning from STOP mode. For the detail of function, see "5.6.6. Warming-up".

Note: Do not shift to STOP mode, while operating warming-up timer.

The warming-up function is also used when returning from STOP mode. In this case, an return interrupt from the low-power consumption mode triggers the automatic timer count. After the specified time is reached, the system clock is output and the CPU starts operation.

In STOP mode, the PLL is disabled. When returning from this mode, configure the warming-up time in consideration of the stability time of the PLL and the internal oscillator.

How to configure the warming-up function.

(1) Specify the count up clock

Specify the count up clock for the warming-up counter in the CGOSCCR<WUPSEL1> bit (Write "0" to <WUPSEL1>.).

Specify the warming-up counter value

The warming-up time can be set by setting the CGOSCCR<WUODR[11:0]>.

The following shows the formula of warming-up and setting example.

$$\text{Warming-up cycles} = \frac{\text{Warming-up time}}{\text{Cycle of warming-up clock}}$$

<example> Setting 5ms of warming-up time with 8MHz high-speed oscillator

$$\frac{\text{Warming-up time}}{\text{Cycle of warming-up clock}} = \frac{5\text{ms}}{1\ /\ 8\text{MHz}} = 40000 \text{ cycles} = \text{"0x9C40"}$$

Drop the lower 4 bits, write "0x9C4" into the CGOSCCR<WUODR[11:0]>.

(2) Start and confirm the completion of warming-up

The CGOSCCR<WUEON> and <WUEF> are used to start and confirm the completion of warming-up, respectively, through software (instruction).

Note: The warming-up timer operates according to the oscillation clock, and it may contain errors if there is any fluctuation in the oscillation frequency. Therefore, the warming-up time should be taken as approximate time.

The following shows the warming-up setting.

<example> Securing the stability time for the PLL (fc = f_{OSC})

CGOSCCR<WUPSEL1> = "0" : Specify the clock source for warming-up timer
CGOSCCR<WUODR[11:0]> = "0x9C4" : Warming-up time setting
CGOSCCR<WUEON> = "1" : Enable warming-up counting (WUP)
Read CGOSCCR<WUEF> : Wait until the state becomes "0" (warming-up is finished)

5.3.6. System Clock

The TMPM370FYDFG/TMPM370FYFG offer high-speed clock as system clock. The high-speed clock is dividable.

- External oscillator frequency : 8 MHz to 10MHz
- Clock gear : Divided by 1, 2, 4, 8, 16 (after reset: divided by 1)

Table 5.1 Range of high-speed frequency (unit: MHz)

Input frequency		Min operating frequency	Max operating frequency	After reset (PLL = OFF, CG = 1 / 1)	Clock gear (CG): PLL = ON					Clock gear (CG): PLL = OFF				
					1 / 1	1 / 2	1 / 4	1 / 8	1 / 16	1 / 1	1 / 2	1 / 4	1 / 8	1 / 16
f_{OSC}	8	1	80	8	64	32	16	8	4	8	4	2	1	-
	10			10	80	40	20	10	5	10	5	2.5	1.25	-

Note1: PLL = ON/OFF: setting by CGOSCCR<PLLON>

Note2: Switching of clock gear is executed when a value is written to the CGSYSCR<GEAR[2:0]> register. The actual switching takes place after a slight delay.

Note3: Do not use 1 / 16 when "PLL =OFF" is used.

Note4: Do not use 1 / 16 when SysTick is used.

5.3.7. Prescaler Clock Control

Each peripheral function has a prescaler for dividing a clock. As the clock ΦT0 to be input to each prescaler, the fperiph clock specified in the CGSYSCR<FPSEL> can be divided according to the setting in the CGSYSCR<PRCK[2:0]>. After reset, fperiph /1 is selected as ΦT0.

Note: To use the clock gear, ensure that you make the time setting such that prescaler output ΦTn for each peripheral function is slower than fsys (ΦTn < fsys). Do not switch the clock gear while peripheral function such as timer counter is operating.

5.4. Modes and Mode Transitions

5.4.1. Mode Transitions

The NORMAL mode uses the high-speed clock for the system clock .

The IDLE and STOP modes can be used as the low-power consumption mode that enables to reduce power consumption by halting processor core operation.
Figure 5.3 shows mode transition diagram.
For a detail of sleep-on-exit and WFI instruction, refer to "Arm documentation set of the Arm Cortex-M series processor".

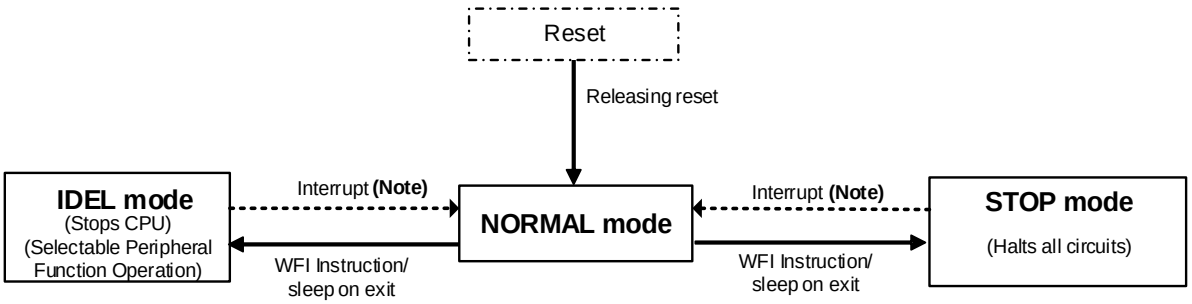


Figure 5.3 Mode Transition Diagram

Note: The warming-up is needed, when returning to NORMAL mode. The warming-up time must be set in NORMAL mode before changing to IDLE or STOP mode. Regarding warming-up time, refer to "5.6.6. Warming-up".

5.5. Operation Mode

As an operation mode, NORMAL is available. The features of NORMAL mode are described in the following section.

5.5.1. NORMAL mode

This mode is to operate the CPU core and the peripheral function by using the high-speed clock.
It is shifted to the NORMAL mode after reset.

5.6. Low Power Consumption Modes

The low-power consumption modes have IDLE and STOP modes. To shift to the low-power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction. When shifting to low-power consumption mode by WFI instruction, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. Refer to "6. Exceptions" for details.

Note1: TMPM370FYDFG/TMPM370FYFG do not offer any event for releasing the low-power consumption mode. Transition to the low-power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.

Note2: TMPM370FYDFG/TMPM370FYFG do not support the low-power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

The features of IDLE and STOP modes are described as follows.

5.6.1. IDLE Mode

Only the CPU is stopped in this mode. Each peripheral function has one bit in its control register for enabling or disabling operation in the IDLE mode. When the IDLE mode is entered, peripheral functions whose operations are disabled in the IDLE mode stop operation and hold the state at that time.
The following peripheral functions can be enabled or disabled in the IDLE mode. For setting details, refer to each peripheral function.

- 16-bit timer/event counter (TMRB)
- Serial channel (SIO/UART)
- Watchdog timer (WDT)
- Vector Engine (VE)
-

Note: WDT must be stopped before entering IDLE mode.

5.6.2. STOP mode

All the internal circuits including the internal oscillator are brought to a stop in the STOP mode.

By releasing the STOP mode, the device returns to the preceding mode of the STOP mode and starts operation.

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 5.2 shows the pin status in the STOP mode.

Table 5.2 Pin States in the STOP mode

	Pin name	I/O	<DRVE> = 0	<DRVE> = 1
Not port	X1	Input only	-	
	X2	Output only	"High" level output	
	$\overline{\text{RESET}}$, MODE	Input only	✓	
	VOUT15, VOUT3	Output only	✓	
Port	TMS TCK TDI $\overline{\text{TRST}}$	Input	✓	
	TDO/SWV	Output	Enabled when data is valid. Disabled when data is invalid.	
	SWCLK	Input	✓	
	SWDIO	Input	✓	
		Output	Enabled when data is valid. Disabled when data is invalid.	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	Output	✓	
	UO0,1, VO0,1, WO0,1 XO0,1, YO0,1, ZO0,1	Output	-	✓
	INT0, INT1, INT2, INT3, INT4, INT5, INT6, INT7, INT8, INT9, INTA, INTB, INTC, INTD, INTE, INTF	Input	✓	
	Other pins of peripheral functions except above or the ports that are used as general purpose ports.	Input	-	✓
		Output	-	✓

✓: Input or output enabled., - : Input or output disabled.

5.6.3. Low power Consumption Mode Setting

The low-power consumption mode is specified by the setting of the standby control register CGSTBYCR<STBY[2:0]>.

Table 5.3 shows the mode setting in the <STBY[2:0]>.

Table 5.3 Low-power consumption mode setting

Mode	CGSTBYCR <STBY[2:0]>
STOP	001
IDLE	011

Note: Do not set any value other than the above.

5.6.4. Operational Status in Each Mode

Table 5.4 shows the operational status in each mode.

For I/O port, "✓" and "-" indicate that input/output is enabled and disabled respectively. For other functions, "✓" and "-" indicate that clock is supplied and is not supplied respectively.

Table 5.4 Operational Status in Each Mode

Peripheral function	NORMAL	IDLE	STOP
Processor core	✓	-	-
I/O port	✓	✓	- (Note2)
PMD	✓	✓	-
ENC	✓	✓	-
OFD	✓	✓	-
ADC	✓	✓	-
VE	✓	ON/OFF selectable for each module	-
SIO/UART	✓		-
TMRB	✓		-
WDT	✓		-
AMP/CMP	✓	✓	✓ (Note3)
VLTD	✓	✓	✓ (Note3)
POR	✓	✓	✓ (Note3)
CG	✓	✓	-
PLL	✓	✓	-
High-speed oscillator (f_c)	✓	✓	-

Note1: ✓: Can operate in the target mode, - : The clock to the module automatically stops when the target mode is entered.

Note2: It depends on CGSTBYCR<DRVE>.

Note3: The peripheral functions are not stopped even though the clock is halted.

5.6.5. Releasing the Low Power Consumption Mode

The low-power consumption mode can be released by an interrupt request, NMI or reset. The release source that can be used is determined by the low-power consumption mode selected.

Details are shown in Table 5.5.

Table 5.5 Release Source in Each Mode

Low power consumption mode			IDLE (programmable)	STOP
Release source	Interrupt	INT0 to F (Note1)	✓	✓
		INTRX0 to 3, INTTX0 to 3	✓	-
		INTVCNA, INTVCNB	✓	-
		INTEMG0, 1	✓	-
		INTOVV0, 1	✓	-
		INTADAPDA, INTADBPDA	✓	-
		INTADAPDB, INTADBPDB	✓	-
		INTTB00, 10, 20, 30, 40, 50, 60, 70	✓	-
		INTTB01, 11, 21, 31, 41, 51, 61, 71	✓	-
		INTPMD0, 1	✓	-
		INTCAP00, 10, 20, 30, 40, 50, 60, 70	✓	-
		INTCAP01, 11, 21, 31, 41, 51, 61, 71	✓	-
		INTADACPA, INTADBCPA	✓	-
		INTADACPB, INTADBCPB	✓	-
		INTADASFT, INTADBSFT	✓	-
		INTADATMR, INTADBTMR	✓	-
		INTENC0, 1	✓	-
	SysTick interrupt		✓	-
	NMI (INTWDT)		-	-
	RESET ($\overline{\text{RESET}}$ pin)		✓	✓

Note1: ✓: Starts the interrupt handling after the mode is released. (The RESET initializes the MCU), - : Unavailable

Note2: To release the low-power consumption mode by using the level mode interrupt, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt handling from starting properly.

Note3: For shifting to the low-power consumption mode, set the CPU to prohibit all the interrupts other than the return source. If it is not prohibited, it may be released by an interrupt other than the return source.

Note4: Refer to "5.6.6. Warming-up " about warming-up time for returning from each mode.

- Release by interrupt request
  To release the low-power consumption mode by an interrupt, the CPU must be set in advance to detect the interrupt. In addition to the setting in the CPU, the clock generator must be set to detect the interrupt to be used to release the STOP modes.

- Release by Non-Maskable Interrupt (NMI)
  There is a watchdog timer interrupt (INTWDT) as a non-maskable interrupt source. INTWDT can only be used in the IDLE mode.

Note: Note that the WDT cannot be cleared by the CPU operation in IDLE mode.

- Release by reset
- All low-power consumption modes can be released by reset from the $\overline{\text{RESET}}$ pin. After that, the mode switches to NORMAL mode and all the registers are initialized as is the case with normal reset.

- Release by SysTick interrupt
  SysTick interrupt can only be used in IDLE mode.

Refer to "6. Exceptions" for detail.

5.6.6. Warming-up

Mode transition may require the warming-up so that the internal oscillator provides stable oscillation.
In the mode transition from STOP to the NORMAL, the warming-up counter is activated automatically. And then the system clock output is started after the elapse of configured warming-up time. It is necessary to set a oscillator to be used for warming-up in the CGOSCCR<WUPSEL1> (Note1) and to set a warming-up time in the CGOSCCR<WUODR> before executing the instruction to enter the STOP mode.

Note1: For , always set CGOSCCR<WUPSEL1> to "0".

Note2: In STOP modes, the PLL is disabled. When returning from STOP mode, configure the warming-up time in consideration of the stability time of the PLL and the internal oscillator. It takes approximately 200μs for the PLL to be stabilized.

Note3: It is not necessary to set CGOSCCR<WUEON> , when returning from the low consumption mode that automatic warming-up.

Table 5.6 shows whether the warming-up setting of each mode transition is required or not.

Table 5.6 Warming-up setting in mode transition

Mode transition	Warming-up setting
NORMAL → IDLE	Not required
NORMAL → STOP	Not required
IDLE → NORMAL	Not required
STOP → NORMAL	Required

5.6.7. Clock Operation in Mode Transition

The clock operation in mode transition is described below.

5.6.7.1. Transition of operation modes: NORMAL → STOP → NORMAL

When returning to the NORMAL mode from the STOP mode, the warming-up is activated automatically. It is necessary to set the warming-up time before entering the STOP mode.

Returning to the NORMAL mode by reset does not induce the automatic warming-up. Keep the reset signal asserted until the oscillator operation becomes stable.

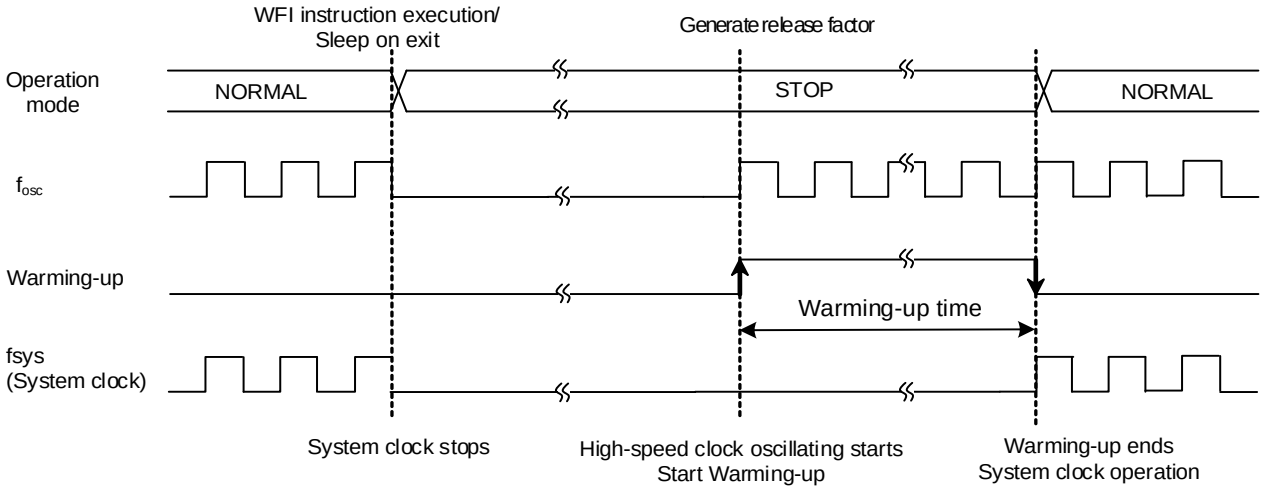


Figure 5.4 Transition of operation modes: NORMAL → STOP → NORMAL

6. Exceptions

This chapter describes features, types and handling of exceptions.

Exceptions have close relation to the CPU core. Refer to "Arm documentation set for the Arm Cortex-M3 processor" if needed.

6.1. Outline

The exceptions require to stop executing process and to branch other process on CPU.

There are two types of exceptions: those that are generated when some error conditions occur or when an instruction to generate an exception is executed; and those that are generated by hardware, such as an interrupt request signal from the external interrupt pins or peripheral functions.

All exceptions are handled by the Nested Vectored Interrupt Controller (NVIC) in the CPU according to the respective priority levels. When an exception occurs, the CPU stores the current state to the stack and branches to the corresponding interrupt service routine. After completion of the interrupt service routine, the state stored to the stack is automatically restored.

6.1.1. Types

There are the following types of exceptions.

Refer to "Arm documentation set for the Arm Cortex-M3 processor" for details.

- RESET
- Non-Maskable Interrupt (NMI)
- Hard Fault
- Memory Management
- Bus Fault
- Usage Fault
- SVCall (Supervisor Call)
- Debug Monitor
- PendSV
- SysTick
- External Interrupt

6.1.2. Handling Flowchart

The outline of handling exceptions and interrupts are shown below.

In a following description, ☐ means process by hardware, and ☐ means process by software.

The details for each process are described in the following sections.

Processing	Description	Refer to
Detecting exception by CG or CPU	CG or CPU detects the exception.	**Section 6.1.2.1**
↓		
Handling exception by CPU	CPU handles the exception process.	**Section 6.1.2.2**
↓		
Branching to interrupt service routine	CPU branches to the interrupt service routine which corresponds detected exception.	
↓		
Executing interrupt service routine	The interrupt service routine is executed.	**Section 6.1.2.3**
↓		
Exiting exception	CPU branches to another interrupt service routine or returns to the previous program.	**Section 6.1.2.4**

Figure 6.1 Handling Flowchart

6.1.2.1. Exception Request and Detection

(1) Exception Request Occurrence

An exception occurs by instruction execution by the CPU, memory accesses, and external interrupts from external interrupt pins or peripheral functions.

An exception request occurs when the CPU executes an instruction that causes an exception or when an error condition occurs during instruction execution.

An exception request also occurs by an instruction fetch from the Execute Never region or an access violation to the Fault region.

An exception request of external interrupts occurs from an external interrupt pins or peripheral functions. When external interrupt pins are used for releasing the low-power consumption mode, the CG must be set properly. For details, refer to "6.4. Interrupts".

(2) Exception Detection

When multiple exceptions occur simultaneously, the CPU takes the exception with the highest priority according to priority setting.

Table 6.1 shows the priority of exceptions. "Configurable" means that a priority level to that exception can be set. Memory Management, Bus Fault, and Usage Fault exceptions can be enabled or disabled. When a disabled exception occurs, it is handled as Hard Fault.

Table 6.1 Exception Types and Priority

No.	Exception	Priority	Factor
1	RESET	-3 (highest)	$\overline{\text{RESET}}$ pin, WDT, POR, VLTD, OFD, <SYSRESETREQ>
2	Non-maskable interrupt	-2	WDT
3	Hard fault	-1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4	Memory management	Configurable	Exception from the Memory Protection Unit (MPU) (Note1) Instruction fetch from the Execute Never (XN) region
5	Bus fault	Configurable	Access violation to the Hard Fault region of the memory map
6	Usage fault	Configurable	Undefined instruction execution or other faults related to instruction execution
7 to 10	Reserved	-	-
11	SVCall	Configurable	System service call with SVC instruction
12	Debug monitor	Configurable	Debug monitor when the CPU is not faulting
13	Reserved	-	-
14	PendSV	Configurable	Pendable request for system service
15	SysTick	Configurable	Notification from system timer
16 to	External interrupt	Configurable	External interrupt pins or peripheral functions (Note2)

Note1: TMPM370FYDFG/TMPM370FYFG do not contain the MPU.

Note2: In each product, external interrupts have deferent sources and numbers. For concrete sources and numbers, refer to "6.4.4. List of Interrupt Sources for Each Product".

(3) Priority setting

- Priority level

  The external interrupt priority is set to the interrupt priority register and other exceptions are set to corresponding <PRI_n[7:0]> in the system handler priority register.

  The configuration <PRI_n[7:0]> can be changed, and the number of bits required for setting the priority varies from 3 bits to 8 bits depending on products. Thus, the range of priority values you can specify is different depending on products.

  In the case of 8-bit configuration, the priority can be configured in the range from 0 to 255. The highest priority is "0". When multiple factors are set to the same priority, a factor which has the smaller number has the higher priority.

Note: The number of bits in which priority level is set in <PRI_n[7:0]> is 3 with TMPM370FYDFG/TMPM370FYFG.

- Priority grouping

  The priority can be split into groups. By setting the <PRIGROUP[2:0]> of the application interrupt and reset control register, <PRI_n[7:0]> can be divided into the pre-emption priority and the sub priority.

  Each priority of exceptions is compared by the pre-emption priority. When each priority is the same, then it is compared with the sub priority. When each sub priority is the same, an exception which has the smaller number has the higher priority.

  Table 6.2 shows the priority group setting. The pre-emption priority and the sub priority in the table are the number in the case that <PRI_n[7:0]> is defined as an 8-bit configuration.

Table 6.2 Priority Grouping Setting

<PRIGROUP[2:0]> setting	<PRI_n[7:0]>		Number of pre-emption priorities	Number of sub priorities
	Pre-emption field	Sub priority field		
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
111	None	[7:0]	1	256

Note: When the configuration of <PRI_n[7:0]> is less than 8 bits, the lower bit is "0".
For the example, in the case of 3-bit configuration, the priority is set as <PRI_n[7:5]> and <PRI_n[4:0]> is "0_0000".

6.1.2.2. Exception Handling and Branch to the Interrupt Service Routine (Pre-emption)

When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "Pre-emption".

(1) Stacking Registers

When the CPU detects an exception, it pushes the contents of the eight registers to the stack in the following order:

- Program Status Register (xPSR)
- Program Counter (PC)
- Link Register (LR)
- r12
- r3 to r0

The SP is decremented by eight words after the completion of the stack push. The following shows the state of the stack after the register contents have been pushed.

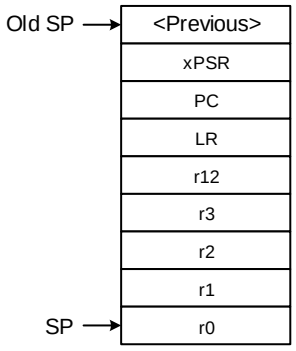


Figure 6.2 Stack After Completion of Stack Push

(2) Fetching Interrupt Service Routine

The CPU fetches the instruction of interrupt service routine with stacking registers.

Prepare a vector table containing the top addresses of interrupt service routines for each exception. After reset, the vector table is located at address 0x0000_0000 in the Code area. By setting the Vector Table Offset Register, the vector table can be placed at any address in the Code or SRAM space.

The vector table should also contain the initial value of the main stack.

(3) Late-arriving

When the CPU detects a higher priority exception before executing the interrupt service routine for a previous exception, the CPU handles the higher priority exception first. This is called "Late-arriving".

When a Late-arriving exception causes, the CPU fetches an instruction of the new detecting exception, branches to the corresponding interrupt service routine. But the CPU does not newly push the register contents to the stack.

(4) Configuration of Vector Table

The configuration of vector table is shown in the below.

It is necessary to set the first four words (stack top address, reset interrupt service routine's address, NMI interrupt service routine's address, and Hard Fault interrupt service routine's address). Set interrupt service routine's addresses for other exceptions if necessary.

Table 6.3 Configuration of Vector Table

Offset	Exception	Contents	Setting
0x00	RESET	Initial value of the main stack	Required
0x04	RESET	Interrupt service routine's address	Required
0x08	Non-Maskable interrupt (NMI)	Interrupt service routine's address	Required
0x0C	Hard Fault	Interrupt service routine's address	Required
0x10	Memory Management	Interrupt service routine's address	Option
0x14	Bus Fault	Interrupt service routine's address	Option
0x18	Usage Fault	Interrupt service routine's address	Option
0x1C to 0x28	Reserved	-	-
0x2C	SVCall	Interrupt service routine's address	Option
0x30	Debug Monitor	Interrupt service routine's address	Option
0x34	Reserved	-	-
0x38	PendSV	Interrupt service routine's address	Option
0x3C	SysTick	Interrupt service routine's address	Option
0x40	External interrupt	Interrupt service routine's address	Option

6.1.2.3. Executing Interrupt Service Routine

An interrupt service routine performs necessary processing for the corresponding exception. The interrupt service routines must be prepared by the user.

An interrupt service routine may need to include code for clearing the interrupt request so that the same interrupt will not occur again upon return to normal program.

For details about interrupt handling, refer to "6.4. Interrupts".

When a higher priority exception occurs during interrupt service routine execution for the current exception, the CPU abandons the current executing interrupt service routine and processes an interrupt routine for the newly detected exception.

6.1.2.4. Exiting Exception

(1) Actions After Returning from Interrupt Service Routine

An action is depended on the state after returning from an interrupt service routine.

- Tail-chaining
  In the case that a stacked exception exists, when there are no pending exceptions or a stacked exception has higher priority than all pending exceptions, the CPU returns to the interrupt service routine of the stacked exception.
  In this case, the CPU skips restoring registers from stack and storing registers to stack. This is called "Tail-chaining".
- Returning to Pending Interrupt Service Routine
  When there are no stacked exceptions or a priority of an pending exception is higher than the stacked exception in spite of existing the stacked exception, the CPU returns to the last pending interrupt service routine.
- Returning to Previous Program
  When no stacked or pending exceptions, the CPU returns to the previous program.

(2) Returning Operation

When returning from an interrupt service routine, the CPU executes the following operations:

- Restoring Registers
  Restore registers (xPSR、PC、LR、r12、r3 to r0), adjust the SP.
- Load current active interrupt number
  Load the current active interrupt number from the stacked xPSR. The CPU uses this interrupt number to control returning to which interrupt.
- Select SP
  When returning to an exception (Handler mode), the SP_main is used as the SP. When returning to the Thread mode, the SP_main or SP_process is used as the SP.

6.2. RESET Exceptions

There are the following six sources for RESET exception.

Refer to CGRSTFLG of CG register to confirm the source of the RESET.

- RESET Exception by $\overline{\text{RESET}}$ Pin
  A RESET exception occurs after $\overline{\text{RESET}}$ pin changes from "Low" level to "High" level.
- RESET Exception by POR
  The POR has the function which generates RESET exception. Refer to "15. Power-on Reset circuit (POR)" for details.
- RESET Exception by VLTD
  The VLTD has the function which generates RESET exception. Refer to "16. Voltage Detection circuit (VLTD)" for details.
- RESET Exception by OFD
  The OFD has the function which generates RESET exception. Refer to "17. Oscillation Frequency Detector Circuit (OFD)" for details.
- RESET Exception by WDT
  The WDT has the function which generates RESET exception. Refer to "18. Watchdog Timer (WDT)" for details.
- RESET Exception by <SYSRESETREQ>
  RESET exception occurs by setting <SYSRESETREQ> in application interrupt and reset control register of the NVIC.

6.3. SysTick

SysTick provides interrupt functions of using the CPU's system timer.

When setting a value to SysTick Reload Value Register and SysTick function is enabled by the SysTick Control and Status Register, the timer is loaded with the value set in the Reload Value Register and begins counting down. When the timer counts to "0", a SysTick exception occurs. When pending exceptions, it can be known that the timer counts to "0" by checking a flag.

The SysTick Calibration Value Register holds a reload value for counting 10 ms with the system timer. The count clock cycle varies with each product, and so the value set in the SysTick Calibration Value Register also varies with each product.

Note: TMPM370FYDFG/TMPM370FYFG use the clock which is obtained by dividing f_{OSC} (a clock selected by CGOSCCR<OSCSEL>) by 32 as an external reference clock.

6.4. Interrupts

This section describes routes where a signal travels, sources and required settings of interrupts.

6.4.1. Non-Maskable Interrupt (NMI)

For sources of non-maskable interrupt, refer to Table 6.4 in "6.4.4. List of Interrupt Sources for Each Product".

6.4.2. Maskable Interrupt

For sources of maskable interrupt, refer to Table 6.5 in "6.4.4. List of Interrupt Sources for Each Product".

6.4.3. Interrupt Request

The CPU is notified of interrupt request by the signal from each interrupt source.

The CPU sets priority on interrupt, and generates interrupt with the highest priority.

6.4.3.1. Interrupt Request Route

The interrupt request route is shown in Figure 6.3.

The interrupt, of external interrupts, request signal from peripheral functions which is not used for releasing the low-power consumption mode is input to the CPU (route (1)).

The interrupt request signal (route (2)) from peripheral functions which is used for releasing the low-power consumption mode is input to the CG, and is input to the CPU via a circuit for releasing the low-power consumption mode in the CG (route (4)).

The interrupt, of external interrupts, request signal from external interrupt pins is input to the CG (route (3)). The signal which is not via a circuit for releasing the low-power consumption mode (route (6)) and the signal which is via a circuit for releasing the low-power consumption mode (route (7)) are selected by CGIMCGn<INTmEN>. The selected signal is input to the CPU (route (5)).

When an external interrupt pin is used for releasing the low-power consumption mode, route (7) is selected. When not, route (6) is selected.

Note: "n" means a suffix of a register name. "m" means a suffix of a bit symbol.

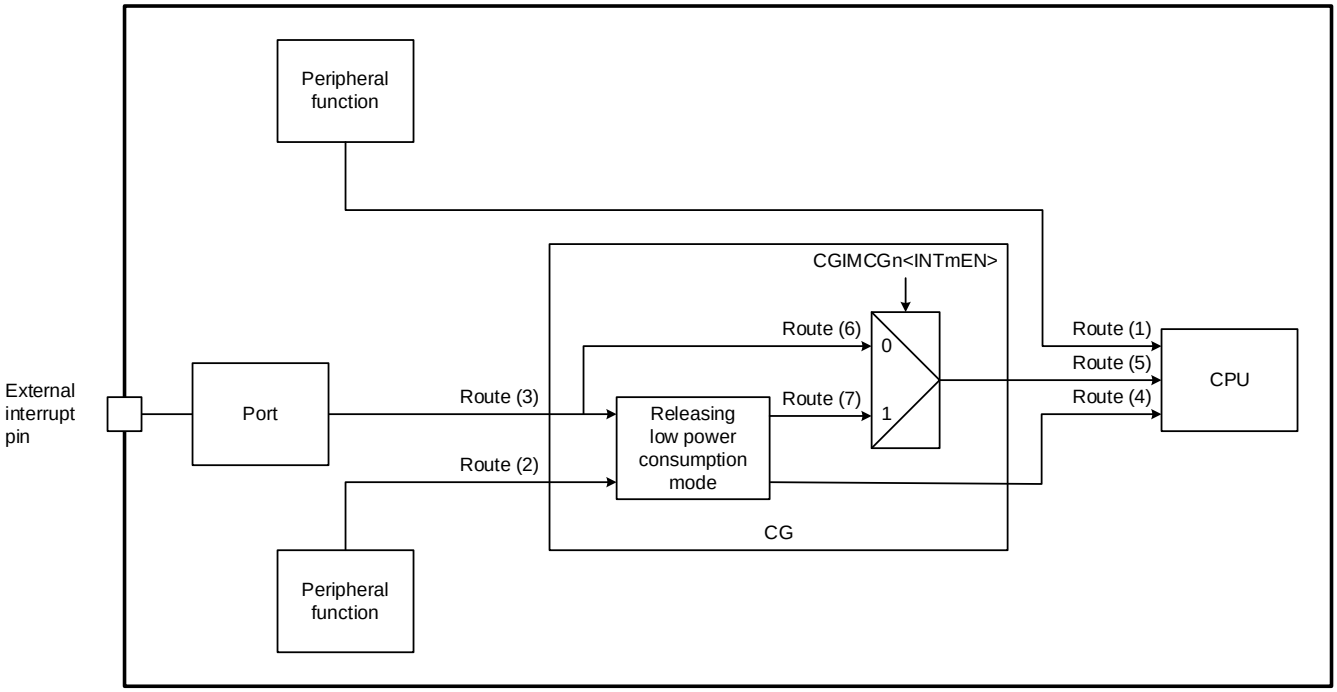


Figure 6.3 Interrupt Request Route

6.4.3.2. Generation of Interrupt Requests

An interrupt request is generated from an external interrupt pin or peripheral function assigned as an interrupt source or by setting the NVIC's Interrupt Set-Pending Register.

- From External Interrupt Pin
  Set the port control register so that the external interrupt pin can be used as an interrupt function.
- From peripheral function
  Set the peripheral function to make it possible to output interrupt requests.
  Refer to the section of each peripheral function for details.
- Forced Pending Interrupt
  An interrupt request can be generated by setting the corresponding bit of the Interrupt Set-Pending Register.

6.4.3.3. Monitoring Interrupt Detection Level

The CG has a monitor flag of the interrupt detection level.
CGIMCGn<EMSTm[1:0]> is read to monitor an interrupt detection level.

Note: "n" means a suffix of a register name. "m" means a suffix of a bit symbol.

6.4.3.4. Transmission of Interrupt Requests

The interrupt request which is not used as the source for releasing the low-power consumption mode is directly input to the CPU.
The interrupt request from a peripheral functions and/or external interrupt pin which are used as the sources for releasing the low-power consumption mode is input to the CPU via the CG. Therefore, the CG must be set.

6.4.3.5. Caution when Using External Interrupt Pins

When using external interrupt pins, take following notices not to generate unexpected interrupts.

When disable input mode in Port function, the signal from an external interrupt pin is "High" level.
Also, When no using an external interrupt pin for releasing the low-power consumption mode (route (6) in "Figure 6.3 Interrupt Request Route"), the input signal from an external interrupt pin is directly input to CPU.
The CPU recognizes "High" level as an interrupt request.
Therefore, an interrupt request occurs when enabling interrupt with disabling input mode.
When an external interrupt pin is not used for releasing low-power consumption mode, interrupt must be enabled after enabling input mode while an external interrupt pin is "Low" level.

Note1: For setting Port, refer to "8. Input/output Port".

Note2: "n" means a suffix of a register name. "m" means a suffix of a bit symbol.

6.4.4. List of Interrupt Sources for Each Product

The source of non-maskable interrupt is shown in Table 6.4.

Table 6.4 List of Interrupt Source (Non-maskable Interrupt)

Interrupt Number	Interrupt source		NMI Generation factor flag
1	INTWDT	WDT interrupt	CGNMIFLG<MNIFLG0>

The sources of maskable interrupts are shown in Table 6.5.

Table 6.5 List of Interrupt Source

Interrupt Number	Interrupt source		CG interrupt mode control register
0	INT0	External interrupt pin 0	CGIMCGA
1	INT1	External interrupt pin 1	CGIMCGA
2	INT2	External interrupt pin 2	CGIMCGA
3	INT3	External interrupt pin 3	CGIMCGA
4	INT4	External interrupt pin 4	CGIMCGB
5	INT5	External interrupt pin 5	CGIMCGB
6	INTRX0	SIO/UART ch0 serial reception	
7	INTTX0	SIO/UART ch0 serial transmission	
8	INTRX1	SIO/UART ch1 serial reception	
9	INTTX1	SIO/UART ch1 serial transmission	
10	INTVCNA	Vector engine interrupt A	
11	INTVCNB	Vector engine interrupt B	
12	INTEMG0	PMD ch0 EMG interrupt	
13	INTEMG1	PMD ch1 EMG interrupt	
14	INTOVV0	PMD ch0 OVV interrupt	
15	INTOVV1	PMD ch1 OVV interrupt	
16	INTADAPDA	ADC unit A PMD ch0 trigger conversion completion	
17	INTADBPDA	ADC unit B PMD ch0 trigger conversion completion	
18	INTADAPDB	ADC unit A PMD ch1 trigger conversion completion	
19	INTADBPDB	ADC unit B PMD ch1 trigger conversion completion	
20	INTTB00	TMRB ch0 compare match 0/overflow	

Interrupt Number	Interrupt source		CG interrupt mode control register
21	INTTB01	TMRB ch0 compare match 1	
22	INTTB10	TMRB ch1 compare match 0/overflow	
23	INTTB11	TMRB ch1 compare match 1	
24	INTTB40	TMRB ch4 compare match 0/overflow	
25	INTTB41	TMRB ch4 compare match 1	
26	INTTB50	TMRB ch5 compare match 0/overflow	
27	INTTB51	TMRB ch5 compare match 1	
28	INTPMD0	PMD ch0 PWM interrupt	
29	INTPMD1	PMD ch1 PWM interrupt	
30	INTCAP00	TMRB ch0 input capture 0	
31	INTCAP01	TMRB ch0 input capture 1	
32	INTCAP10	TMRB ch1 input capture 0	
33	INTCAP11	TMRB ch1 input capture 1	
34	INTCAP40	TMRB ch4 input capture 0	
35	INTCAP41	TMRB ch4 input capture 1	
36	INTCAP50	TMRB ch5 input capture 0	
37	INTCAP51	TMRB ch5 input capture 1	
38	INT6	External interrupt pin 6	CGIMCGB
39	INT7	External interrupt pin 7	CGIMCGB
40	INTRX2	SIO/UART ch2 serial reception	
41	INTTX2	SIO/UART ch2 serial transmission	
42	INTADACPA	ADC unit A conversion monitoring function interrupt A	
43	INTADBCPA	ADC unit B conversion monitoring function interrupt A	
44	INTADACPB	ADC unit A conversion monitoring function interrupt B	
45	INTADBCPB	ADC unit B conversion monitoring function interrupt B	
46	INTTB20	TMRB ch2 compare match 0/overflow	
47	INTTB21	TMRB ch2 compare match 1	
48	INTTB30	TMRB ch3 compare match 0/overflow	
49	INTTB31	TMRB ch3 compare match 1	
50	INTCAP20	TMRB ch2 input capture 0	
51	INTCAP21	TMRB ch2 input capture 1	

Interrupt Number	Interrupt source		CG interrupt mode control register
52	INTCAP30	TMRB ch3 input capture 0	
53	INTCAP31	TMRB ch3 input capture 1	
54	INTADASFT	ADC unit A software trigger start conversion completion	
55	INTADBSFT	ADC unit B software trigger start conversion completion	
56	INTADATMR	ADC unit A timer trigger conversion completion	
57	INTADBTMR	ADC unit B timer trigger conversion completion	
58	INT8	External interrupt pin 8	CGIMCGC
59	INT9	External interrupt pin 9	CGIMCGC
60	INTA	External interrupt pin A	CGIMCGC
61	INTB	External interrupt pin B	CGIMCGC
62	INTENC0	Encoder input ch0	
63	INTENC1	Encoder input ch1	
64	INTRX3	SIO/UART ch3 serial reception	
65	INTTX3	SIO/UART ch3 serial transmission	
66	INTTB60	TMRB ch6 compare match 0/overflow	
67	INTTB61	TMRB ch6 compare match 1	
68	INTTB70	TMRB ch7 compare match 0/overflow	
69	INTTB71	TMRB ch7 compare match 1	
70	INTCAP60	TMRB ch6 input capture 0	
71	INTCAP61	TMRB ch6 input capture 1	
72	INTCAP70	TMRB ch7 input capture 0	
73	INTCAP71	TMRB ch7 input capture 1	
74	INTC	External interrupt pin C	CGIMCGD
75	INTD	External interrupt pin D	CGIMCGD
76	INTE	External interrupt pin E	CGIMCGD
77	INTF	External interrupt pin F	CGIMCGD

6.4.5. Interrupt Detection Level

When using an interrupt via the CG, an interrupt detection level is selected from "Low" level, "High" level, rising edge, falling edge or both edges by CG Interrupt Mode Control Register CGIMCGn<EMCGm>. A signal of the detected interrupt is output to the CPU as "High" level signal.

An interrupt signal, which is directly input to the CPU, from each peripheral function is output as "High" level pulse.

The CPU recognizes "High" level of an interrupt signal as an interrupt source.

Note: "n" means a suffix of a register name. "m" means a suffix of a bit symbol.

6.4.5.1. Notice When Releasing Low-power Consumption Mode

When an external interrupt is used for releasing low-power consumption mode, the following settings are needed.

- Interrupt detection level is set by CGIMCGn<EMCGm>.
- CGIMCGn<INTmEN> is set to "1".

Note: "n" means a suffix of a register name. "m" means a suffix of a bit symbol.

6.4.6. Handling Method

6.4.6.1. Handling Flowchart

The outline of handling exceptions and interrupts are shown bellows.

In a follow description, ☐ means process by hardware, and ☐ means process by software.

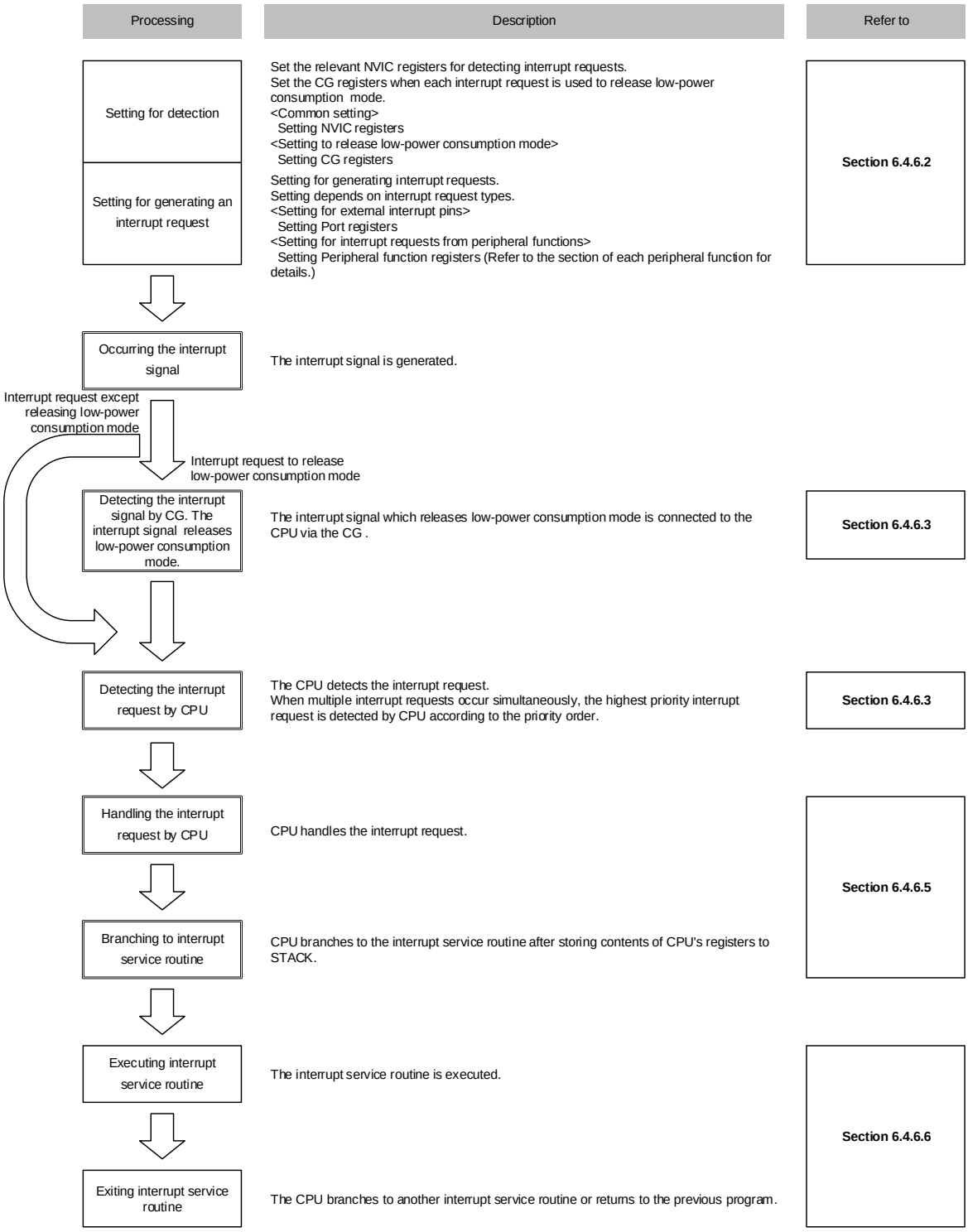


Figure 6.4 Handling Flowchart

6.4.6.2. Preparation

When preparing for an interrupt, take an attention to the order of configuration to avoid any unexpected interrupt on the way.

Initiating an interrupt or changing its configuration must be implemented in the following order basically.

Disable the interrupt by the CPU. Configure from the farthest route from the CPU. Then enable the interrupt by the CPU.

To configure the CG, first, configure the condition. Secondly, clear the data related with the interrupt in the CG to avoid any unexpected interrupt and then enable the interrupt.

The followings are procedure of interrupt handling and the description for each step.

(1) Disabling CPU's interrupt

(2) Setting the CPU registers for interrupt

(3) Preparation of an interrupt source (1) (Interrupt from external interrupt pins)

(4) Preparation of an interrupt source (2) (Interrupt from peripheral functions)

(5) Preparation of an interrupt source (3) (Interrupt Set-Pending Registers)

(6) Configuring the CG

(7) Enabling CPU's interrupt

(1) Disabling CPU's interrupt

To make the CPU not accept any interrupt, write "1" to the corresponding bit of the PRIMASK. All interrupts and exceptions other than non-maskable interrupts and hard faults are masked.

Use "MSR" instruction to set this register.

Table 6.6 Setting Interrupt Mask Register

Interrupt mask register
PRIMASK ← "1" (Interrupt disabled)

Note1: PRIMASK register cannot be modified by the user access level.

Note2: When PRIMASK is set to "1" and a fault causes, it is handled as a hard fault.

(2) Setting the CPU registers for interrupt

A priority level is set by <PRI_n[7:5]> in Interrupt Priority Register of the NVIC register.

This register is assigned to each interrupt source. The number is depended on each product.

When bit width of this register is 8 bits, a priority level can be assigned from 0 to 255. Priority level 0 is the highest priority level. When multiple sources have the same priority, the smallest-numbered interrupt source has the highest priority.

A grouping priority is also set by using <PRIGROUP[2:0]> in the Application Interrupt and Reset Control Register.

Table 6.7 Setting NVIC registers

NVIC registers		
<PRI_n[7:5]>	←	Priority
<PRIGROUP[2:0]>	←	Group priority (if required)

Note: "n" means the exception number or interrupt number.

The bit width of Interrupt Priority Register is 3 bits on TMPM370FYDFG/TMPM370FYFG.

(3) Preparation of an interrupt source (1) (Interrupt from external interrupt pins)

When using an external interrupt pin, configure the setting of corresponding pin. To use as function pin, set PxFRn<PxmFn> to "1". And set PxIE<PxmIE> to "1" to use as input pin.

Table 6.8 Setting for External Interrupt Pin

Port setting registers		
PxFRn<PxmFn>	←	"1"
PxIE<PxmIE>	←	"1"

Note1: "x" means the Port number. "m" means the bit number. "n" means the function register number.

Note2: In the mode except STOP mode, when enabling input function by PxIE, the input of an external interrupt pin is enabled regardless of setting PxFRn. When setting an external interrupt pin, the unused external interrupt pin must not be enable to input mode. For details, refer to "6.4.3.5. Caution when Using External Interrupt Pins".

(4) Preparation of an interrupt source (2) (Interrupt from peripheral functions)

When using the interrupt of peripheral functions, setting is depend on peripheral function. Refer to the section of each peripheral function.

(5) Preparation of an interrupt source (3) (Interrupt Set-Pending Registers)

When an interrupt is generated by Interrupt Set-Pending Registers, set corresponding bit of it to "1".

Table 6.9 Setting for Interrupt Set-Pending Register

NVIC register		
<SETPEND>	←	"1"

Note: "<SETPEND>" means the corresponding bit of Interrupt Set-Pending Register.

(6) CG Setting

For an interrupt source to be used for releasing low-power consumption mode, set CGIMCGn<EMCGm[2:0]> of CG to the interrupt detection level. And set CMIMCGn<INTmEN> to low-power consumption mode releasing enabled.

Before enabling interrupt, interrupt requests are cleared by CGICRCG beforehand to avoid unexpected interrupt.

The held interrupt request can be cleared by writing corresponding value of an external interrupt to CGICRCG. For details, refer to "6.5.3.1. CGICRCG (CG Interrupt Request Clear Register)".

Table 6.10 Setting for CG

CG registers		
CGIMCGn<EMCGm>	←	Interrupt detection level
CGICRCG<ICRCG>	←	Clear the interrupt request
CGIMCGn<INTmEN>	←	"1" (Low-power consumption mode releasing enabled)

Note: "n" means a suffix of a register name. "m" means a suffix of a bit symbol.

(7) Enabling CPU's interrupt

Enable the interrupt by the CPU as shown below.

The pending interrupt request can be cleared by setting the corresponding bit of interrupt clear-pending register to "1". And interrupt can be enabled by setting the corresponding bit of interrupt set-enable register to "1".

These are assigned for each interrupt source bit by bit.

However, when an interrupt is generated by interrupt set-pending register, above operation is not required because interrupt source is cleared by setting the corresponding bit of interrupt clear-pending register to "1".

At last, PRIMASK is cleared to "0".

Table 6.11 Setting Enabling CPU's Interrupt

NVIC registers		
<CLRPEND>	←	"1"
<SETENA>	←	"1"
Interrupt mask register		
PRIMASK	←	"0" (Interrupt enabled)

Note1: "<CLRPEND>" means the corresponding bit of interrupt clear-pending register. "<SETENA>" means the corresponding bit of interrupt set-enable register.

Note2 : PRIMASK register cannot be modified by the user access level.

6.4.6.3. Detection (CG)

The signal which is used for releasing low-power consumption mode is generated according to the interrupt detection level set in the CG, and then it is input to the CPU. After the interrupt signal whose interrupt level is rising edge, falling edge or both edges outputs the interrupt signal of "High" level to the CPU from detecting by the CG until clearing the interrupt request by CGICRCG. Therefore, same interrupt occurs again after exiting an interrupt service routine without clearing interrupt request. To avoid generating same interrupt, clear the interrupt request in an interrupt service routine.

And "High" level or "Low" level interrupt signal is recognized as no interrupt when the level is changed from the set interrupt detection level. Therefore, the level of interrupt signal must be hold until interrupt detection.

6.4.6.4. Detection (CPU)

The highest priority interrupt request is detected by the CPU according to the priority order.

6.4.6.5. CPU Processing

When detecting an interrupt, the CPU stores the contents of xPSR, PC, LR, r12 and r0 to r3 to the stack. Then, the CPU branches to an interrupt service routine.

6.4.6.6. Processing in Interrupt Service Routine (Clearing Interrupt Source)

An interrupt service routine requires specific programming according to the application to be used. This section describes what is recommended at the interrupt service routine programming and how the interrupt source is cleared.

(1) Processing in Interrupt Service Routine

An interrupt service routine normally stores the content of the required registers to the stack and handles an interrupt processing. The Cortex-M3 core automatically stores the contents of xPSR, PC, LR, r12 and r0 to r3 to the stack. No extra programming is required for them.

Store the contents of other registers if needed.

Interrupt requests with higher priority and exceptions such as non-maskable interrupt (NMI) are accepted even when an interrupt service routine is being executed. It is recommend that the contents of general-purpose registers that might be rewritten be stored.

(2) Clearing an interrupt source

An interrupt source is used for releasing low-power consumption mode must be cleared by CGICRCG.

When an interrupt detection level of an external interrupt is rising edge, falling edge or both edges, the held interrupt source is cleared by writing CGICRCG to the corresponding value. Therefore, the set interrupt detection level is detected again when it is generated.

When an interrupt detection level of an external interrupt is "High" or "Low" level, an interrupt source is kept continuously until an interrupt signal is cleared. Therefore, an interrupt signal must be cleared. The interrupt source from the CG is cleared after clearing an interrupt signal.

6.5. Exception/Interrupt-Related Registers

6.5.1. Register List

The following table shows control registers and addresses.

6.5.1.1. NVIC Registers

Register name	Address (Base+)
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Interrupt Set-Enable Register 2	0x0104
Interrupt Set-Enable Register 3	0x0108
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Clear-Enable Register 3	0x0188
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Set-Pending Register 3	0x0208
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Clear-Pending Register 3	0x0288
Interrupt Priority Register	0x0400 to 0x0460
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18, 0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

6.5.1.2. CG Registers

Register name		Address (Base+)
CG Interrupt Request Clear Register	CGICRCG	0x0014
NMI Generation Factor Flag Register	CGNMIFLG	0x0018
Reset Flag Register	CGRSTFLG	0x001C
CG Interrupt Mode Control Register A	CGIMCGA	0x0020
CG Interrupt Mode Control Register B	CGIMCGB	0x0024
CG Interrupt Mode Control Register C	CGIMCGC	0x0028
CG Interrupt Mode Control Register D	CGIMCGD	0x002C
Reserved	-	0x0030 to 0x003F

Note: Access to the "Reserved" areas is prohibited.

6.5.2. NVIC Registers

6.5.2.1. SysTick Control and Status Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	COUNTFLAG
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	CLKSOURCE	TICKINT	ENABLE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:17	-	R	Read as "0".
16	COUNTFLAG	R/W	0: Timer not counted to "0" 1: Timer counted to "0" When this bit is "1", this indicates that the timer counts to "0" after the last reading. When reading any part of this register, this bit is cleared to "0".
15:3	-	R	Read as "0".
2	CLKSOURCE	R/W	0: External reference clock (f_{OSC} / 32) (Note) 1: CPU clock (fsys)
1	TICKINT	R/W	0: Do not pend SysTick 1: Pend SysTick
0	ENABLE	R/W	0: Disabled 1: Enabled When this bit is set to "1", a timer is loaded with the value of SysTick Reload Value Register and starts count down.

Note: TMPM370FYDFG/TMPM370FYFG use the clock divided fosc (the clock selected by CGOSCCR<OSCSEL>) by 32 as an external reference clock.

6.5.2.2. SysTick Reload Value Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RELOAD							
After reset	Undefined							
	15	14	13	12	11	10	9	8
bit symbol	RELOAD							
After reset	Undefined							
	7	6	5	4	3	2	1	0
bit symbol	RELOAD							
After reset	Undefined							

Bit	Bit Symbol	Type	Function
31:24	-	R	Read as "0".
23:0	RELOAD[23:0]	R/W	Reload value The value to load to SysTick Current Value Register when the timer counts to "0" is set.

6.5.2.3. SysTick Current Value Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CURRENT							
After reset	Undefined							
	15	14	13	12	11	10	9	8
bit symbol	CURRENT							
After reset	Undefined							
	7	6	5	4	3	2	1	0
bit symbol	CURRENT							
After reset	Undefined							

Bit	Bit Symbol	Type	Function
31:24	-	R	Read as "0".
23:0	CURRENT[23:0]	R	Current SysTick timer value
		W	Any value: Clear the Timer The timer is cleared to "0" by writing any value to <CURRENET[23:0]>. When the timer is cleared to "0", <COUNTFLAG> of SysTick Control and Status Register is also cleared.

6.5.2.4. SysTick Calibration Value Register

	31	30	29	28	27	26	25	24
bit symbol	NOREF	SKEW	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	TENMS							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TENMS							
After reset	0	0	0	0	1	0	0	1
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TENMS							
After reset	1	1	0	0	0	1	0	0

Bit	Bit Symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10ms. 1: Calibration value is not 10ms.
29:24	-	R	Read as "0".
23:0	TENMS[23:0]	R	Calibration value Reload value (0x9C4) to use counting for 10ms by an external reference clock. (Note)

Note: When SysTick is used as multi shot, use the value which is subtracted one from <TENMS[23:0]>.

6.5.2.5. Interrupt Control Registers

Following registers will be used to control each interrupt factor; Interrupt Set-Enable Register, Interrupt Clear-Enable Register, Interrupt Set-Pending Register, and Interrupt Clear-Pending Register.
Each bit corresponds to specified interrupt.

Note: The interrupt factors varies each products. For details, refer to "6.4.4. List of Interrupt Sources for Each Product".

(1) Interrupt Set-Enable Register

Each bit corresponds to the specified interrupt number.
It can enable interrupts and check if interrupts are enabled.
Writing "1" to a bit in this register enables the corresponding interrupt.
Writing "0" has no effect.
Read the enable status of the corresponding interrupts.
Writing "1" to a corresponding bit in the Interrupt Clear-Enable Register clears the bit in this register.

(a) Interrupt Set-Enable Register 1

Bit	Bit Symbol	After reset	Type	Function
31	SETENA (Interrupt 31)	0	R/W	[Reading] 0: Interrupt is disabled. 1: Interrupt is enabled. [writing] (Note) 0: No effect 1: Enable interrupt
30	SETENA (Interrupt 30)	0		
29	SETENA (Interrupt 29)	0		
28	SETENA (Interrupt 28)	0		
27	SETENA (Interrupt 27)	0		
26	SETENA (Interrupt 26)	0		
25	SETENA (Interrupt 25)	0		
24	SETENA (Interrupt 24)	0		
23	SETENA (Interrupt 23)	0		
22	SETENA (Interrupt 22)	0		
21	SETENA (Interrupt 21)	0		
20	SETENA (Interrupt 20)	0		
19	SETENA (Interrupt 19)	0		
18	SETENA (Interrupt 18)	0		
17	SETENA (Interrupt 17)	0		
16	SETENA (Interrupt 16)	0		
15	SETENA (Interrupt 15)	0		
14	SETENA (Interrupt 14)	0		
13	SETENA (Interrupt 13)	0		
12	SETENA (Interrupt 12)	0		
11	SETENA (Interrupt 11)	0		
10	SETENA (Interrupt 10)	0		
9	SETENA (Interrupt 9)	0		
8	SETENA (Interrupt 8)	0		
7	SETENA (Interrupt 7)	0		
6	SETENA (Interrupt 6)	0		
5	SETENA (Interrupt 5)	0		
4	SETENA (Interrupt 4)	0		
3	SETENA (Interrupt 3)	0		
2	SETENA (Interrupt 2)	0		
1	SETENA (Interrupt 1)	0		
0	SETENA (Interrupt 0)	0		

Note: Write "0" to the bit whose bit symbol is "-".

(b) Interrupt Set-Enable Register 2

Bit	Bit Symbol	After reset	Type	Function
31	SETENA (Interrupt 63)	0	R/W	[Reading] 0: Interrupt is disabled. 1: Interrupt is enabled. [writing] (Note) 0: No effect 1: Enable interrupt
30	SETENA (Interrupt 62)	0		
29	SETENA (Interrupt 61)	0		
28	SETENA (Interrupt 60)	0		
27	SETENA (Interrupt 59)	0		
26	SETENA (Interrupt 58)	0		
25	SETENA (Interrupt 57)	0		
24	SETENA (Interrupt 56)	0		
23	SETENA (Interrupt 55)	0		
22	SETENA (Interrupt 54)	0		
21	SETENA (Interrupt 53)	0		
20	SETENA (Interrupt 52)	0		
19	SETENA (Interrupt 51)	0		
18	SETENA (Interrupt 50)	0		
17	SETENA (Interrupt 49)	0		
16	SETENA (Interrupt 48)	0		
15	SETENA (Interrupt 47)	0		
14	SETENA (Interrupt 46)	0		
13	SETENA (Interrupt 45)	0		
12	SETENA (Interrupt 44)	0		
11	SETENA (Interrupt 43)	0		
10	SETENA (Interrupt 42)	0		
9	SETENA (Interrupt 41)	0		
8	SETENA (Interrupt 40)	0		
7	SETENA (Interrupt 39)	0		
6	SETENA (Interrupt 38)	0		
5	SETENA (Interrupt 37)	0		
4	SETENA (Interrupt 36)	0		
3	SETENA (Interrupt 35)	0		
2	SETENA (Interrupt 34)	0		
1	SETENA (Interrupt 33)	0		
0	SETENA (Interrupt 32)	0		

Note: Write "0" to the bit whose bit symbol is "-".

(c) Interrupt Set-Enable Register 3

Bit	Bit Symbol	After reset	Type	Function
31	-	0	R/W	[Reading] 0: Interrupt is disabled. 1: Interrupt is enabled. [writing] (Note) 0: No effect 1: Enable interrupt
30	-	0		
29	-	0		
28	-	0		
27	-	0		
26	-	0		
25	-	0		
24	-	0		
23	-	0		
22	-	0		
21	-	0		
20	-	0		
19	-	0		
18	-	0		
17	-	0		
16	-	0		
15	-	0		
14	-	0		
13	SETENA(Interrupt 77)	0		
12	SETENA(Interrupt 76)	0		
11	SETENA(Interrupt 75)	0		
10	SETENA(Interrupt 74)	0		
9	SETENA(Interrupt 73)	0		
8	SETENA(Interrupt 72)	0		
7	SETENA(Interrupt 71)	0		
6	SETENA(Interrupt 70)	0		
5	SETENA(Interrupt 69)	0		
4	SETENA(Interrupt 68)	0		
3	SETENA(Interrupt 67)	0		
2	SETENA(Interrupt 66)	0		
1	SETENA(Interrupt 65)	0		
0	SETENA(Interrupt 64)	0		

Note: Write "0" to the bit whose bit symbol is "-".

(2) Interrupt Clear-Enable Register

Each bit corresponds to the specified interrupt number.

It can disable interrupts and check if interrupts are disabled.

Writing "1" to a bit in this register disables the corresponding interrupt.

Writing "0" has no effect.

Read the disable status of the corresponding interrupts.

(a) Interrupt Clear-Enable Register 1

Bit	Bit Symbol	After reset	Type	Function
31	CLRENA (Interrupt 31)	0	R/W	[Reading] 0: Interrupt is disabled. 1: Interrupt is enabled. [writing] (Note) 0: No effect 1: Disable interrupt
30	CLRENA (Interrupt 30)	0		
29	CLRENA (Interrupt 29)	0		
28	CLRENA (Interrupt 28)	0		
27	CLRENA (Interrupt 27)	0		
26	CLRENA (Interrupt 26)	0		
25	CLRENA (Interrupt 25)	0		
24	CLRENA (Interrupt 24)	0		
23	CLRENA (Interrupt 23)	0		
22	CLRENA (Interrupt 22)	0		
21	CLRENA (Interrupt 21)	0		
20	CLRENA (Interrupt 20)	0		
19	CLRENA (Interrupt 19)	0		
18	CLRENA (Interrupt 18)	0		
17	CLRENA (Interrupt 17)	0		
16	CLRENA (Interrupt 16)	0		
15	CLRENA (Interrupt 15)	0		
14	CLRENA (Interrupt 14)	0		
13	CLRENA (Interrupt 13)	0		
12	CLRENA (Interrupt 12)	0		
11	CLRENA (Interrupt 11)	0		
10	CLRENA (Interrupt 10)	0		
9	CLRENA (Interrupt 9)	0		
8	CLRENA (Interrupt 8)	0		
7	CLRENA (Interrupt 7)	0		
6	CLRENA (Interrupt 6)	0		
5	CLRENA (Interrupt 5)	0		
4	CLRENA (Interrupt 4)	0		
3	CLRENA (Interrupt 3)	0		
2	CLRENA (Interrupt 2)	0		
1	CLRENA (Interrupt 1)	0		
0	CLRENA (Interrupt 0)	0		

Note: Write "0" to the bit whose bit symbol is "-".

(b) Interrupt Clear-Enable Register 2

Bit	Bit Symbol	After reset	Type	Function
31	CLRENA (Interrupt 63)	0	R/W	[Reading] 0: Interrupt is disabled. 1: Interrupt is enabled. [writing] (Note) 0: No effect 1: Disable interrupt
30	CLRENA (Interrupt 62)	0		
29	CLRENA (Interrupt 61)	0		
28	CLRENA (Interrupt 60)	0		
27	CLRENA (Interrupt 59)	0		
26	CLRENA (Interrupt 58)	0		
25	CLRENA (Interrupt 57)	0		
24	CLRENA (Interrupt 56)	0		
23	CLRENA (Interrupt 55)	0		
22	CLRENA (Interrupt 54)	0		
21	CLRENA (Interrupt 53)	0		
20	CLRENA (Interrupt 52)	0		
19	CLRENA (Interrupt 51)	0		
18	CLRENA (Interrupt 50)	0		
17	CLRENA (Interrupt 49)	0		
16	CLRENA (Interrupt 48)	0		
15	CLRENA (Interrupt 47)	0		
14	CLRENA (Interrupt 46)	0		
13	CLRENA (Interrupt 45)	0		
12	CLRENA (Interrupt 44)	0		
11	CLRENA (Interrupt 43)	0		
10	CLRENA (Interrupt 42)	0		
9	CLRENA (Interrupt 41)	0		
8	CLRENA (Interrupt 40)	0		
7	CLRENA (Interrupt 39)	0		
6	CLRENA (Interrupt 38)	0		
5	CLRENA (Interrupt 37)	0		
4	CLRENA (Interrupt 36)	0		
3	CLRENA (Interrupt 35)	0		
2	CLRENA (Interrupt 34)	0		
1	CLRENA (Interrupt 33)	0		
0	CLRENA (Interrupt 32)	0		

Note: Write "0" to the bit whose bit symbol is "-".

(c) Interrupt Clear-Enable Register 3

Bit	Bit Symbol	After reset	Type	Function
31	-	0	R/W	[Reading] 0: Interrupt is disabled. 1: Interrupt is enabled. [writing] (Note) 0: No effect 1: Disable interrupt
30	-	0		
29	-	0		
28	-	0		
27	-	0		
26	-	0		
25	-	0		
24	-	0		
23	-	0		
22	-	0		
21	-	0		
20	-	0		
19	-	0		
18	-	0		
17	-	0		
16	-	0		
15	-	0		
14	-	0		
13	CLRENA (Interrupt 77)	0		
12	CLRENA (Interrupt 76)	0		
11	CLRENA (Interrupt 75)	0		
10	CLRENA (Interrupt 74)	0		
9	CLRENA (Interrupt 73)	0		
8	CLRENA (Interrupt 72)	0		
7	CLRENA (Interrupt 71)	0		
6	CLRENA (Interrupt 70)	0		
5	CLRENA (Interrupt 69)	0		
4	CLRENA (Interrupt 68)	0		
3	CLRENA (Interrupt 67)	0		
2	CLRENA (Interrupt 66)	0		
1	CLRENA (Interrupt 65)	0		
0	CLRENA (Interrupt 64)	0		

Note: Write "0" to the bit whose bit symbol is "-".

(3) Interrupt Set-Pending Register

Each bit corresponds to the specified interrupt number.

It can force interrupts into the pending state and checks which interrupts are currently pending.

Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled.

Writing "0" has no effect.

Read the current status of the corresponding interrupts; pending or not.

Writing "1" to a corresponding bit in the Interrupt Clear-Pending Register clears the bit in this register.

(a) Interrupt Set-Pending Register 1

Bit	Bit Symbol	After reset	Type	Function
31	SETPEND (Interrupt 31)	Undefined	R/W	[Reading] 0: Not pended 1: Pended [writing] (Note) 0: No effect 1: Pend interrupt
30	SETPEND (Interrupt 30)	Undefined		
29	SETPEND (Interrupt 29)	Undefined		
28	SETPEND (Interrupt 28)	Undefined		
27	SETPEND (Interrupt 27)	Undefined		
26	SETPEND (Interrupt 26)	Undefined		
25	SETPEND (Interrupt 25)	Undefined		
24	SETPEND (Interrupt 24)	Undefined		
23	SETPEND (Interrupt 23)	Undefined		
22	SETPEND (Interrupt 22)	Undefined		
21	SETPEND (Interrupt 21)	Undefined		
20	SETPEND (Interrupt 20)	Undefined		
19	SETPEND (Interrupt 19)	Undefined		
18	SETPEND (Interrupt 18)	Undefined		
17	SETPEND (Interrupt 17)	Undefined		
16	SETPEND (Interrupt 16)	Undefined		
15	SETPEND (Interrupt 15)	Undefined		
14	SETPEND (Interrupt 14)	Undefined		
13	SETPEND (Interrupt 13)	Undefined		
12	SETPEND (Interrupt 12)	Undefined		
11	SETPEND (Interrupt 11)	Undefined		
10	SETPEND (Interrupt 10)	Undefined		
9	SETPEND (Interrupt 9)	Undefined		
8	SETPEND (Interrupt 8)	Undefined		
7	SETPEND (Interrupt 7)	Undefined		
6	SETPEND (Interrupt 6)	Undefined		
5	SETPEND (Interrupt 5)	Undefined		
4	SETPEND (Interrupt 4)	Undefined		
3	SETPEND (Interrupt 3)	Undefined		
2	SETPEND (Interrupt 2)	Undefined		
1	SETPEND (Interrupt 1)	Undefined		
0	SETPEND (Interrupt 0)	Undefined		

Note: Write "0" to the bit whose bit symbol is "-".

(b) Interrupt Set-Pending Register 2

Bit	Bit Symbol	After reset	Type	Function
31	SETPEND (Interrupt 63)	Undefined	R/W	[Reading] 0: Not pended 1: Pended [writing] (Note) 0: No effect 1: Pend interrupt
30	SETPEND (Interrupt 62)	Undefined		
29	SETPEND (Interrupt 61)	Undefined		
28	SETPEND (Interrupt 60)	Undefined		
27	SETPEND (Interrupt 59)	Undefined		
26	SETPEND (Interrupt 58)	Undefined		
25	SETPEND (Interrupt 57)	Undefined		
24	SETPEND (Interrupt 56)	Undefined		
23	SETPEND (Interrupt 55)	Undefined		
22	SETPEND (Interrupt 54)	Undefined		
21	SETPEND (Interrupt 53)	Undefined		
20	SETPEND (Interrupt 52)	Undefined		
19	SETPEND (Interrupt 51)	Undefined		
18	SETPEND (Interrupt 50)	Undefined		
17	SETPEND (Interrupt 49)	Undefined		
16	SETPEND (Interrupt 48)	Undefined		
15	SETPEND (Interrupt 47)	Undefined		
14	SETPEND (Interrupt 46)	Undefined		
13	SETPEND (Interrupt 45)	Undefined		
12	SETPEND (Interrupt 44)	Undefined		
11	SETPEND (Interrupt 43)	Undefined		
10	SETPEND (Interrupt 42)	Undefined		
9	SETPEND (Interrupt 41)	Undefined		
8	SETPEND (Interrupt 40)	Undefined		
7	SETPEND (Interrupt 39)	Undefined		
6	SETPEND (Interrupt 38)	Undefined		
5	SETPEND (Interrupt 37)	Undefined		
4	SETPEND (Interrupt 36)	Undefined		
3	SETPEND (Interrupt 35)	Undefined		
2	SETPEND (Interrupt 34)	Undefined		
1	SETPEND (Interrupt 33)	Undefined		
0	SETPEND (Interrupt 32)	Undefined		

Note: Write "0" to the bit whose bit symbol is "-".

(c) Interrupt Set-Pending Register 3

Bit	Bit Symbol	After reset	Type	Function
31	-	Undefined	R/W	[Reading] 0: Not pended 1: Pended [writing] (Note) 0: No effect 1: Pend interrupt
30	-	Undefined		
29	-	Undefined		
28	-	Undefined		
27	-	Undefined		
26	-	Undefined		
25	-	Undefined		
24	-	Undefined		
23	-	Undefined		
22	-	Undefined		
21	-	Undefined		
20	-	Undefined		
19	-	Undefined		
18	-	Undefined		
17	-	Undefined		
16	-	Undefined		
15	-	Undefined		
14	-	Undefined		
13	SETPEND (Interrupt 77)	Undefined		
12	SETPEND (Interrupt 76)	Undefined		
11	SETPEND (Interrupt 75)	Undefined		
10	SETPEND (Interrupt 74)	Undefined		
9	SETPEND (Interrupt 73)	Undefined		
8	SETPEND (Interrupt 72)	Undefined		
7	SETPEND (Interrupt 71)	Undefined		
6	SETPEND (Interrupt 70)	Undefined		
5	SETPEND (Interrupt 69)	Undefined		
4	SETPEND (Interrupt 68)	Undefined		
3	SETPEND (Interrupt 67)	Undefined		
2	SETPEND (Interrupt 66)	Undefined		
1	SETPEND (Interrupt 65)	Undefined		
0	SETPEND (Interrupt 64)	Undefined		

Note: Write "0" to the bit whose bit symbol is "-".

(4) Interrupt Clear-Pending Register

Each bit corresponds to the specified interrupt number.

It can clear pending interrupts and checks which interrupts are currently pending.

Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced.

Writing "0" has no effect.

Read the current status of the corresponding interrupts; pending or not..

(a) Interrupt Clear-Pending Register 1

Bit	Bit Symbol	After reset	Type	Function
31	CLRPEND (Interrupt 31)	Undefined	R/W	[Reading] 0: Not pended 1: Pended [writing] (Note) 0: No effect 1: Clear pending interrupt
30	CLRPEND (Interrupt 30)	Undefined		
29	CLRPEND (Interrupt 29)	Undefined		
28	CLRPEND (Interrupt 28)	Undefined		
27	CLRPEND (Interrupt 27)	Undefined		
26	CLRPEND (Interrupt 26)	Undefined		
25	CLRPEND (Interrupt 25)	Undefined		
24	CLRPEND (Interrupt 24)	Undefined		
23	CLRPEND (Interrupt 23)	Undefined		
22	CLRPEND (Interrupt 22)	Undefined		
21	CLRPEND (Interrupt 21)	Undefined		
20	CLRPEND (Interrupt 20)	Undefined		
19	CLRPEND (Interrupt 19)	Undefined		
18	CLRPEND (Interrupt 18)	Undefined		
17	CLRPEND (Interrupt 17)	Undefined		
16	CLRPEND (Interrupt 16)	Undefined		
15	CLRPEND (Interrupt 15)	Undefined		
14	CLRPEND (Interrupt 14)	Undefined		
13	CLRPEND (Interrupt 13)	Undefined		
12	CLRPEND (Interrupt 12)	Undefined		
11	CLRPEND (Interrupt 11)	Undefined		
10	CLRPEND (Interrupt 10)	Undefined		
9	CLRPEND (Interrupt 9)	Undefined		
8	CLRPEND (Interrupt 8)	Undefined		
7	CLRPEND (Interrupt 7)	Undefined		
6	CLRPEND (Interrupt 6)	Undefined		
5	CLRPEND (Interrupt 5)	Undefined		
4	CLRPEND (Interrupt 4)	Undefined		
3	CLRPEND (Interrupt 3)	Undefined		
2	CLRPEND (Interrupt 2)	Undefined		
1	CLRPEND (Interrupt 1)	Undefined		
0	CLRPEND (Interrupt 0)	Undefined		

Note: Write "0" to the bit whose bit symbol is "-".

(b) Interrupt Clear-Pending Register 2

Bit	Bit Symbol	After reset	Type	Function
31	CLRPEND (Interrupt 63)	Undefined	R/W	[Reading] 0: Not pended 1: Pended [writing] (Note) 0: No effect 1: Clear pending interrupt
30	CLRPEND (Interrupt 62)	Undefined		
29	CLRPEND (Interrupt 61)	Undefined		
28	CLRPEND (Interrupt 60)	Undefined		
27	CLRPEND (Interrupt 59)	Undefined		
26	CLRPEND (Interrupt 58)	Undefined		
25	CLRPEND (Interrupt 57)	Undefined		
24	CLRPEND (Interrupt 56)	Undefined		
23	CLRPEND (Interrupt 55)	Undefined		
22	CLRPEND (Interrupt 54)	Undefined		
21	CLRPEND (Interrupt 53)	Undefined		
20	CLRPEND (Interrupt 52)	Undefined		
19	CLRPEND (Interrupt 51)	Undefined		
18	CLRPEND (Interrupt 50)	Undefined		
17	CLRPEND (Interrupt 49)	Undefined		
16	CLRPEND (Interrupt 48)	Undefined		
15	CLRPEND (Interrupt 47)	Undefined		
14	CLRPEND (Interrupt 46)	Undefined		
13	CLRPEND (Interrupt 45)	Undefined		
12	CLRPEND (Interrupt 44)	Undefined		
11	CLRPEND (Interrupt 43)	Undefined		
10	CLRPEND (Interrupt 42)	Undefined		
9	CLRPEND (Interrupt 41)	Undefined		
8	CLRPEND (Interrupt 40)	Undefined		
7	CLRPEND (Interrupt 39)	Undefined		
6	CLRPEND (Interrupt 38)	Undefined		
5	CLRPEND (Interrupt 37)	Undefined		
4	CLRPEND (Interrupt 36)	Undefined		
3	CLRPEND (Interrupt 35)	Undefined		
2	CLRPEND (Interrupt 34)	Undefined		
1	CLRPEND (Interrupt 33)	Undefined		
0	CLRPEND (Interrupt 32)	Undefined		

Note: Write "0" to the bit whose bit symbol is "-".

(c) Interrupt Clear-Pending Register 3

Bit	Bit Symbol	After reset	Type	Function
31	-	Undefined	R/W	[Reading] 0: Not pended 1: Pended [writing] (Note) 0: No effect 1: Clear pending interrupt
30	-	Undefined		
29	-	Undefined		
28	-	Undefined		
27	-	Undefined		
26	-	Undefined		
25	-	Undefined		
24	-	Undefined		
23	-	Undefined		
22	-	Undefined		
21	-	Undefined		
20	-	Undefined		
19	-	Undefined		
18	-	Undefined		
17	-	Undefined		
16	-	Undefined		
15	-	Undefined		
14	-	Undefined		
13	CLRPEND (Interrupt 77)	Undefined		
12	CLRPEND (Interrupt 76)	Undefined		
11	CLRPEND (Interrupt 75)	Undefined		
10	CLRPEND (Interrupt 74)	Undefined		
9	CLRPEND (Interrupt 73)	Undefined		
8	CLRPEND (Interrupt 72)	Undefined		
7	CLRPEND (Interrupt 71)	Undefined		
6	CLRPEND (Interrupt 70)	Undefined		
5	CLRPEND (Interrupt 69)	Undefined		
4	CLRPEND (Interrupt 68)	Undefined		
3	CLRPEND (Interrupt 67)	Undefined		
2	CLRPEND (Interrupt 66)	Undefined		
1	CLRPEND (Interrupt 65)	Undefined		
0	CLRPEND (Interrupt 64)	Undefined		

Note: Write "0" to the bit whose bit symbol is "-".

6.5.2.6. Interrupt Priority Register

Each interrupt is provided with 8 bits of an Interrupt Priority Register.

The following shows the addresses of the Interrupt Priority Registers corresponding to interrupt numbers.

Table 6.12 Address of Interrupt Priority Register

Address	31 24	23 16	15 8	7 0
0xE000E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000E434	PRI_55	PRI_54	PRI_53	PRI_52
0xE000E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000E44C	-	-	PRI_77	PRI_76

Note: The interrupt factors varies each products. For details, refer to "6.4.4. List of Interrupt Sources for Each Product".

The number of bits to be used for assigning a priority varies with each product.

TMPM370FYDFG/TMPM370FYFG use 3 bits for assigning a priority.

The following shows the fields of the Interrupt Priority Registers for interrupt numbers 0 to 3. The Interrupt Priority Registers for all other interrupt numbers have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.

Table 6.13 Configuration of Interrupt Priority Register for Interrupt Numbers 0 to 3

	31	30	29	28	27	26	25	24
bit symbol	PRI_3			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	PRI_2			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	PRI_1			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PRI_0			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:29	PRI_3[2:0]	R/W	Priority of interrupt number 3
28:24	-	R	Read as "0".
23:21	PRI_2[2:0]	R/W	Priority of interrupt number 2
20:16	-	R	Read as "0".
15:13	PRI_1[2:0]	R/W	Priority of interrupt number 1
12:8	-	R	Read as "0".
7:5	PRI_0[2:0]	R/W	Priority of interrupt number 0
4:0	-	R	Read as "0".

6.5.2.7. Vector Table Offset Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	TBLBASE	TBLOFF				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	TBLOFF							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TBLOFF							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TBLOFF	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:30	-	R	Read as "0".
29	TBLBASE	R/W	Table base Specify the memory space assigned vector table. 0: Code space 1: SRAM space
28:7	TBLOFF[24:0]	R/W	Offset value Set the offset value from the top of the space specified in <TBLBASE>. The offset must be aligned based on the number of exceptions in the table. This means that the minimum alignment is 32 words that interrupts can be used up to 16. For more interrupts, the alignment must be adjusted by rounding up to the next power of two.
6:0	-	R	Read as "0".

6.5.2.8. Application Interrupt and Reset Control Register

	31	30	29	28	27	26	25	24
bit symbol	VECTKEY/VECTKEYSTAT							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	VECTKEY/VECTKEYSTAT							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENDIANESS	-	-	-	-	PRIGROUP		
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	SYSRESET REQ	VECTCLR ACTIVE	VECTRESET
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	VECTKEYSTAT [15:0]	R	Register key Read as "0xFA05".
	VECTKEY[15:0]	W	Register key <VECTKEY[15:0]> must be written "0x05FA" when writing Application Interrupt and Reset Control Register.
15	ENDIANESS	R/W	Endianness bit (Note1) 0: Little endian 1: Big endian
14:11	-	R	Read as "0".
10:8	PRIGROUP[2:0]	R/W	Interrupt priority grouping 000: Pre-emption priority 7bits, sub priority 1bit 001: Pre-emption priority 6bits, sub priority 2bits 010: Pre-emption priority 5bits, sub priority 3bits 011: Pre-emption priority 4bits, sub priority 4bits 100: Pre-emption priority 3bits, sub priority 5bits 101: Pre-emption priority 2bits, sub priority 6bits 110: Pre-emption priority 1bit, sub priority 7bits 111: Pre-emption priority 0bit, sub priority 8bits The bit configuration to split the interrupt priority register <PRI_n> into pre-emption priority and sub priority.
7:3	-	R	Read as "0".
2	SYSRESETREQ	R/W	System reset request CPU outputs SYSRESETREQ signal when setting to "1". (Note2)
1	VECTCLR ACTIVE	R/W	Clear active vector bit 0: Do not clear 1: Clear all information for active NMI, faults, and interrupts. <VECTCLRACTIVE> is cleared by setting to "1". The re-initialization of STACK by application is required.
0	VECTRESET	R/W	System reset 0: Do not reset system. 1: Reset system. When setting to "1", the internal circuits of the CPU except debug components (FPB、DWT、ITM) are reset and <VECTRESET> is also cleared to "0".

Note1: The memory format of TMPM370FYDFG/TMPM370FYFG must be little endian.

Note2: When SYSRESETREQ signal is output, a warm reset occurs on TMPM370FYDFG/TMPM370FYFG. <SYSRESETREQ> is cleared to "0" by a warm reset.

6.5.2.9. System Handler Priority Register

Each exception is provided with 8 bits of a System Handler Priority Register.

The following shows the addresses of the System Handler Priority Registers corresponding to each exception

Table 6.14 Address of System Handler Priority Register

Address	31 24	23 16	15 8	7 0
0xE000ED18	PRI_7	PRI_6 (Usage fault)	PRI_5 (Bus fault)	PRI_4 (Memory management)
0xE000ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12 (Debug monitor)

The number of bits to be used for assigning a priority varies with each product.

TMPM370FYDFG/TMPM370FYFG use 3 bits for assigning a priority.

The following shows the fields of the System Handler Priority Registers for Usage fault, Bus fault, and Memory Management. Unused bits return "0" when read, and writing to unused bits has no effect.

Table 6.15 Configuration of System Handler Priority Register

	31	30	29	28	27	26	25	24
bit symbol	PRI_7			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	PRI_6			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	PRI_5			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PRI_4			-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:29	PRI_7[2:0]	R/W	Reserved
28:24	-	R	Read as "0".
23:21	PRI_6[2:0]	R/W	Priority of Usage fault
20:16	-	R	Read as "0".
15:13	PRI_5[2:0]	R/W	Priority of Bus fault
12:8	-	R	Read as "0".
7:5	PRI_4[2:0]	R/W	Priority of Memory management
4:0	-	R	Read as "0".

6.5.2.10. System Handler Control and State Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICKACT	PENDSVACT	-	MONITOR ACT
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:19	-	R	Read as "0".
18	USGFAULTENA	R/W	Usage fault 0: Disabled 1: Enabled
17	BUSFAULTENA	R/W	Bus fault 0: Disabled 1: Enabled
16	MEMFAULTENA	R/W	Memory management 0: Disabled 1: Enabled
15	SVCALLPENDED	R/W	SVCall 0: Not pended 1: Pended
14	BUSFAULT PENDED	R/W	Bus fault 0: Not pended 1: Pended
13	MEMFAULT PENDED	R/W	Memory management 0: Not pended 1: Pended
12	USGFAULT PENDED	R/W	Usage fault 0: Not pended 1: Pended
11	SYSTICKACT	R/W	SysTick 0: Inactive 1: Active
10	PENDSVACT	R/W	PendSV 0: Inactive 1: Active
9	-	R	Read as "0".
8	MONITORACT	R/W	Debug monitor 0: Inactive 1: Active
7	SVCALLACT	R/W	SVCall 0: Inactive 1: Active

6:4	-	R	Read as "0".
3	USGFAULTACT	R/W	Usage fault 0: Inactive 1: Active
2	-	R	Read as "0".
1	BUSFAULTACT	R/W	Bus fault 0: Inactive 1: Active
0	MEMFAULTACT	R/W	Memory management 0: Inactive 1: Active

Note: Clearing or setting the active bits requires extreme caution because clearing or setting these bits does not modify stack contents.

6.5.3. CG Registers

(1) When using interrupts, be sure to follow the sequence of actions shown below:

(a) When the general purpose port is shared with an external interrupt pin, enable input of interrupt to corresponding general purpose port.

(b) Set the interrupt detection level, etc. at the initializing.

(c) Clear interrupt requests.

(d) Enable interrupts.

(2) Be sure to make each setting while the interrupts are disabled.

(3) External interrupt pins of TMPM370FYDFG/TMPM370FYFG can be used for releasing low-power consumption mode. Using for releasing low-power consumption mode or not, and interrupt detection level are specified by the CG.

For details, refer to "6.4.3.5. Caution when Using External Interrupt Pins".

6.5.3.1. CGICRCG (CG Interrupt Request Clear Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	ICRCG				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:5	-	R	Read as "0".
4:0	ICRCG[4:0]	W	Clear interrupt requests 0_0000: INT0 0_1000: INT8 0_0001: INT1 0_1001: INT9 0_0010: INT2 0_1010: INTA 0_0011: INT3 0_1011: INTB 0_0100: INT4 0_1100: INTC 0_0101: INT5 0_1101: INTD 0_0110: INT6 0_1110: INTE 0_0111: INT7 0_1111: INTF Others: Prohibit

6.5.3.2. CGNMIFLG (NMI Generation Factor Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	-	NMIFLG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
0	NMIFLG0	R	NMI Generation factor flag 0: Not generated by WDT 1: Generated by WDT

Note: After reading <NMIFLG0>, <NMIFLG0> is cleared to "0" automatically.

6.5.3.3. CGRSTFLG (Reset Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	OFDRSTF	DBGRSTF	VLTDRSTF	WDTRSTF	PINRSTF	PONRSTF
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
5	OFDRSTF	R	OFD reset flag 0: RESET is not generated by OFD. 1: RESET is generated by OFD.
		W	0: Clear <OFDRSTF> 1: Don't care
4	DBGRSTF	R	Debug reset flag (Note1) 0: RESET is not generated by <SYSRESETREQ>. 1: RESET is generated by <SYSRESETREQ>.
		W	0: Clear <DBGRSTF> 1: Don't care
3	VLTDRSTF	R	VLTD reset flag 0: RESET is not generated by VLTD. 1: RESET is generated by VLTD.
		W	0: clear <VLTDRSTF> 1: Don't care
2	WDTRSTF	R	WDT reset flag 0: RESET is not generated by WDT. 1: RESET is generated by WDT.
		W	0: Clear <WDTRSTF> 1: Don't care
1	PINRSTF	R	$\overline{\text{RESET}}$ pin reset flag 0: RESET is not generated by $\overline{\text{RESET}}$ pin. 1: RESET is generated by $\overline{\text{RESET}}$ pin.
		W	0: Clear <PINRSTF> 1: Don't care
0	PONRSTF	R	POR reset flag 0: RESET is not generated by POR. 1: RESET is generated by POR.
		W	0: Clear <PONRSTF> 1: Don't care

Note1: This flag indicates the RESET generated by setting <SYSRESETREQ> of Application Interrupt and Reset Control Register in the CPU's NVIC.

Note2: TMPM370FYDFG/TMPM370FYFG have the POR. CGRSTFLG is reset by only POR. When turning on a power supply, <PONRSTF> is set. But, <PORRSTF> is not set by the RESET except POR. At this time, the reset flag which is corresponded to reset factor.

6.5.3.4. CGIMCGA (CG Interrupt Mode Control Register A)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG3			EMST3		-	INT3EN
After reset	0	0	1	0	0	0	Undefined	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	EMCG2			EMST2		-	INT2EN
After reset	0	0	1	0	0	0	Undefined	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	EMCG1			EMST1		-	INT1EN
After reset	0	0	1	0	0	0	Undefined	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	EMCG0			EMST0		-	INT0EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	-	R	Read as "0".
30:28	EMCG3[2:0]	R/W	Set an interrupt detection level of INT3 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
27:26	EMST3[1:0] (Note1)	R	Monitor of an interrupt detection level of INT3 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
25	-	R	Read as undefined value.
24	INT3EN (Note2)	R/W	Enable releasing low-power consumption mode by INT3 0: Disabled 1: Enabled
23	-	R	Read as "0".
22:20	EMCG2[2:0]	R/W	Set an interrupt detection level of INT2 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
19:18	EMST2[1:0] (Note1)	R	Monitor of an interrupt detection level of INT2 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
17	-	R	Read as undefined value.
16	INT2EN (Note2)	R/W	Enable releasing low-power consumption mode by INT2 0: Disabled 1: Enabled
15	-	R	Read as "0".

14:12	EMCG1[2:0]	R/W	Set an interrupt detection level of INT1 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
11:10	EMST1[1:0] (Note1)	R	Monitor of an interrupt detection level of INT1 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
9	-	R	Read as undefined value.
8	INT1EN (Note2)	R/W	Enable releasing low-power consumption mode by INT1 0: Disabled 1: Enabled
7	-	R	Read as "0".
6:4	EMCG0[2:0]	R/W	Set an interrupt detection level of INT0 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
3:2	EMST0[1:0] (Note1)	R	Monitor of an interrupt detection level of INT0 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
1	-	R	Read as undefined value.
0	INT0EN (Note2)	R/W	Enable releasing low-power consumption mode by INT0 0: Disabled 1: Enabled

Note1: <EMSTm> is effective only when <EMCGm[2:0]> is set to "100" for both edges. The interrupt detection level which is used for releasing low-power consumption mode can be checked by reading <EMSTm>. When an interrupt is cleared by CGICRCG, <EMSTm> is also cleared to "0".

Note2:<EMCGm[2:0]> and <INTmEN> are not set simultaneously. After setting <EMCGm[2:0]>, set <INTmEN>.

Note3: "m" means a suffix of a bit symbol.

6.5.3.5. CGIMCGB (CG Interrupt Mode Control Register B)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG7			EMST7		-	INT7EN
After reset	0	0	1	0	0	0	Undefined	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	EMCG6			EMST6		-	INT6EN
After reset	0	0	1	0	0	0	Undefined	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	EMCG5			EMST5		-	INT5EN
After reset	0	0	1	0	0	0	Undefined	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	EMCG4			EMST4		-	INT4EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	-	R	Read as "0".
30:28	EMCG7[2:0]	R/W	Set an interrupt detection level of INT7 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
27:26	EMST7[1:0] (Note1)	R	Monitor of an interrupt detection level of INT7 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
25	-	R	Read as undefined value.
24	INT7EN (Note2)	R/W	Enable releasing low-power consumption mode by INT7 0: Disabled 1: Enabled
23	-	R	Read as "0".
22:20	EMCG6[2:0]	R/W	Set an interrupt detection level of INT6 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
19:18	EMST6[1:0] (Note1)	R	Monitor of an interrupt detection level of INT6 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
17	-	R	Read as undefined value.
16	INT6EN (Note2)	R/W	Enable releasing low-power consumption mode by INT6 0: Disabled 1: Enabled
15	-	R	Read as "0".

14:12	EMCG5[2:0]	R/W	Set an interrupt detection level of INT5 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
11:10	EMST5[1:0] (Note1)	R	Monitor of an interrupt detection level of INT5 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
9	-	R	Read as undefined value.
8	INT5EN (Note2)	R/W	Enable releasing low-power consumption mode by INT5 0: Disabled 1: Enabled
7	-	R	Read as "0".
6:4	EMCG4[2:0]	R/W	Set an interrupt detection level of INT4 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
3:2	EMST4[1:0] (Note1)	R	Monitor of an interrupt detection level of INT4 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge 10: Detected falling edge 11: Detected both edges
1	-	R	Read as undefined value.
0	INT4EN (Note2)	R/W	Enable releasing low-power consumption mode by INT4 0: Disabled 1: Enabled

Note1: <EMSTm> is effective only when <EMCGm[2:0]> is set to "100" for both edges. The interrupt detection level which is used for releasing low-power consumption mode can be checked by reading <EMSTm>. When an interrupt is cleared by CGICRCG, <EMSTm> is also cleared to "0".

Note2:<EMCGm[2:0]> and <INTmEN> are not set simultaneously. After setting <EMCGm[2:0]>, set <INTmEN>.

Note3: "m" means a suffix of a bit symbol.

6.5.3.6. CGIMCGC (CG Interrupt Mode Control Register C)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCGB			EMSTB		-	INTBEN
After reset	0	0	1	0	0	0	Undefined	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	EMCGA			EMSTA		-	INTAEN
After reset	0	0	1	0	0	0	Undefined	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	EMCG9			EMST9		-	INT9EN
After reset	0	0	1	0	0	0	Undefined	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	EMCG8			EMST8		-	INT8EN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	-	R	Read as "0".
30:28	EMCGB[2:0]	R/W	Set an interrupt detection level of INTB for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
27:26	EMSTB[1:0] (Note1)	R	Monitor of an interrupt detection level of INTB when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
25	-	R	Read as undefined value.
24	INTBEN (Note2)	R/W	Enable releasing low-power consumption mode by INTB 0: Disabled 1: Enabled
23	-	R	Read as "0".
22:20	EMCGA[2:0]	R/W	Set an interrupt detection level of INTA for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
19:18	EMSTA[1:0] (Note1)	R	Monitor of an interrupt detection level of INTA when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
17	-	R	Read as undefined value.
16	INTAEN (Note2)	R/W	Enable releasing low-power consumption mode by INTA 0: Disabled 1: Enabled
15	-	R	Read as "0".

14:12	EMCG9[2:0]	R/W	Set an interrupt detection level of INT9 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
11:10	EMST9[1:0] (Note1)	R	Monitor of an interrupt detection level of INT9 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
9	-	R	Read as undefined value.
8	INT9EN (Note2)	R/W	Enable releasing low-power consumption mode by INT9 0: Disabled 1: Enabled
7	-	R	Read as "0".
6:4	EMCG8[2:0]	R/W	Set an interrupt detection level of INT8 for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
3:2	EMST8[1:0] (Note1)	R	Monitor of an interrupt detection level of INT8 when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
1	-	R	Read as undefined value.
0	INT8EN (Note2)	R/W	Enable releasing low-power consumption mode by INT8 0: Disabled 1: Enabled

Note1: <EMSTm> is effective only when <EMCGm[2:0]> is set to "100" for both edges. The interrupt detection level which is used for releasing low-power consumption mode can be checked by reading <EMSTm>. When an interrupt is cleared by CGICRCG, <EMSTm> is also cleared to "0".

Note2:<EMCGm[2:0]> and <INTmEN> are not set simultaneously. After setting <EMCGm[2:0]>, set <INTmEN>.

Note3: "m" means a suffix of a bit symbol.

6.5.3.7. CGIMCGD (CG Interrupt Mode Control Register D)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCGF			EMSTF		-	INTFEN
After reset	0	0	1	0	0	0	Undefined	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	EMCGE			EMSTE		-	INTEEN
After reset	0	0	1	0	0	0	Undefined	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	EMCGD			EMSTD		-	INTDEN
After reset	0	0	1	0	0	0	Undefined	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	EMCGC			EMSTC		-	INTCEN
After reset	0	0	1	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31	-	R	Read as "0".
30:28	EMCGF[2:0]	R/W	Set an interrupt detection level of INTF for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
27:26	EMSTF[1:0] (Note1)	R	Monitor of an interrupt detection level of INTF when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
25	-	R	Read as undefined value.
24	INTFEN (Note2)	R/W	Enable releasing low-power consumption mode by INTF 0: Disabled 1: Enabled
23	-	R	Read as "0".
22:20	EMCGE[2:0]	R/W	Set an interrupt detection level of INTE for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
19:18	EMSTE[1:0] (Note1)	R	Monitor of an interrupt detection level of INTE when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
17	-	R	Read as undefined value.
16	INTEEN (Note2)	R/W	Enable releasing low-power consumption mode by INTE 0: Disabled 1: Enabled
15	-	R	Read as "0".

14:12	EMCGD[2:0]	R/W	Set an interrupt detection level of INTD for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
11:10	EMSTD[1:0] (Note1)	R	Monitor of an interrupt detection level of INTD when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
9	-	R	Read as undefined value.
8	INTDEN (Note2)	R/W	Enable releasing low-power consumption mode by INTD 0: Disabled 1: Enabled
7	-	R	Read as "0".
6:4	EMCGC[2:0]	R/W	Set an interrupt detection level of INTC for releasing low-power consumption mode. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Others: Prohibit
3:2	EMSTC[1:0] (Note1)	R	Monitor of an interrupt detection level of INTC when releasing low-power consumption mode. 00: Not detected 01: Detected rising edge/ 10: Detected falling edge 11: Detected both edges
1	-	R	Read as undefined value.
0	INTCN (Note2)	R/W	Enable releasing low-power consumption mode by INTC 0: Disabled 1: Enabled

Note1: <EMSTm> is effective only when <EMCGm[2:0]> is set to "100" for both edges. The interrupt detection level which is used for releasing low-power consumption mode can be checked by reading <EMSTm>. When an interrupt is cleared by CGICRCG, <EMSTm> is also cleared to "0".

Note2:<EMCGm[2:0]> and <INTmEN> are not set simultaneously. After setting <EMCGm[2:0]>, set <INTmEN>.

Note3: "m" means a suffix of a bit symbol.

7. Product Information

This chapter summarizes information on the number of channels or units, pin information, and other product specific functions about peripheral functions.

7.1. Information of Each Peripheral Function

7.1.1. Clock/Mode Control (CG)

7.1.1.1. Built-in List

The following table shows the built-in list for each product.

Table 7.1 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.1.2. Base Address

The following table shows the Base Address for each product.

Table 7.2 Base Address

Product	Base Address
TMPM370FYDFG	0x4004_0200
TMPM370FYFG	0x4004_0200

7.1.2. Exception

7.1.2.1. Built-in List

The following table shows the built-in list for each product.

Table 7.3 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.2.2. Base Address

The following table shows the Base Address for each product.

Table 7.4 Base Address

Product		Base Address
TMPM370FYDFG	NVIC	0xE000_E000
	CG	0x4004_0200
TMPM370FYDFG	NVIC	0xE000_E000
	CG	0x4004_0200

7.1.3. Input/output Port

7.1.3.1. Built-in List

The following table shows the built-in list for each product.

Table 7.5 Built-in List (1/2)

Product	Available or N/A (✓: Available , -: N/A)					
	Port A	Port B	Port C	Port D	Port E	Port F
TMPM370FYDFG	✓	✓	✓	✓	✓	✓
TMPM370FYFG	✓	✓	✓	✓	✓	✓

Table 7.6 Built-in List (2/2)

Product	Available or N/A (✓: Available , -: N/A)					
	Port G	Port H	Port I	Port J	Port K	Port L
TMPM370FYDFG	✓	✓	✓	✓	✓	✓
TMPM370FYFG	✓	✓	✓	✓	✓	✓

7.1.3.2. Base Address

The following table shows the Base Address for each product.

Table 7.7 Base Address (1/2)

Product	Base Address					
	Port A	Port B	Port C	Port D	Port E	Port F
TMPM370FYDFG	0x4000_0000	0x4000_0040	0x4000_0080	0x4000_00C0	0x4000_0100	0x4000_0140
TMPM370FYFG	0x4000_0000	0x4000_0040	0x4000_0080	0x4000_00C0	0x4000_0100	0x4000_0140

Table 7.8 Base Address (2/2)

Product	Base Address					
	Port G	Port H	Port I	Port J	Port K	Port L
TMPM370FYDFG	0x4000_0180	0x4000_01C0	0x4000_0200	0x4000_0240	0x4000_0280	0x4000_02C0
TMPM370FYFG	0x4000_0180	0x4000_01C0	0x4000_0200	0x4000_0240	0x4000_0280	0x4000_02C0

7.1.4. 16-bit Timer/event Counter (TMRB)

7.1.4.1. Built-in List

The following table shows the built-in list for each product.

Table 7.9 Built-in List

Product	Available or N/A (✓: Available , -: N/A)							
	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7
TMPM370FYDFG	✓	✓	✓	✓	✓	✓	✓	✓
TMPM370FYFG	✓	✓	✓	✓	✓	✓	✓	✓

7.1.4.2. Base Address

The following table shows the Base Address for each product.

Table 7.10 Base Address (1/2)

Product	Base Address			
	ch0	ch1	ch2	ch3
TMPM370FYDFG	0x4001_0000	0x4001_0040	0x4001_0080	0x4001_00C0
TMPM370FYFG	0x4001_0000	0x4001_0040	0x4001_0080	0x4001_00C0

Table 7.11 Base Address (2/2)

Product	Base Address			
	ch4	ch5	ch6	ch7
TMPM370FYDFG	0x4001_0100	0x4001_0140	0x4001_0180	0x4001_01C0
TMPM370FYFG	0x4001_0100	0x4001_0140	0x4001_0180	0x4001_01C0

7.1.4.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.12 Function pins and ports

Channel	Functional pin	Port name
TMRB ch0	TB0IN	PA0
	TB0OUT	PA1
TMRB ch1	TB1IN	PA2
	TB1OUT	PA3
TMRB ch2	TB2IN	PE4
	TB2OUT	PE5
TMRB ch3	TB3IN	PE6
	TB3OUT	PE7
TMRB ch4	TB4IN	PA7
	TB4OUT	PE3
TMRB ch5	TB5IN	PD0
	TB5OUT	PD1
TMRB ch6	TB6IN	PA6
	TB6OUT	PA5
TMRB ch7	TB7IN	PF0
	TB7OUT	PF1

7.1.4.4. Internal Signal Connection

The connection of input/output signals and peripheral functions is shown below.

Table 7.13 Connection with Input/output Signals and Peripheral Functions (1/2)

Channel	Signal Name	Connected peripheral function
TMRB ch0	INTCA00	CG
	INTCAP01	CG
	INTTB00	CG
	INTTB01	CG
	TB0OUT	PORT
TMRB ch1	INTCA10	CG
	INTCAP11	CG
	INTTB10	CG
	INTTB11	CG
	TB1OUT	PORT
TMRB ch2	INTCA20	CG
	INTCAP21	CG
	INTTB20	CG
	INTTB21	CG
	TB2OUT	PORT
TMRB ch3	INTCA30	CG
	INTCAP31	CG
	INTTB30	CG
	INTTB31	CG
	TB3OUT	PORT

Table 7.14 Connection with Input/output Signals and Peripheral Functions (2/2)

Channel	Signal Name	Connected peripheral function
TMRB ch4	INTCA40	CG
	INTCAP41	CG
	INTTB40	CG
	INTTB41	CG
	TB4OUT	PORT SIO/UART0 SIO/UART1
TMRB ch5	INTCA50	CG
	INTCAP51	CG
	INTTB50	CG
	INTTB51	ADC CG
	TB5OUT	PORT
TMRB ch6	INTCA60	CG
	INTCAP61	CG
	INTTB60	CG
	INTTB61	CG
	TB6OUT	PORT
TMRB ch7	INTCA70	CG
	INTCAP71	CG
	INTTB70	CG
	INTTB71	CG
	TB7OUT	PORT SIO/UART2 SIO/UART3

7.1.5. Serial Channel (SIO/UART)

7.1.5.1. Built-in List

The following table shows the built-in list for each product.

Table 7.15 Built-in List

Product	Available or N/A (✓: Available , -: N/A)			
	ch0	ch1	ch2	ch3
TMPM370FYDFG	✓	✓	✓	✓
TMPM370FYFG	✓	✓	-	✓

7.1.5.2. Base Address

The following table shows the Base Address for each product.

Table 7.16 Base Address

Product	Base Address			
	ch0	ch1	ch2	ch3
TMPM370FYDFG	0x4002_0080	0x4002_00C0	0x4002_0100	0x4002_0140
TMPM370FYFG	0x4002_0080	0x4002_00C0	0x4002_0100	0x4002_0140

7.1.5.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.17 Function pins and ports

Channel	Functional pins	Port name
UART/SIO ch0	TXD0	PE0
	RXD0	PE1
	$\overline{\text{CTS0}}$/SCLK0	PE2
UART/SIO ch1	TXD1	PA5
	RXD1	PA6
	$\overline{\text{CTS1}}$/SCLK1	PA4
UART/SIO ch2	TXD2	PD5
	RXD2	PD6
	$\overline{\text{CTS2}}$/SCLK2	PD4
UART/SIO ch3	TXD3	PF3
	RXD3	PF4
	$\overline{\text{CTS3}}$/SCLK3	PF2

7.1.5.4. Internal Signal Connection

The connection of input/output signals and peripheral functions is shown below.

Table 7.18 Connection with Input/output Signals and Peripheral Functions

Channel	Signal Name	Connected peripheral function
SIO/UART ch0	SIOCLK in UART mode	TMRB ch4
SIO/UART ch1	SIOCLK in UART mode	TMRB ch4
SIO/UART ch2	SIOCLK in UART mode	TMRB ch7
SIO/UART ch3	SIOCLK in UART mode	TMRB ch7

7.1.6. 12-bit analog-to-digital converter (ADC)

7.1.6.1. Built-in List

The following table shows the built-in list for each product.

Table 7. 19 Built-in List

Product	Available or N/A (✓: Available , -: N/A)	
	unit A	unit B
TMPM370FYDFG	✓	✓
TMPM370FYFG	✓	✓

7.1.6.2. Base Address

The following table shows the Base Address for each product.

Table 7.20 Base Address

Product	Base Address	
	unit A	unit B
TMPM370FYDFG	0x4003_0000	0x4003_0200
TMPM370FYFG	0x4003_0000	0x4003_0200

7.1.6.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.21 Function pins and ports (1/2)

Unit	Input channel	Functional pins	Port name
ADC unit A	ch0	AINA0	PH0
	ch1	AINA1	PH1
	ch2	AINA2	PH2
	ch3	AINA3	PH3
	ch4	AINA4	PH4
	ch5	AINA5	PH5
	ch6	AINA6	PH6
	ch7	AINA7	PH7
	ch8	AINA8	PI0
	ch9	AINA9	PI1
	ch10	AINA10	PI2
	ch11	AINA11	PI3
	ch0	AINA0	PH0

Table 7.22 Function pins and ports (2/2)

Unit	Input channel	Functional pins	Port name
ADC unit B	ch0	AINB0	PI1
	ch1	AINB1	PI2
	ch2	AINB2	PI3
	ch3	AINB3	PJ0
	ch4	AINB4	PJ1
	ch5	AINB5	PJ2
	ch6	AINB6	PJ3
	ch7	AINB7	PJ4
	ch8	AINB8	PJ5
	ch9	AINB9	PJ6
	ch10	AINB10	PJ7
	ch11	AINB11	PK0
	ch12	AINB12	PK1

7.1.6.4. Internal Signal Connection

The connection of input/output signals and peripheral functions is shown below.

Table 7.23 Connection with Input/output Signals and Peripheral Functions

Unit	Signal Name	Connected peripheral function
ADC unit A	INTTB51	TMRB ch5
ADC unit B	INTTB51	TMRB ch5

Note: For connecting with PMD, refer to "7.1.7. Motor Control Circuit (PMD)".

7.1.7. Motor Control Circuit (PMD)

7.1.7.1. Built-in List

The following table shows the built-in list for each product.

Table 7.24 Built-in List

Product	Available or N/A (✓: Available , -: N/A)	
	ch0	ch1
TMPM370FYDFG	✓	✓
TMPM370FYFG	✓	✓

7.1.7.2. Base Address

The following table shows the Base Address for each product.

Table 7.25 Base Address

Product	Base Address	
	ch0	ch1
TMPM370FYDFG	0x4005_0400	0x4005_0480
TMPM370FYFG	0x4005_0400	0x4005_0480

7.1.7.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.26 Function pins and ports

Channel	Functional pins	Port name
PMD ch0	UO0	PC0
	VO0	PC2
	WO0	PC4
	XO0	PC1
	YO0	PC3
	ZO0	PC5
	$\overline{\text{EMG0}}$	PC6
	$\overline{\text{OVV0}}$	PC7

Table 7.27 Function pins and ports

Channel	Functional pins	Port name
PMD ch1	UO1	PG0
	VO1	PG2
	WO1	PG4
	XO1	PG1
	YO1	PG3
	ZO1	PG5
	$\overline{\text{EMG1}}$	PG6
	$\overline{\text{OVV1}}$	PG7

7.1.7.4. Internal Signal Connection

The connection of input/output signals and peripheral functions is shown below.

Table 7.28 Connection with Input Signals and Peripheral Functions

Output peripheral function	Signal name	Connected peripheral function
VE ch0	VECMPU0	PMD ch0
	VECMPV0	PMD ch0
	VECMPW0	PMD ch0
	VEOUTCR0	PMD ch0
	VETRGCMP00	PMD ch0
	VETRGCMP10	PMD ch0
	VETRGSEL0	PMD ch0
ADC unit A	ADC unit A monitor signal 0	PMD ch0
	ADC unit A monitor signal 1	PMD ch0

Table 7.29 Connection with Input Signals and Peripheral Functions

Output peripheral function	Signal name	Connected peripheral function
VE ch1	VECMPU1	PMD ch1
	VECMPV1	PMD ch1
	VECMPW1	PMD ch1
	VEOUTCR1	PMD ch1
	VETRGCMP01	PMD ch1
	VETRGCMP11	PMD ch1
	VETRGSEL1	PMD ch1
ADC unit B	ADC unit B monitor signal 0	PMD ch1
	ADC unit B monitor signal 1	PMD ch1

Table 7.30 Connection with Output Signals and Peripheral Functions

Input peripheral function	Signal name	Connected peripheral function
ADC unit A	PMD0TRG0	ADC unit A
	PMD0TRG1	ADC unit A
	PMD0TRG2	ADC unit A
	PMD0TRG3	ADC unit A
	PMD0TRG4	ADC unit A
	PMD0TRG5	ADC unit A

Table 7.31 Connection with Output Signals and Peripheral Functions

Input peripheral function	Signal name	Connected peripheral function
ADC unit B	PMD1TRG0	ADC unit B
	PMD1TRG1	ADC unit B
	PMD1TRG2	ADC unit B
	PMD1TRG3	ADC unit B
	PMD1TRG4	ADC unit B
	PMD1TRG5	ADC unit B

7.1.8. Vector Engine (VE)

7.1.8.1. Built-in List

The following table shows the built-in list for each product.

Table 7.32 Built-in List

Product	Available or N/A (✓: Available , -: N/A)	
	ch0	ch1
TMPM370FYDFG	✓	✓
TMPM370FYFG	✓	✓

7.1.8.2. Base Address

The following table shows the Base Address for each product.

Table 7.33 Base Address

Registers	Product	Base Address	
		ch0	ch1
VE Control Registers	TMPM370FYDFG	0x4005_0000	0x4005_0000
	TMPM370FYFG	0x4005_0000	0x4005_0000
Common Registers	TMPM370FYDFG	0x4005_0000	0x4005_0000
	TMPM370FYFG	0x4005_0000	0x4005_0000
Specific Registers	TMPM370FYDFG	0x4005_0044	0x4005_0044
	TMPM370FYFG	0x4005_00DC	0x4005_00DC

7.1.8.3. Internal Signal Connection

For internal signal connection, refer to "7.1.7. Motor Control Circuit (PMD)".

7.1.9. Encoder Input Circuit (ENC)

7.1.9.1. Built-in List

The following table shows the built-in list for each product.

Table 7.34 Built-in List

Product	Available or N/A (✓: Available , -: N/A)	
	ch0	ch1
TMPM370FYDFG	✓	✓
TMPM370FYFG	✓	✓

7.1.9.2. Base Address

The following table shows the Base Address for each product.

Table 7.35 Base Address

Product	Base Address	
	ch0	ch1
TMPM370FYDFG	0x4001_0400	0x4001_0500
TMPM370FYFG	0x4001_0400	0x4001_0500

7.1.9.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.36 Function pins and ports

Channel	Functional pins	Port name
ENC ch0	ENCA0	PD0
	ENCB0	PD1
	ENCZ0	PD2
ENC ch1	ENCA1	PF2
	ENCB1	PF3
	ENCZ1	PF4

7.1.10. Voltage Detection circuit (VLTD)

7.1.10.1. Built-in List

The following table shows the built-in list for each product.

Table 7.37 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.10.2. Base Address

The following table shows the Base Address for each product.

Table 7.38 Base Address

Product	Base Address
TMPM370FYDFG	0x4004_0900
TMPM370FYFG	0x4004_0900

7.1.11. Frequency detection circuit (OFD)

7.1.11.1. Built-in List

The following table shows the built-in list for each product.

Table 7.39 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.11.2. Base Address

The following table shows the Base Address for each product.

Table 7.40 Base Address

Product	Base Address
TMPM370FYDFG	0x4004_0800
TMPM370FYFG	0x4004_0800

7.1.12. Watchdog Timer (WDT)

7.1.12.1. Built-in List

The following table shows the built-in list for each product.

Table 7.41 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.12.2. Base Address

The following table shows the Base Address for each product.

Table 7.42 Base Address

Product	Base Address
TMPM370FYDFG	0x4004_0000
TMPM370FYFG	0x4004_0000

7.1.12.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.43 Function pins and ports

Functional pins	Port name
$\overline{\text{WDTOUT}}$	-

7.1.13. Op-amplifiers/Analog Comparators (AMP/CMP)

7.1.13.1. Built-in List

The following table shows the built-in list for each product.

Table 7.44 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.13.2. Base Address

The following table shows the Base Address for each product.

Table 7.45 Base Address

Registers	Product	Base Address
Op-amplifier	TMPM370FYDFG	0x4003_0400
	TMPM370FYFG	0x4003_0400
Analog Comparator	TMPM370FYDFG	0x4003_0420
	TMPM370FYFG	0x4003_0420

7.1.13.3. Function Pins and Ports

The port assignment of the function pins is shown below.

Table 7.46 AMP Function Pins and Ports

Functional pins	Port name
AMPA input	PI1
AMPB input	PI2
AMPC input	PI3
AMPD input	PJ0

Table 7.47 CMP Function Pins and Ports

Functional pins	Port name
CMPA input	PI1
CMPB input	PI2
CMPC input	PI3
CMPD input	PJ0

7.1.14. Flash memory

7.1.14.1. Built-in List

The following table shows the built-in list for each product.

Table 7.48 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.14.2. Base Address

The following table shows the Base Address for each product.

Table 7.49 Base Address

Product	Base Address
TMPM370FYDFG	0x41FF_F000
TMPM370FYFG	0x41FF_F000

7.1.14.3. Flash Memory Block Configuration

The following table shows the flash memory block configuration for each product.

Table 7.50 Flash Memory Block Configurations

Block name	Block size (KB)
Block0	64
Block1	64
Block2	64
Block3	32
Block4	16
Block5	16

7.1.14.4. Macro Code with ID-Read

The following table shows the macro codes for each product.

Table 7.51 Macro code with ID-Read

	Code (ID[7:0])
Macro code	0x13

7.1.14.5. Used Resources in Single Boot Mode

The following table shows the used resources in single boot mode.

Table 7.52 Used Resources in Single Boot Mode

Peripheral functions	Channel	Interface	Pin name
BOOT	-	-	PF0 ($\overline{\text{BOOT}}$)
SIO/UART	ch0	UART mode	PE0 (TXD0)/PE1 (RXD0)
		SIO mode	PE0 (TXD0)/PE1 (RXD0)/PE2 (SCLK0)/ PE4 (Handshake signal)
TMRB	ch0	-	-

The following table shows the RAM address range which can be transferred by the RAM transfer command for each product.

Table 7.53 RAM Address Range

Product name	RAM address range
TMPM370FYDFG	0x2000_0400 to 0x2000_27FF
TMPM370FYFG	0x2000_0400 to 0x2000_27FF

7.1.15. Protect/Security Function

7.1.15.1. Built-in List

The following table shows the built-in list for each product.

Table 7.54 Built-in List

Product	Available or N/A (✓: Available , -: N/A)
TMPM370FYDFG	✓
TMPM370FYFG	✓

7.1.15.2. Base Address

The following table shows the Base Address for each product.

Table 7.55 Base Address

Product	Base Address
TMPM370FYDFG	0x41FF_F000
TMPM370FYFG	0x41FF_F000

7.1.16. Debug Interface

7.1.16.1. Debug Interface Pin List

The debug interface includes a JTAG (TMS, TCK, TDI, $\overline{\text{TRST}}$) and a serial wire (SWDIO, SWCLK).
There are also trace output (TRACEDATA0 to 1), clock output (TRACECLK), and serial wire viewer (SWV).

Table 7.56 Debug Interface Pin List

SWJ-DP pin	Port name
TMS/SWDIO	PB3
TCK/SWCLK	PB4
TDO/SWV	PB5
TDI	PB6
$\overline{\text{TRST}}$	PB7
TRACECLK	PB0
TRACEDATA0	PB1
TRACEDATA1	PB2

8. Input/output Port

8.1. Outlines

It is described about the register and setting of port. A list of the functions is indicated below.

Function Classification	Function	Description
Port	-	Programmable pull-up/Programmable pull-down/Open-drain output are selectable.
Peripheral Function pins	External Interrupt pin	External interrupt pin has a noise filter (Filter width 30ns typ.)
	16-bit Timer/Event Counter	TMRB input capture pin, TMRB output pin
	Serial Channel	Transmit pin, Receive pin, Clock pin, Hand shake pin
	12-bit Analog-to-Digital Convertor	Analog input pin
	Motor Control Circuit	X/Y/Z phase output pins, U/V/W phase output pins, EMG detection input pin, OVV detection input pin
	Encoder Input Circuit	Encoder input pin
	Op-Amps/Analog Comparators	Analog input pin
Debug pins	JTAG	Debug pins
	SW	Debug pins
	Trace	Debug pins
Control pins	High-speed clock	High-speed oscillator connection pin
	BOOT mode control	BOOT mode control pin

8.2. Signal connection list

This table is sorted the function pins by the signal name of the block diagram which is described each section. Register setting of the peripherals function is being explained in the port order, so please use for a reverse lookup of port name.

The numerical value shows the pin number.

Table 8.1 Signal connection list: SIO/UART

Reference peripheral function chapter	Function pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYFG (LQFP100)
SIO/UART	RXD0	PE1	13	11
	TXD0	PE0	12	10
	$\overline{\text{CTS0}}$/SCLK0	PE2	14	12
	RXD1	PA6	10	8
	TXD1	PA5	9	7
	$\overline{\text{CTS1}}$/SCLK1	PA4	8	6
	RXD2	PD6	38	36
	TXD2	PD5	37	35
	$\overline{\text{CTS2}}$/SCLK2	PD4	36	34
	RXD3	PF4	56	54
	TXD3	PF3	55	53
	$\overline{\text{CTS3}}$/SCLK3	PF2	54	52

Table 8.2 Signal connection list: TMRB

Reference peripheral function chapter	Function pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYFG (LQFP100)
TMRB	TB0IN	PA0	4	2
	TB0OUT	PA1	5	3
	TB1IN	PA2	6	4
	TB1OUT	PA3	7	5
	TB2IN	PE4	17	15
	TB2OUT	PE5	18	16
	TB3IN	PE6	19	17
	TB3OUT	PE7	20	18
	TB4IN	PA7	11	9
	TB4OUT	PE3	15	13
	TB5IN	PD0	32	30
	TB5OUT	PD1	33	31
	TB6IN	PA6	10	8
	TB6OUT	PA5	9	7
	TB7IN	PF0	52	50
	TB7OUT	PF1	53	51

Table 8.3 Signal connection list: ADC/AMP,CMP

Reference peripheral function chapter	Function pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYFG (LQFP100)
ADC	AINA0	PH0	98	96
	AINA1	PH1	97	95
	AINA2	PH2	96	94
	AINA3	PH3	95	93
	AINA4	PH4	94	92
	AINA5	PH5	93	91
	AINA6	PH6	92	90
	AINA7	PH7	91	89
	AINA8	PI0	90	88
	AINA9/AINB0	PI1	87	85
	AINA10/AINB1	PI2	86	84
	AINA11/AINB2	PI3	85	83
	AINB3	PJ0	82	80
	AINB4	PJ1	81	79
	AINB5	PJ2	80	78
	AINB6	PJ3	79	77
	AINB7	PJ4	78	76
	AINB8	PJ5	77	75
	AINB9	PJ6	76	74
	AINB10	PJ7	75	73
	AINB11	PK0	74	72
	AINB12	PK1	73	71
AMP/CMP	AINA9/AINB0	PI1	87	85
	AINA10/AINB1	PI2	86	84
	AINA11/AINB2	PI3	85	83
	AINB3	PJ0	82	80

Table 8.4 Signal connection list: INT/PMD

Reference peripheral function chapter	Function pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYFG (LQFP100)
INT	INT0	PH0	98	96
	INT1	PH1	97	95
	INT2	PH2	96	94
	INT3	PA0	4	2
	INT4	PA2	6	3
	INT5	PE4	17	15
	INT6	PE6	19	17
	INT7	PE7	20	18
	INT8	PA7	11	9
	INT9	PD3	35	33
	INTA	PL1	23	21
	INTB	PL0	22	20
	INTC	PJ6	76	74
	INTD	PJ7	75	73
	INTE	PK0	74	72
	INTF	PK1	73	71
PMD	UO0	PC0	24	22
	XO0	PC1	25	23
	VO0	PC2	26	24
	YO0	PC3	27	25
	WO0	PC4	28	26
	ZO0	PC5	29	27
	$\overline{\text{EMG0}}$	PC6	30	28
	$\overline{\text{OVV0}}$	PC7	31	29
	UO1	PG0	39	37
	XO1	PG1	40	38
	VO1	PG2	41	39
	YO1	PG3	44	42
	WO1	PG4	45	43
	ZO1	PG5	46	44
	$\overline{\text{EMG1}}$	PG6	47	45
	$\overline{\text{OVV1}}$	PG7	48	46

Table 8.5 Signal connection list: ENC/DEBUG/FLASH

Reference peripheral function chapter	Function pin name	Port name	TMPM370FYDFG (QFP100)	TMPM370FYFG (LQFP100)
ENC	ENCA0	PD0	32	30
	ENCB0	PD1	33	31
	ENCZ0	PD2	34	32
	ENCA1	PF2	54	52
	ENCB1	PF3	55	53
	ENCZ1	PF4	56	54
DEBUG	TMS/SWDIO	PB3	68	66
	TCK/SWCLK	PB4	69	67
	TDO/SWV	PB5	70	68
	TDI	PB6	71	69
	$\overline{\text{TRST}}$	PB7	72	70
	TRACECLK	PB0	65	63
	TRACEDATA0	PB1	66	64
	TRACEDATA1	PB2	67	65
FLASH	$\overline{\text{BOOT}}$	PF0	52	50

8.3. Registers

The following registers should be set appropriately to use the ports.

Each register is 32 bits. The configuration of the register depends on the number of bits of the port and its function assignment.

"x" and "n" in the following table show a port name and a function number, respectively.

Register Name		Type	Setting Value	Description
PxDATA	Data Register	R/W	"0" or "1"	Read from and write to a port.
PxCR	Output Control Register	R/W	0: Output disabled 1: Output enabled	Output control
PxFRn	Function Register n	R/W	0: PORT 1: Function	Function setting When this register is set to “1”, the assigned function becomes available. Each function assigned to a port has its own function register. When multiple functions are assigned to one port, only one function should be enabled.
PxOD	Open-Drain Control Register	R/W	0: CMOS 1: Open-drain	Programmable open-drain control The programmable open-drain is a pseudo open-drain. An output buffer is disabled when the output data is "1", which is set by PxOD = "1"
PxPUP	Pull-up Control Register	R/W	0: Pull-up disabled 1: Pull-up enabled	Programmable pull-up control
PxPDN	Pull-down Control Register	R/W	0: Pull-down disabled 1: Pull-down enabled	Programmable pull-down control
PxIE	Input Control Register	R/W	0: Input disabled 1: Input enabled	Input control

8.3.1. List of Register

When the bit which is assigned to no functions is read, "0" is returned. The write to the bit is no meaning.

Table 8.6 Register List (1/2)

Register Name	Address (Base+)	Port A	Port B	Port C	Port D	Port E	Port F
Data Register	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA	PFDATA
Output Control Register	0x0004	PACR	PBCR	PCCR	PDCR	PECR	PFCR
Function Register 1	0x0008	PAFR1	PBFR1	PCFR1	PDFR1	PEFR1	PFFR1
Function Register 2	0x000C	PAFR2	-	-	PDFR2	PEFR2	PFFR2
Function Register 3	0x0010	-	-	-	-	-	PFFR3
Open-Drain Control Register	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD	PFOD
Pull-up Control Register	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP	PFPUP
Pull-down Control Register	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN	PFPDN
Input Control Register	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE	PFIE

Note: Do not access the addresses described as "-".

Table 8.7 Register List (2/2)

Register Name	Address (Base+)	Port G	Port H	Port I	Port J	Port K	Port L
Data Register	0x0000	PGDATA	PHDATA	PIDATA	PJDATA	PKDATA	PLDATA
Output Control Register	0x0004	PGCR	PHCR	PICR	PJCR	PKCR	-
Function Register 1	0x0008	PGFR1	PHFR1	-	PJFR1	PKFR1	PLFR1
Function Register 2	0x000C	-	-	-	-	-	-
Function Register 3	0x0010	-	-	-	-	-	-
Open-Drain Control Register	0x0028	PGOD	PHOD	PIOD	PJOD	PKOD	-
Pull-up Control Register	0x002C	PGPUP	PHPUP	PIPUP	PJPUP	PKPUP	-
Pull-down Control Register	0x0030	PGPDN	PHPDN	PIPDN	PJPDN	PKPDN	-
Input Control Register	0x0038	PGIE	PHIE	PIIE	PJIE	PKIE	PLIE

Note: Do not access the addresses described as "-".

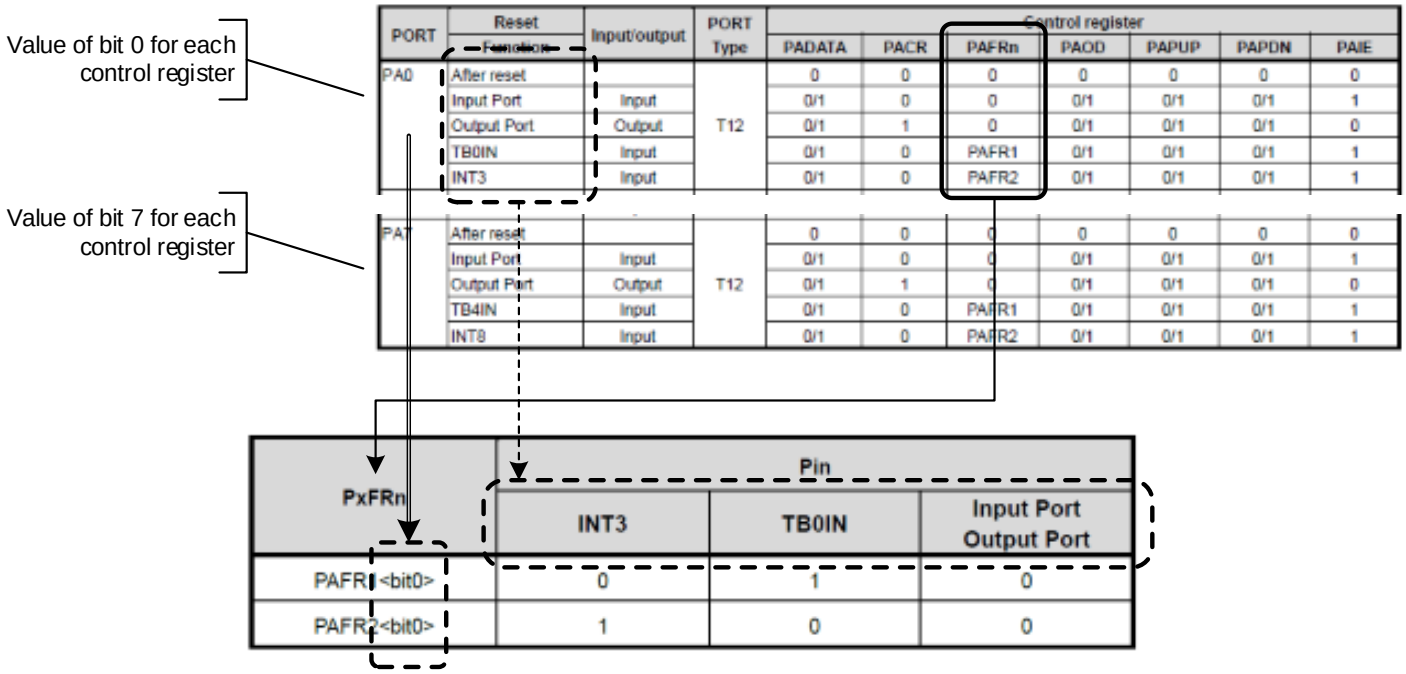
8.3.2. List of Port Functions and Settings

It is explained about viewpoint of a port register setting table.

The column of PxFRn shows the function register which should be set. When this register is set to "1", the corresponding function is enabled. ("x" is a port name and "n" is a function number.)

The bit which has the N/A in the table returns "0" when it is read. To write to the bit is no meaning.

"0" or "1" in the table shows the value which should be set. "0"/"1" means either value can be set.



PORT	Reset Function	Input/output	PORT Type	Control register PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB0IN	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
	INT3	Input		0/1	0	PAFR2	0/1	0/1	0/1	1
PA7	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB4IN	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
	INT8	Input		0/1	0	PAFR2	0/1	0/1	0/1	1

PxFRn	Pin: INT3	Pin: TB0IN	Pin: Input Port / Output Port
PAFR1 <bit0>	0	1	0
PAFR2 <bit0>	1	0	0

Note: The register value after reset is the initial value before the clock is enabled.

8.3.2.1. Setting of using the assigned pin

To use the assigned pins as output pins of the peripheral function, set the function register (PxFRn<bit m>=1) for peripheral function and enable output control register (PxCR<bit m>=1), then set the peripheral functions. When output is enabled before setting the function register, the data register value of the port is output until the function register is set.

To use the assigned pins as input pins of the peripheral function, set the input control register of the port (PxIE<bit m>=1) and set the function register (PxFRn<bit m>=1) for peripheral function, then set the peripheral functions.

To use the assigned pins input/output pins of the peripheral function such as SIO/UART, set the input control register of the port (PxIE<bit m>=1), set the peripheral function (PxFRn<bit m>=1) and set the output control register to output enable (PxCR<bit m>=1), then set the peripheral function.

- When multiple functions are assigned to the same pin, please choose only one function for usage.
- When same function is assigned to multiple pins, please use the function exclusively.

8.3.2.2. PORT A

Table 8.8 Port A registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB0IN	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
	INT3	Input		0/1	0	PAFR2	0/1	0/1	0/1	1
PA1	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB0OUT	Output		0/1	1	PAFR1	0/1	0/1	0/1	0
PA2	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB1IN	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
	INT4	Input		0/1	0	PAFR2	0/1	0/1	0/1	1
PA3	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB1OUT	Output		0/1	1	PAFR1	0/1	0/1	0/1	0
PA4	After reset		T9	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SCLK1	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
		Output		0/1	1		0/1	0/1	0/1	0
	$\overline{\text{CTS1}}$	Input		0/1	0	PAFR2	0/1	0/1	0/1	1
PA5	After reset		T13	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TXD1	Output		0/1	1	PAFR1	0/1	0/1	0/1	0
	TB6OUT	Output		0/1	1	PAFR2	0/1	0/1	0/1	0
PA6	After reset		T11	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	RXD1	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
	TB6IN	Input		0/1	0	PAFR2	0/1	0/1	0/1	1
PA7	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB4IN	Input		0/1	0	PAFR1	0/1	0/1	0/1	1
	INT8	Input		0/1	0	PAFR2	0/1	0/1	0/1	1

8.3.2.3. PORT B

Table 8.9 Port B registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0	After reset		T18	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TRACECLK	Output		0/1	1	PBFR1	0	0	0	0
PB1	After reset		T18	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TRACEDATA0	Output		0/1	1	PBFR1	0	0	0	0
PB2	After reset		T18	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TRACEDATA1	Output		0/1	1	PBFR1	0	0	0	0
PB3	After reset (TMS/SWDIO)	I/O	T6	0	1(Note1)	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PB4	After reset (TCK/SWCLK)	Input	T8	0	0	PBFR1	0	0	1	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PB5	After reset (TDO/SWV)	Output	T19	0	1(Note1)	PBFR1	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PB6	After reset (TDI)	Input	T7	0	0	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
PB7	After reset ($\overline{\text{TRST}}$)	Input	T7 (Note2)	0	0	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0

Note1: When receiving the command from TOOL, it becomes output.

Note2: A noise filter (30ns typ.) is added.

8.3.2.4. PORT C

Table 8.10 Port C registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE
PC0	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	UO0	Output		0/1	1	PCFR1	0/1	0/1	0/1	0
PC1	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	XO0	Output		0/1	1	PCFR1	0/1	0/1	0/1	0
PC2	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	VO0	Output		0/1	1	PCFR1	0/1	0/1	0/1	0
PC3	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	YO0	Output		0/1	1	PCFR1	0/1	0/1	0/1	0
PC4	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	WO0	Output		0/1	1	PCFR1	0/1	0/1	0/1	0
PC5	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ZO0	Output		0/1	1	PCFR1	0/1	0/1	0/1	0
PC6	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	$\overline{\text{EMG0}}$	Input		0/1	0	PCFR1	0/1	0/1	0/1	1
PC7	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	$\overline{\text{OVV0}}$	Input		0/1	0	PCFR1	0/1	0/1	0/1	1

8.3.2.5. PORT D

Table 8.11 Port D registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE
PD0	After reset		T11	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ENCA0	Input		0/1	0	PDFR1	0/1	0/1	0/1	1
	TBIN5	Input		0/1	0	PDFR2	0/1	0/1	0/1	1
PD1	After reset		T10	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ENCB0	Input		0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5OUT	Output		0/1	1	PDFR2	0/1	0/1	0/1	0
PD2	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ENCZ0	Input		0/1	0	PDFR1	0/1	0/1	0/1	1
PD3	After reset		T4	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INT9	Input		0/1	0	PDFR1	0/1	0/1	0/1	1
PD4	After reset		T9	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SCLK2	Input		0/1	0	PDFR1	0/1	0/1	0/1	1
		Output		0/1	1		0/1	0/1	0/1	0
	$\overline{\text{CTS2}}$	Input		0/1	0	PDFR2	0/1	0/1	0/1	1
PD5	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TXD2	Output		0/1	1	PDFR1	0/1	0/1	0/1	0
PD6	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	RXD2	Input		0/1	0	PDFR1	0/1	0/1	0/1	1

8.3.2.6. PORT E

Table 8.12 Port E registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TXD0	Output		0/1	1	PEFR1	0/1	0/1	0/1	0
PE1	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	RXD0	Input		0/1	0	PEFR1	0/1	0/1	0/1	1
PE2	After reset		T9	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SCLK0	Input		0/1	0	PEFR1	0/1	0/1	0/1	1
		Output		0/1	1		0/1	0/1	0/1	0
	$\overline{\text{CTS0}}$	Input		0/1	0	PEFR2	0/1	0/1	0/1	1
PE3	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB4OUT	Output		0/1	1	PEFR1	0/1	0/1	0/1	0
PE4	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB2IN	Input		0/1	0	PEFR1	0/1	0/1	0/1	1
	INT5	Input		0/1	0	PEFR2	0/1	0/1	0/1	1
PE5	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB2OUT	Output		0/1	1	PEFR1	0/1	0/1	0/1	0
PE6	After reset		T12	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB3IN	Input		0/1	0	PEFR1	0/1	0/1	0/1	1
	INT6	Input		0/1	0	PEFR2	0/1	0/1	0/1	1
PE7	After reset		T14	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB3OUT	Output		0/1	1	PEFR1	0/1	0/1	0/1	0
	INT7	Input		0/1	0	PEFR2	0/1	0/1	0/1	1

8.3.2.7. PORT F

Table 8.13 Port F registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF0	During reset ($\overline{\text{BOOT}}$) (Note)	Input	T20	0	0	0	0	1	0	0
	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB7IN	Input		0/1	0	PFFR1	0/1	0/1	0/1	1
PF1	After reset		T2	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB7OUT	Output		0/1	1	PFFR1	0/1	0/1	0/1	0
PF2	After reset		T15	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ENCA1	Input		0/1	0	PFFR1	0/1	0/1	0/1	1
	SCLK3	Input		0/1	0	PFFR2	0/1	0/1	0/1	1
		Output		0/1	1		0/1	0/1	0/1	0
	$\overline{\text{CTS3}}$	Input		0/1	0	PFFR3	0/1	0/1	0/1	1
PF3	After reset		T10	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ENCB1	Input		0/1	0	PFFR1	0/1	0/1	0/1	1
	TXD3	Output		0/1	1	PFFR2	0/1	0/1	0/1	0
PF4	After reset		T11	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ENCZ1	Input		0/1	0	PFFR1	0/1	0/1	0/1	1
	RXD3	Input		0/1	0	PFFR2	0/1	0/1	0/1	1

Note: PFPUP is enabled during reset by the reset pin ($\overline{\text{RESET}}$). PFIE is "0", but $\overline{\text{BOOT}}$ signal can be input.

8.3.2.8. PORT G

Table 8.14 Port G registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PG0	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	UO1	Output		0/1	1	PGFR1	0/1	0/1	0/1	0
PG1	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	XO1	Output		0/1	1	PGFR1	0/1	0/1	0/1	0
PG2	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	VO1	Output		0/1	1	PGFR1	0/1	0/1	0/1	0
PG3	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	YO1	Output		0/1	1	PGFR1	0/1	0/1	0/1	0
PG4	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	WO1	Output		0/1	1	PGFR1	0/1	0/1	0/1	0
PG5	After reset		T1	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	ZO1	Output		0/1	1	PGFR1	0/1	0/1	0/1	0
PG6	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	$\overline{\text{EMG1}}$	Input		0/1	0	PGFR1	0/1	0/1	0/1	1
PG7	After reset		T3	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	$\overline{\text{OVV1}}$	Input		0/1	0	PGFR1	0/1	0/1	0/1	1

8.3.2.9. PORT H

Table 8.15 Port H registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PHDATA	PHCR	PHFRn	PHOD	PHPUP	PHPDN	PHIE
PH0	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INT0	Input		0/1	0	PHFR1	0/1	0/1	0/1	1
	AINA0	Input		0/1	0	0	0/1	0	0	0
PH1	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INT1	Input		0/1	0	PHFR1	0/1	0/1	0/1	1
	AINA1	Input		0/1	0	0	0/1	0	0	0
PH2	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INT2	Input		0/1	0	PHFR1	0/1	0/1	0/1	1
	AINA2	Input		0/1	0	0	0/1	0	0	0
PH3	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA3	Input		0/1	0	N/A	0/1	0	0	0
PH4	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA4	Input		0/1	0	N/A	0/1	0	0	0
PH5	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA5	Input		0/1	0	N/A	0/1	0	0	0
PH6	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA6	Input		0/1	0	N/A	0/1	0	0	0
PH7	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA7	Input		0/1	0	N/A	0/1	0	0	0

8.3.2.10. PORT I

Table 8.16 Port I registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PIDATA	PICR	PIFRn	PIOD	PIPUP	PIPDN	PIIE
PI0	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA8	Input		0/1	0	N/A	0/1	0	0	0
PI1	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA9/AINB0	Input		0/1	0	N/A	0/1	0	0	0
PI2	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA10/AINB1	Input		0/1	0	N/A	0/1	0	0	0
PI3	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINA11/AINB2	Input		0/1	0	N/A	0/1	0	0	0

8.3.2.11. PORT J

Table 8.17 Port J registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ0	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINB3	Input		0/1	0	N/A	0/1	0	0	0
PJ1	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINB4	Input		0/1	0	N/A	0/1	0	0	0
PJ2	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINB5	Input		0/1	0	N/A	0/1	0	0	0
PJ3	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINB6	Input		0/1	0	N/A	0/1	0	0	0
PJ4	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINB7	Input		0/1	0	N/A	0/1	0	0	0
PJ5	After reset		T16	0	0	N/A	0	0	0	0
	Input Port	Input		0/1	0	N/A	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	N/A	0/1	0/1	0/1	0
	AINB8	Input		0/1	0	N/A	0/1	0	0	0
PJ6	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INTC	Input		0/1	0	PJFR1	0/1	0/1	0/1	1
	AINB9	Input		0/1	0	0	0/1	0	0	0
PJ7	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INTD	Input		0/1	0	PJFR1	0/1	0/1	0/1	1
	AINB10	Input		0/1	0	0	0/1	0	0	0

8.3.2.12. PORT K

Table 8.18 Port K registers setting

PORT	Reset Function	Input/output	PORT Type	Control register						
				PKDATA	PKCR	PKFRn	PKOD	PKPUP	PKPDN	PKIE
PK0	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INTE	Input		0/1	0	PKFR1	0/1	0/1	0/1	1
	AINB11	Input		0/1	0	0	0/1	0	0	0
PK1	After reset		T17	0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	INTF	Input		0/1	0	PKFR1	0/1	0/1	0/1	1
	AINB12	Input		0/1	0	0	0/1	0	0	0

8.3.2.13. PORT L

Table 8.19 Port L registers setting

PORT	Reset Function	Input/Output	PORT Type	Control register						
				PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE
PL0	After reset		T5	N/A	N/A	0	N/A	N/A	N/A	0
	Input Port	Input		N/A	N/A	0	N/A	N/A	N/A	1
	Output Port	Output		N/A	N/A	0	N/A	N/A	N/A	0
	INTB	Input		N/A	N/A	PLFR1	N/A	N/A	N/A	1
PL1	After reset		T5	N/A	N/A	0	N/A	N/A	N/A	0
	Input Port	Input		N/A	N/A	0	N/A	N/A	N/A	1
	Output Port	Output		N/A	N/A	0	N/A	N/A	N/A	0
	INTA	Input		N/A	N/A	PLFR1	N/A	N/A	N/A	1

8.4. Block Diagrams of Ports

The port has T1 to T21 types. Each circuit diagram is shown in the following page.

The "I/O Reset" shown in the circuit diagram is described the power on reset (POR) or the reset pin. Although, "I/O Reset" of debug pins (TMS/SWDIO, TDI, TDO/SWV, TCK/SWCLK, $\overline{\text{TRST}}$) is the power on reset (POR) only.

8.4.1. Type T1

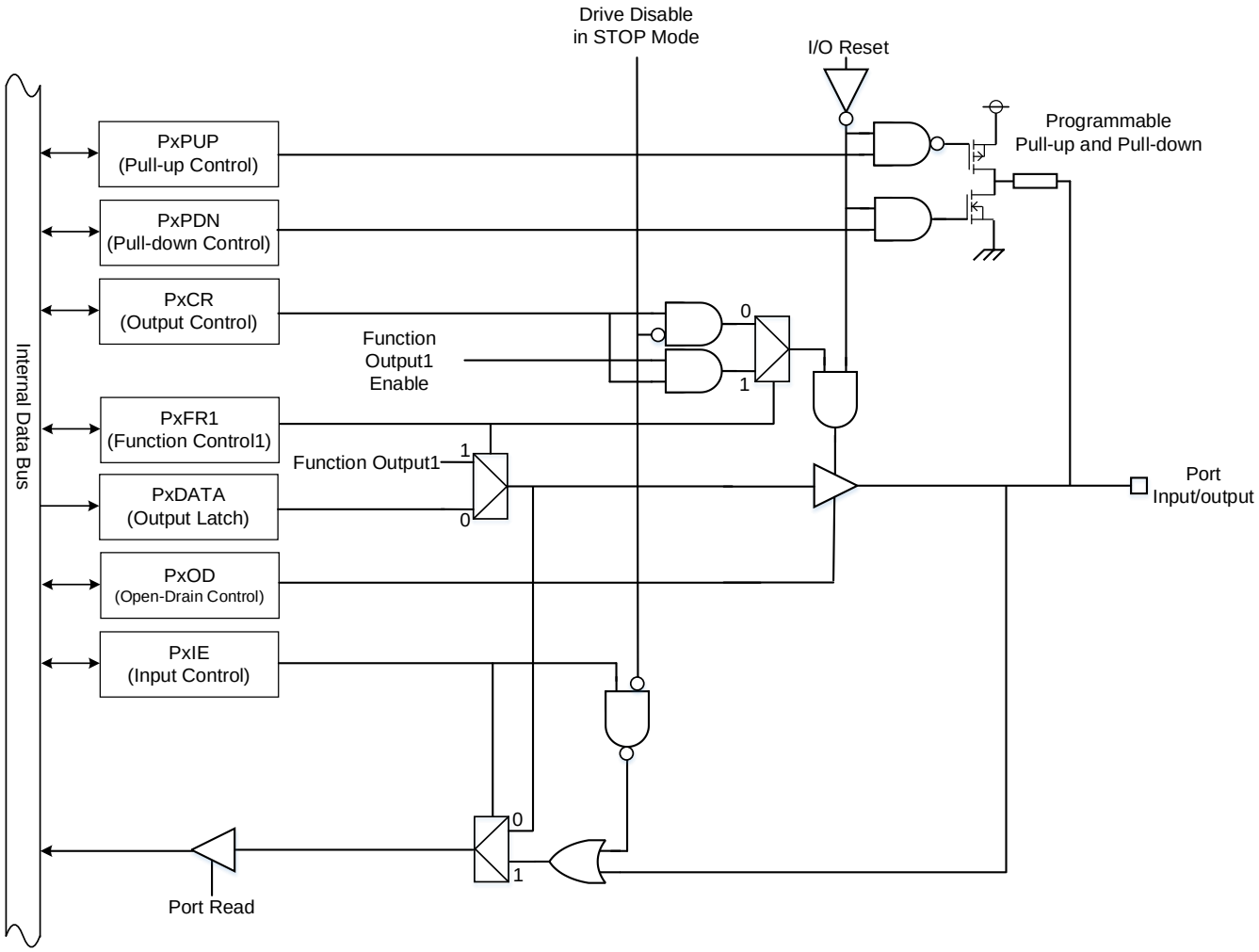


Figure 8.1 Port Type T1

8.4.2. Type T2

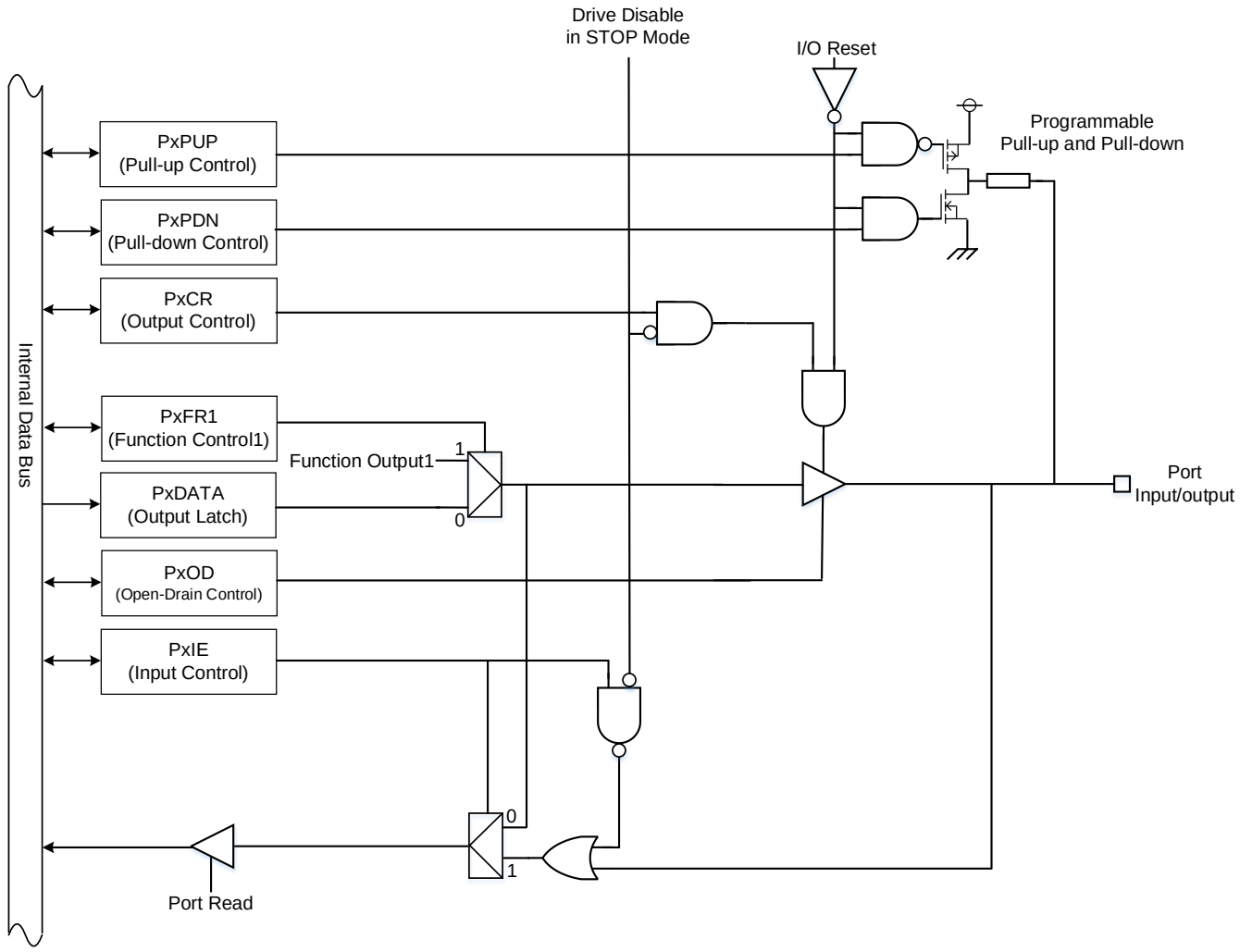


Figure 8.2 Port Type T2

8.4.3. Type T3

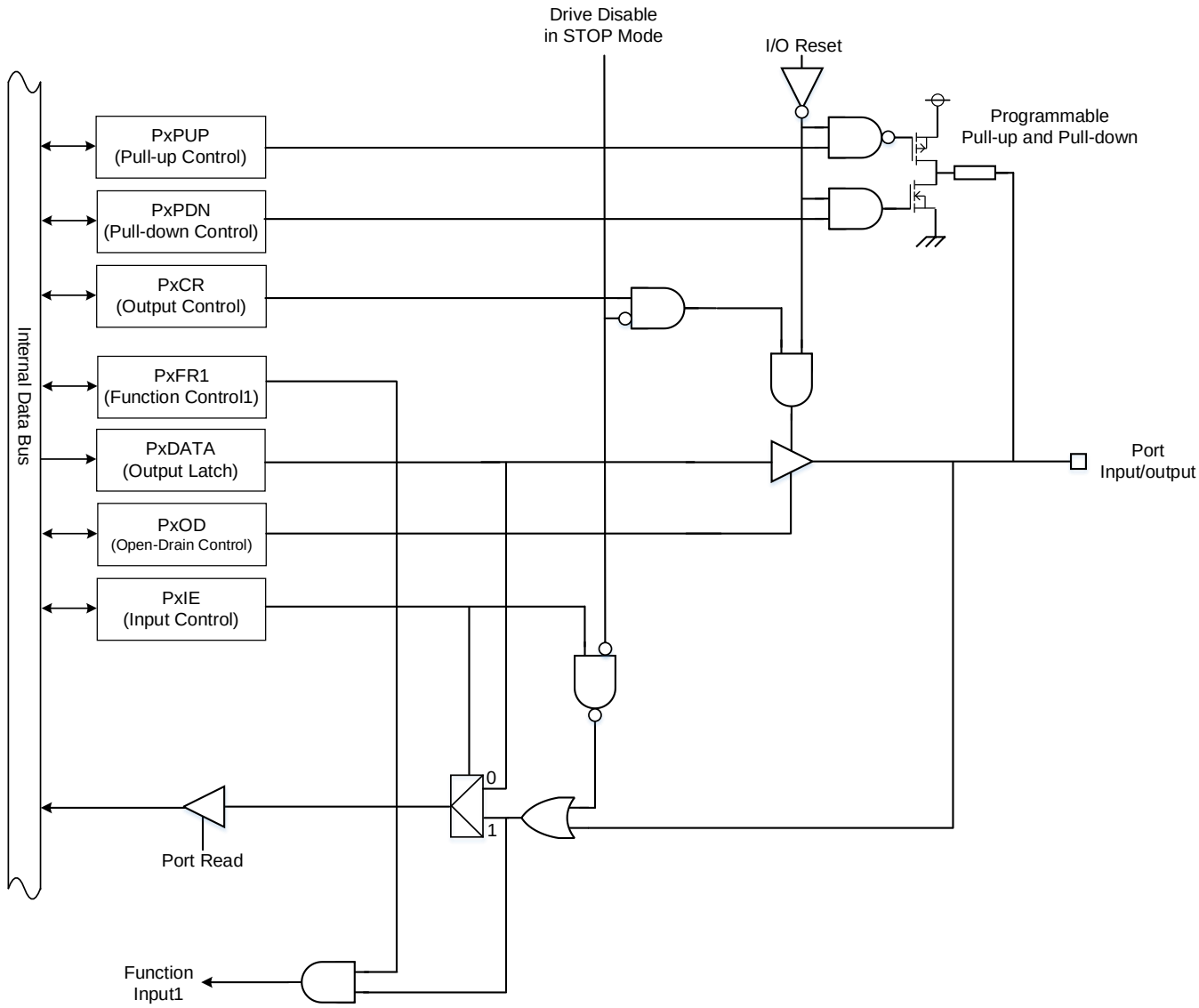


Figure 8.3 Port Type T3

8.4.4. Type T4

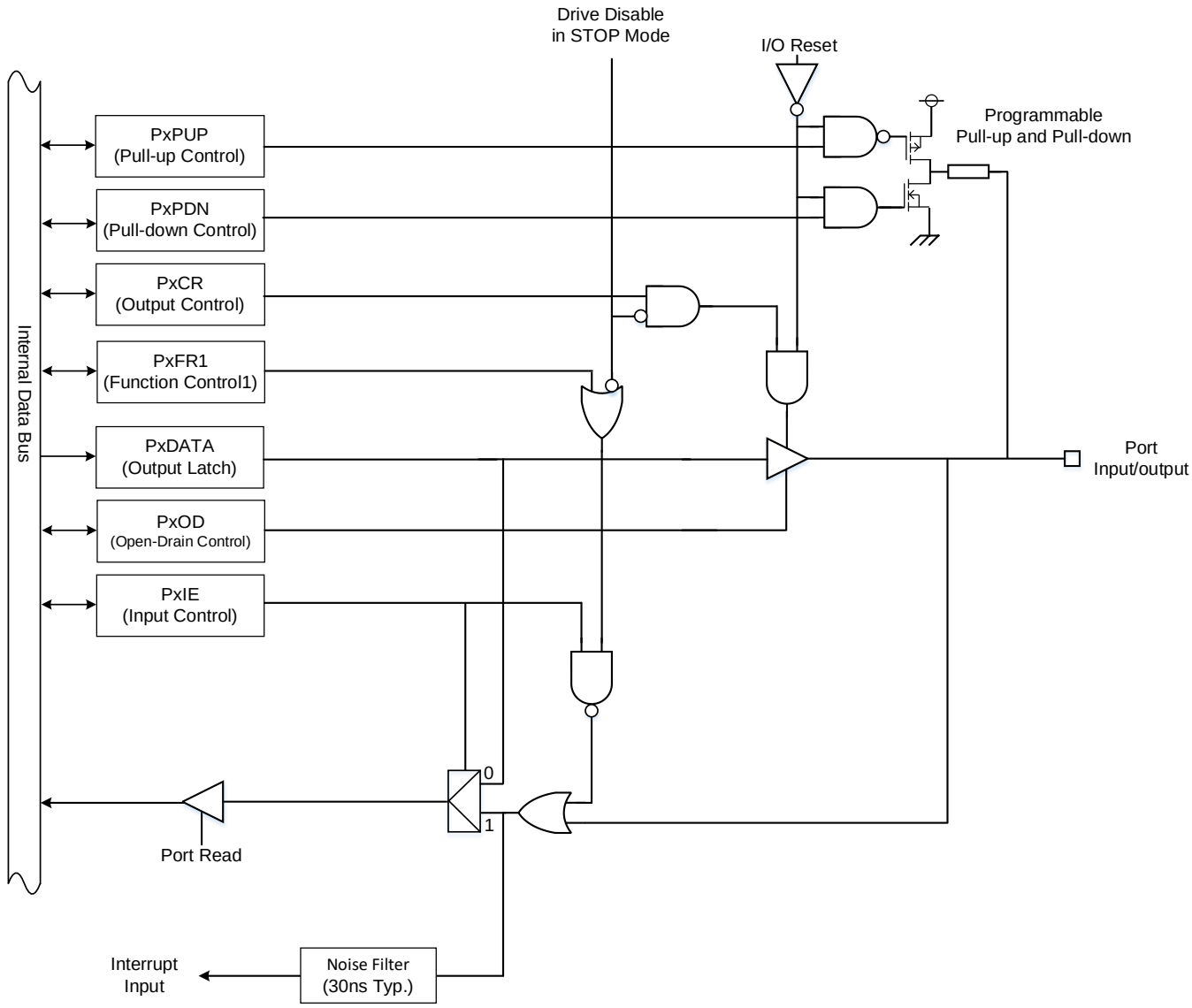


Figure 8.4 Port Type T4

8.4.5. Type T5

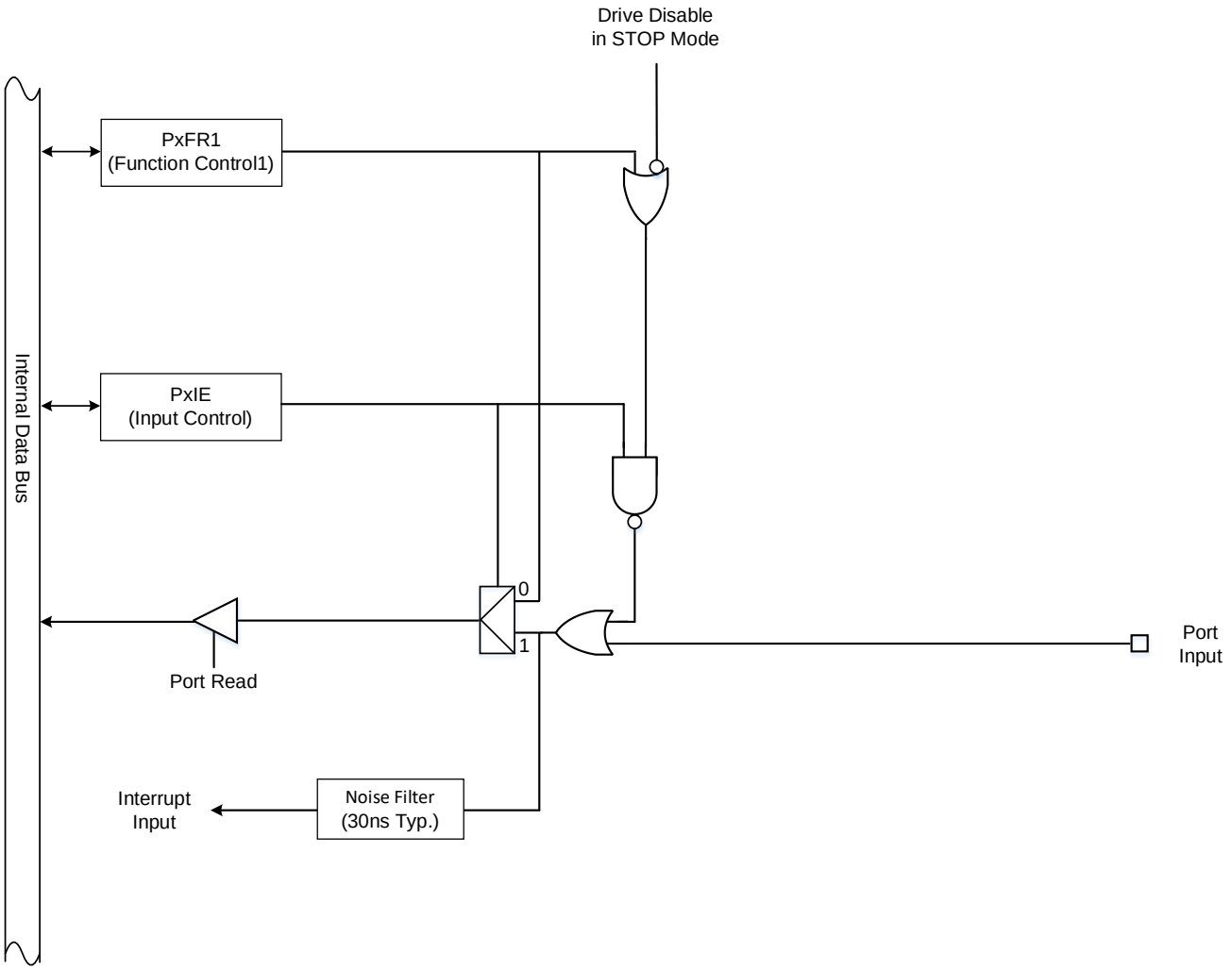


Figure 8.5 Port Type T5

8.4.6. Type T6

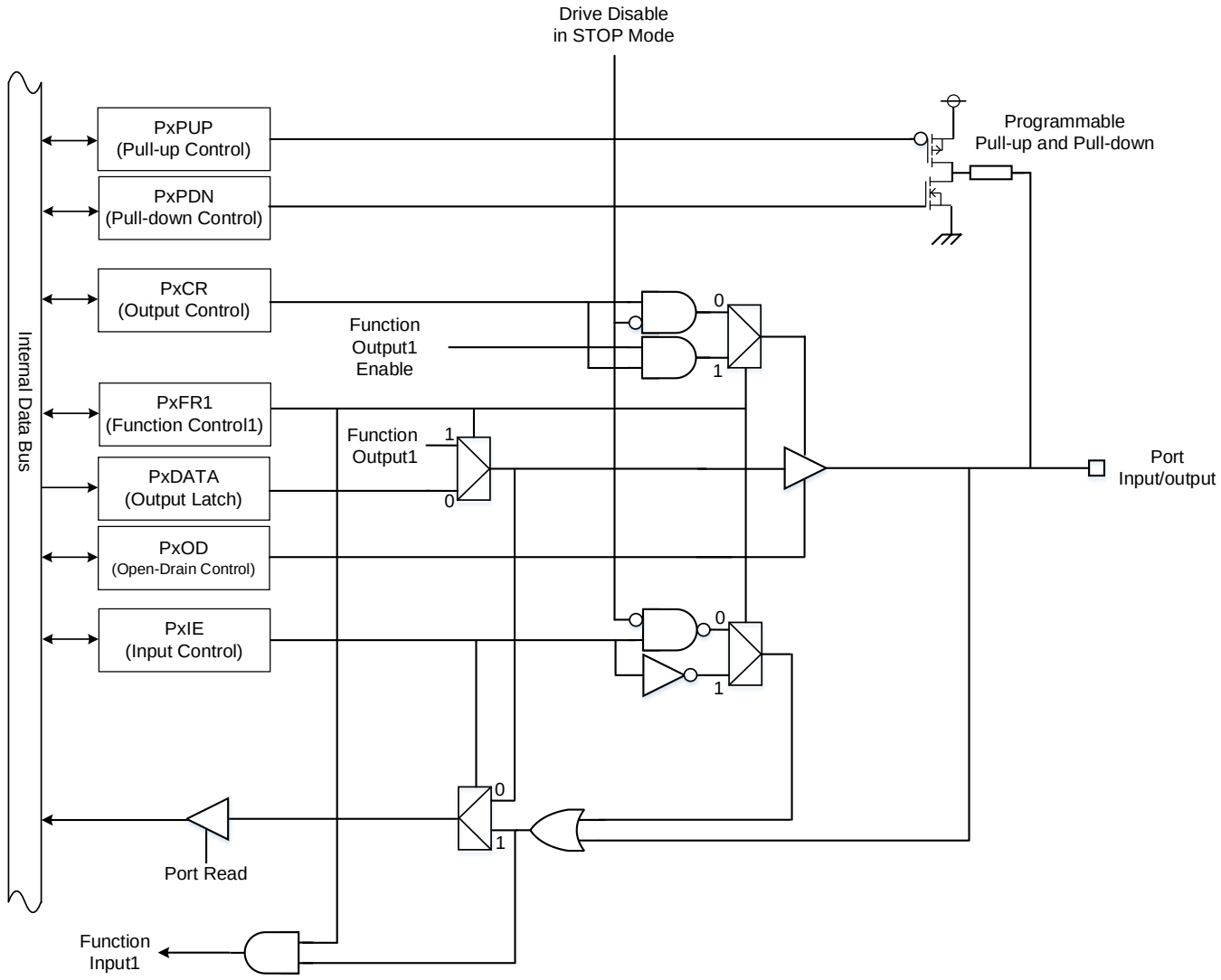


Figure 8.6 Port Type T6

8.4.7. Type T7

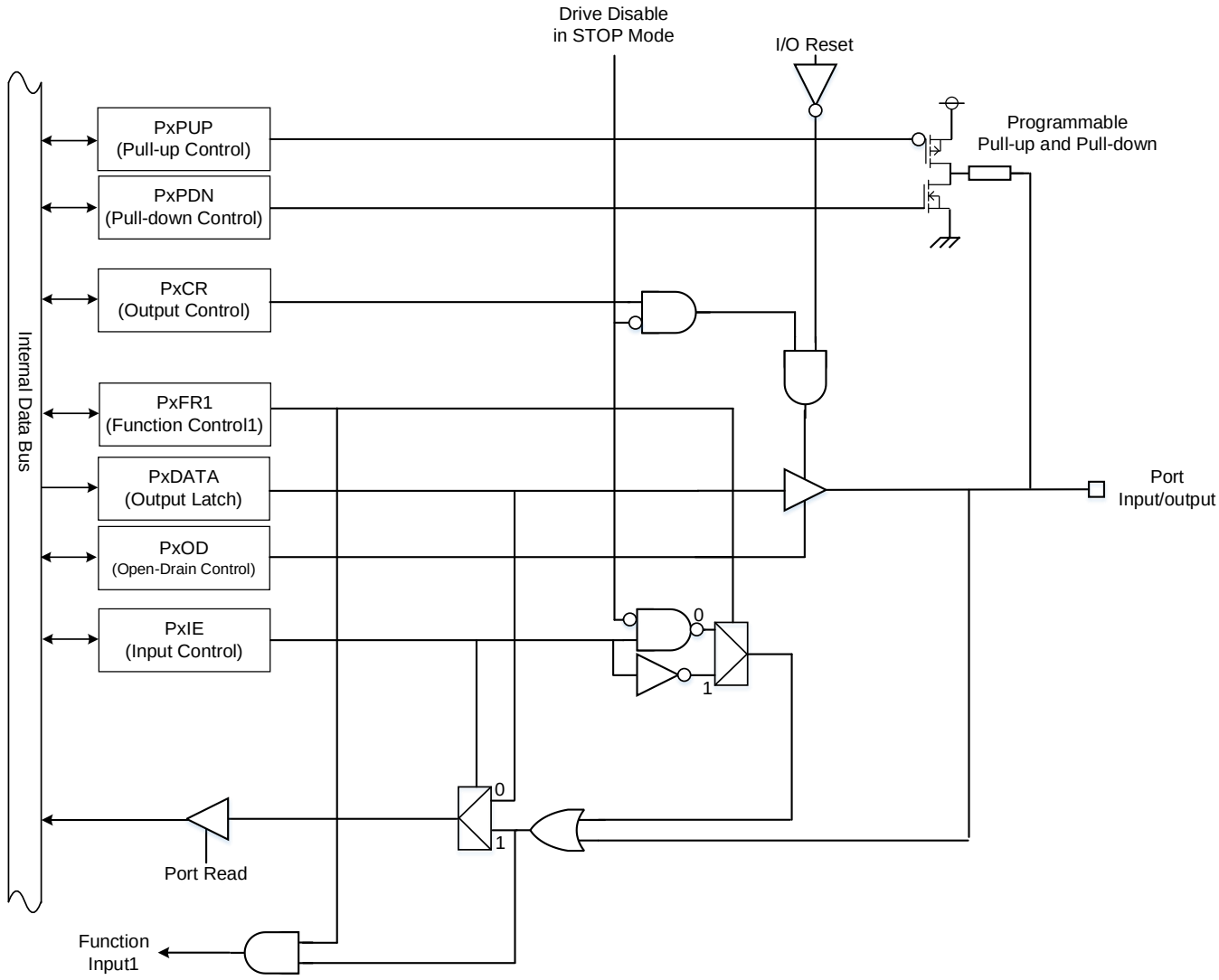


Figure 8.7 Port Type T7

8.4.8. Type T8

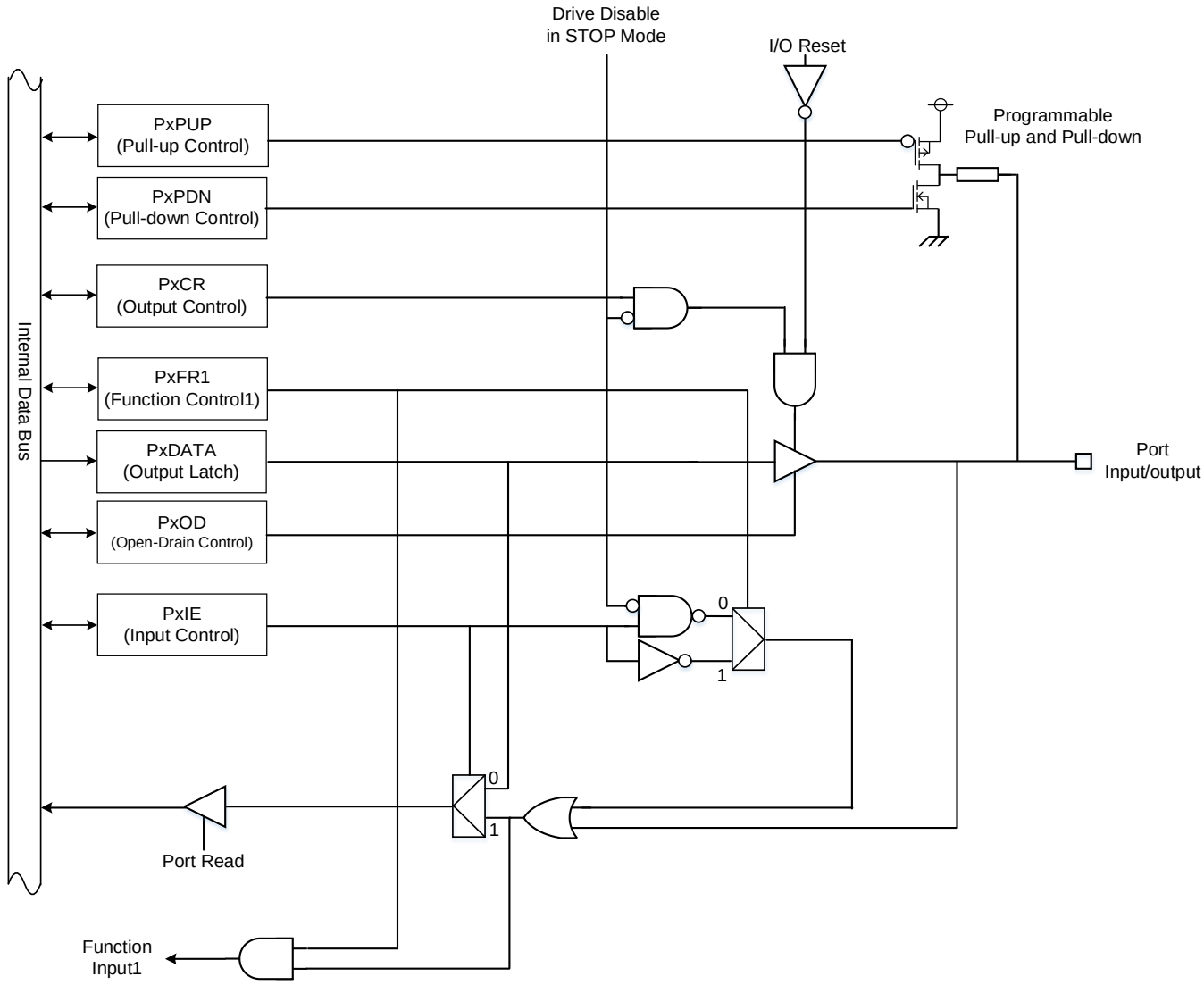


Figure 8.8 Port Type T8

8.4.9. Type T9

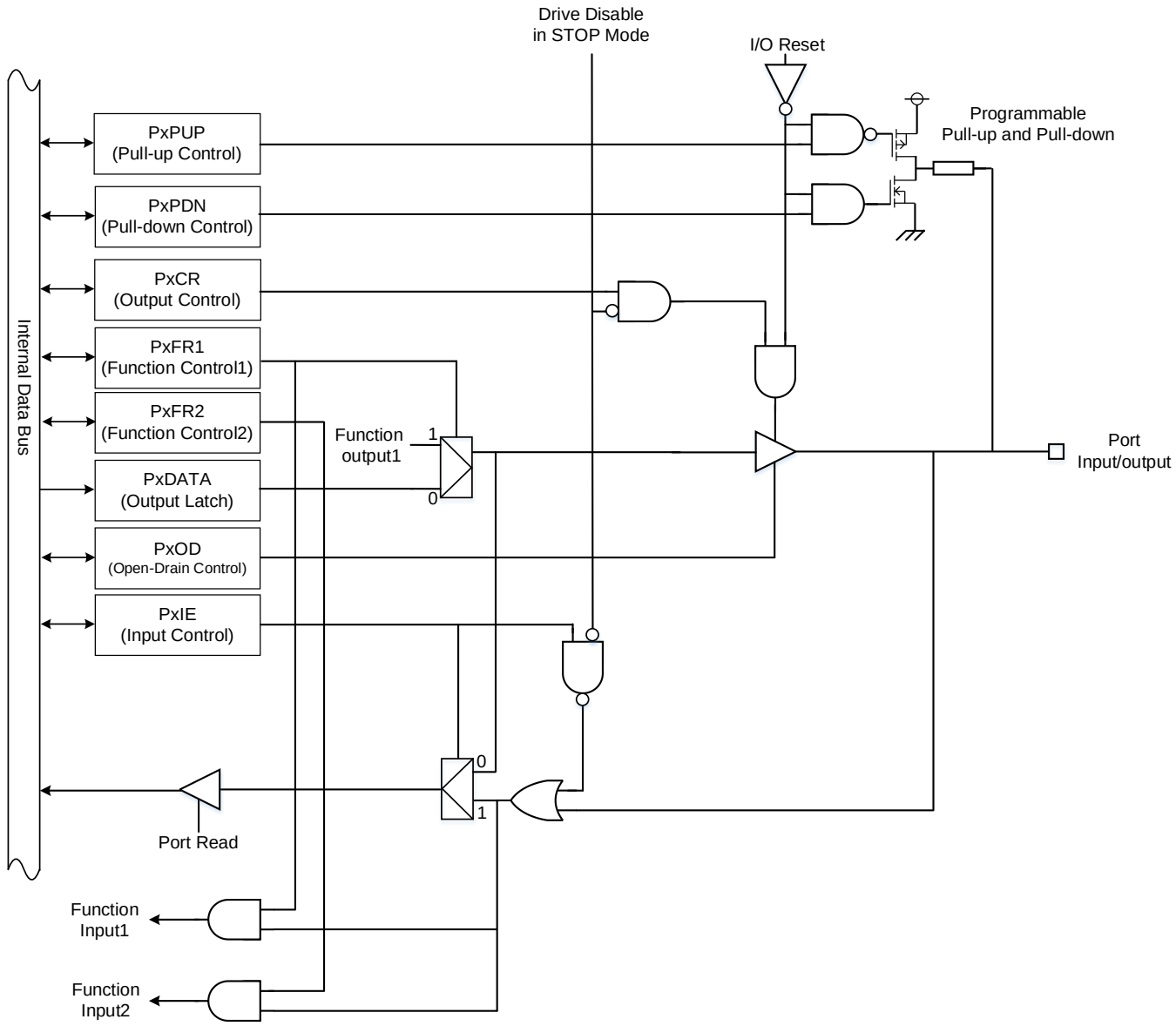


Figure 8.9 Port Type T9

8.4.10. Type T10

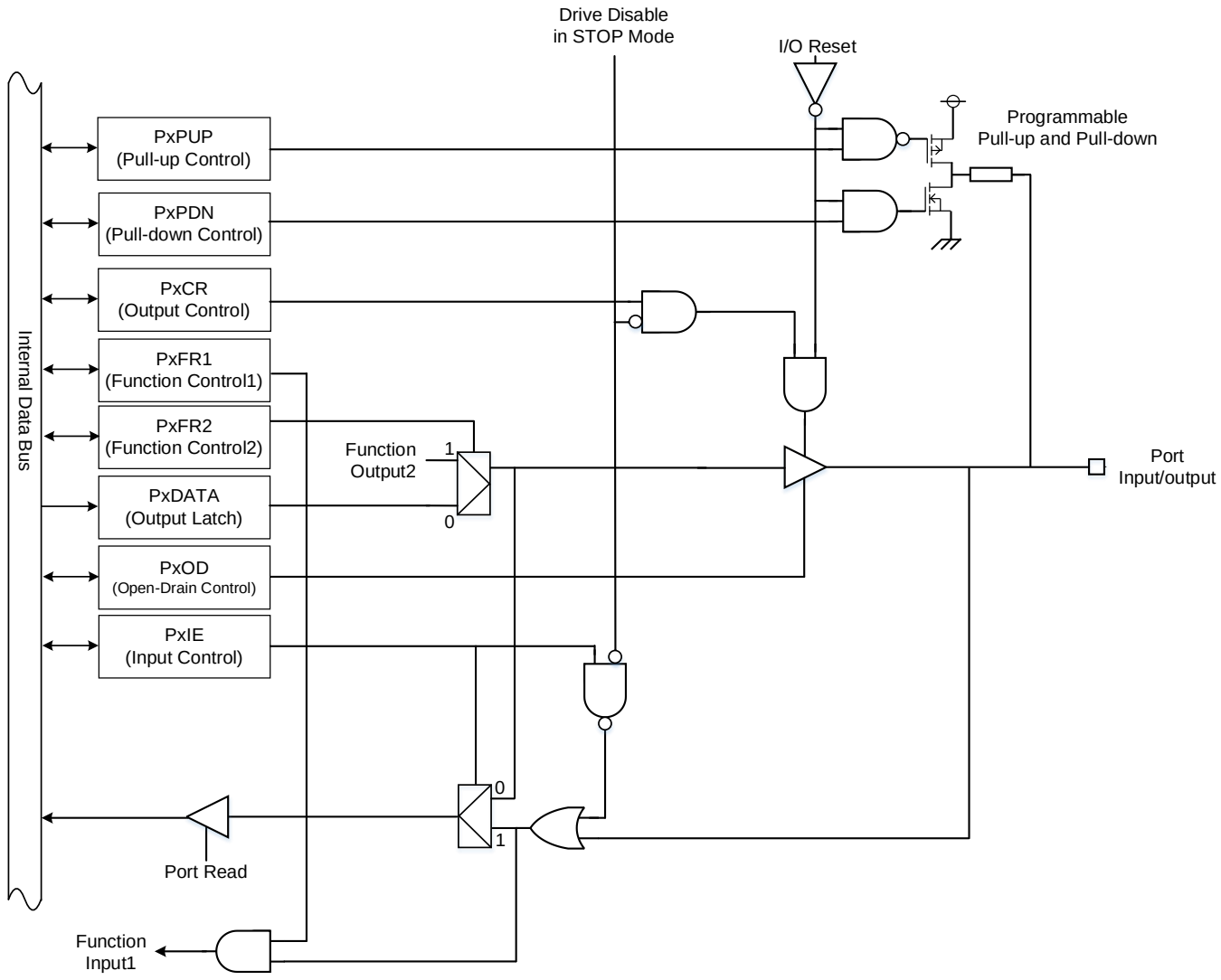


Figure 8.10 Port Type T10

8.4.11. Type T11

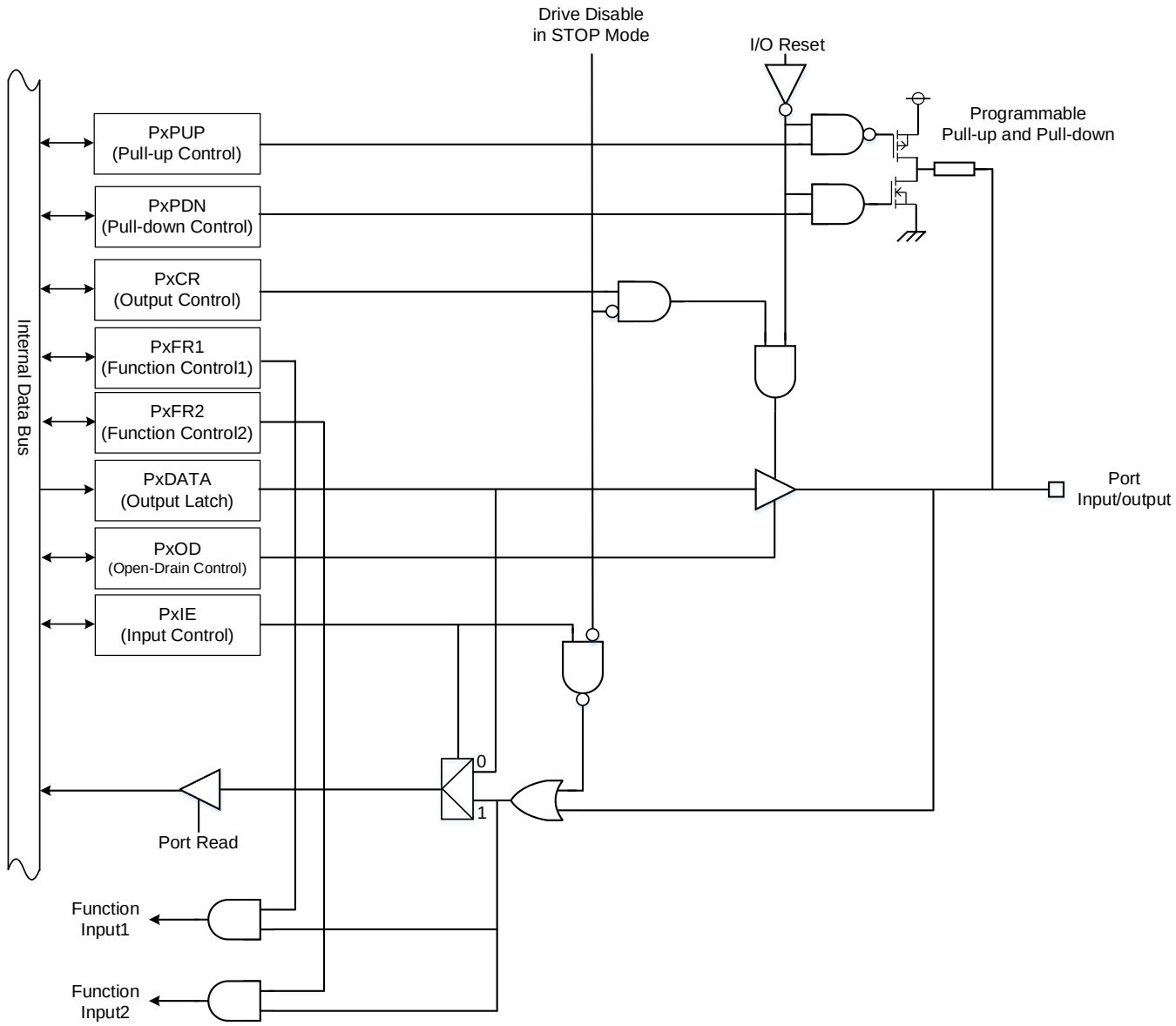


Figure 8.11 Port Type T11

8.4.12. Type T12

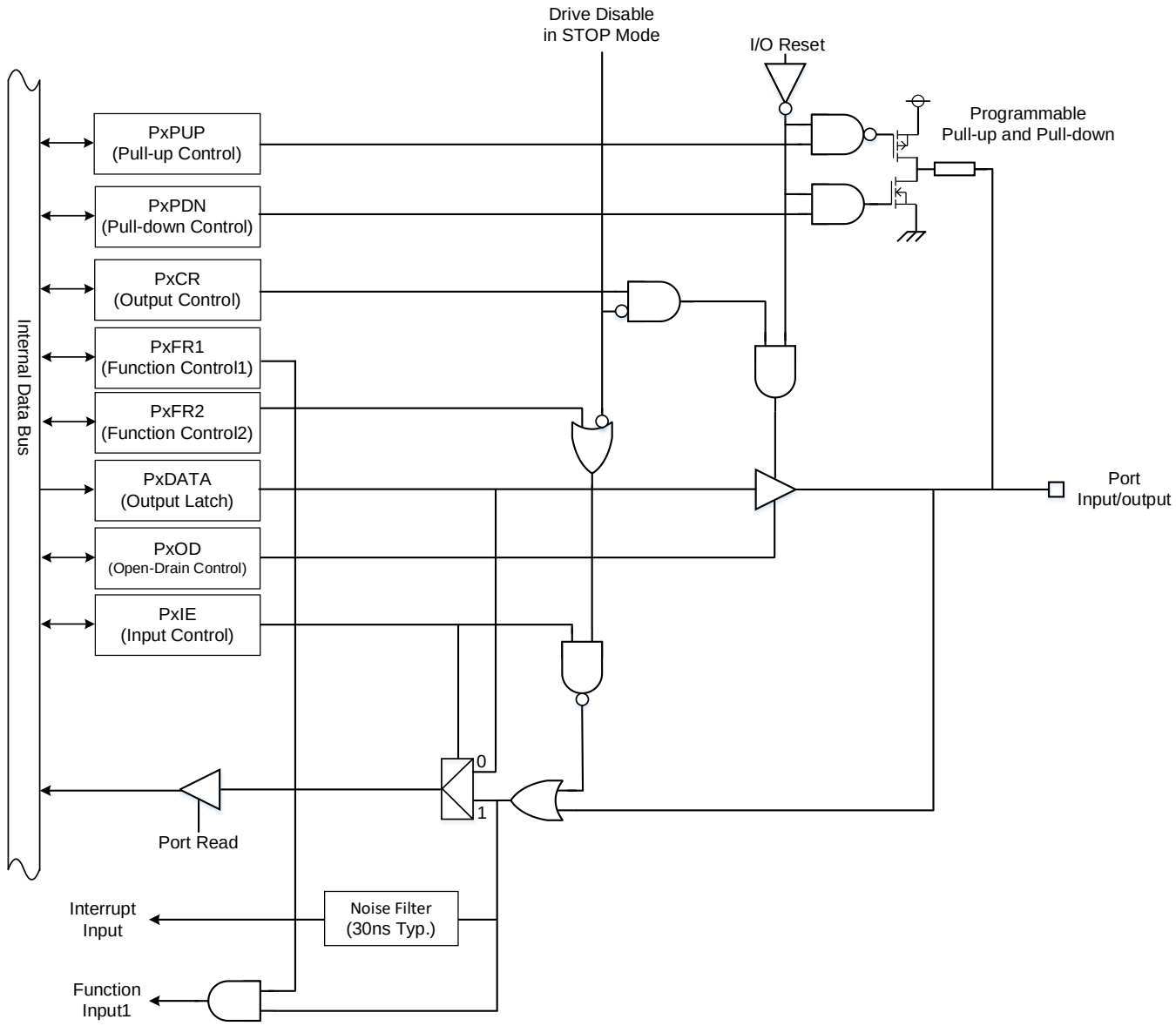


Figure 8.12 Port Type T12

8.4.13. Type T13

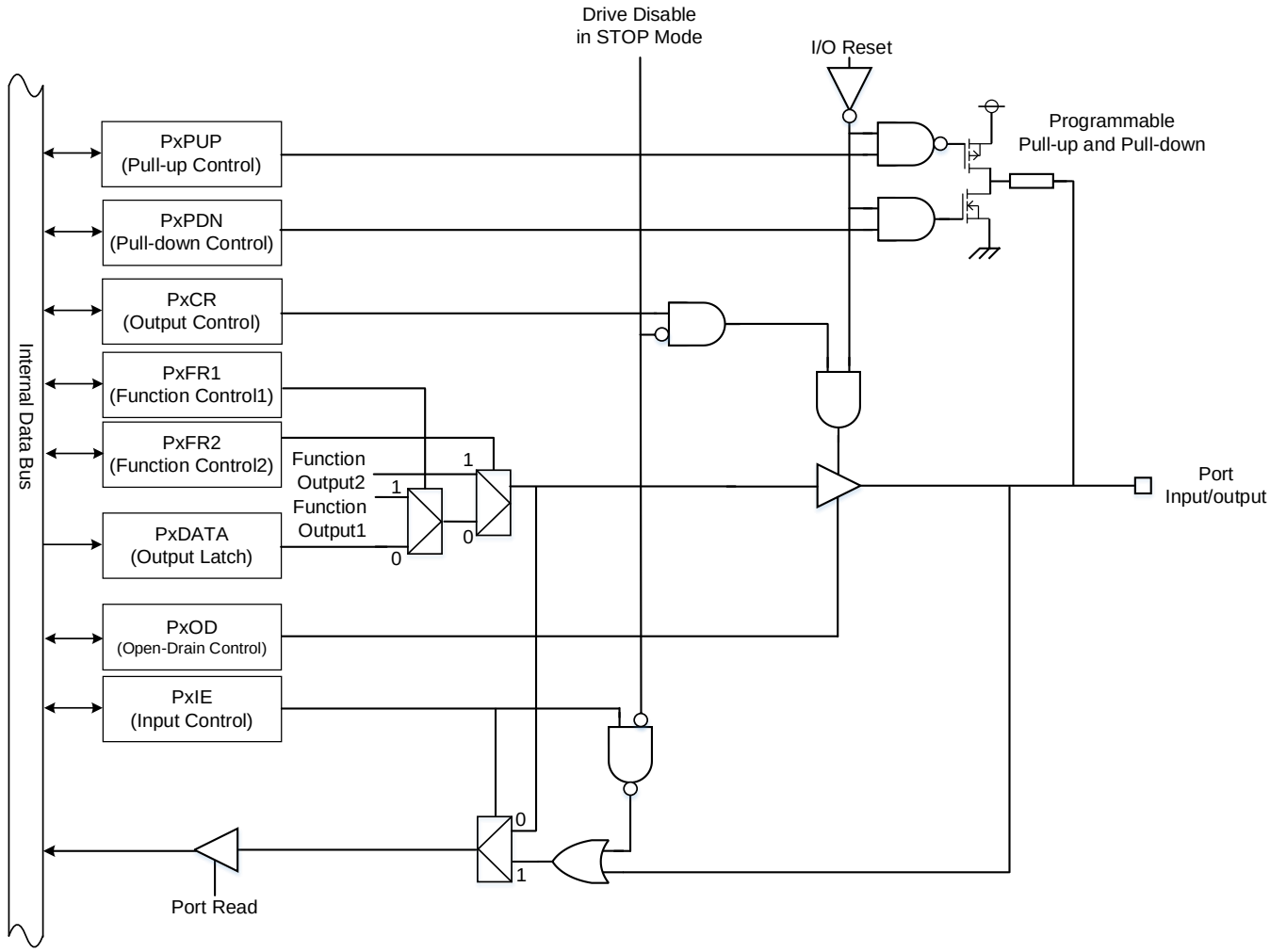


Figure 8.13 Port Type T13

8.4.14. Type T14

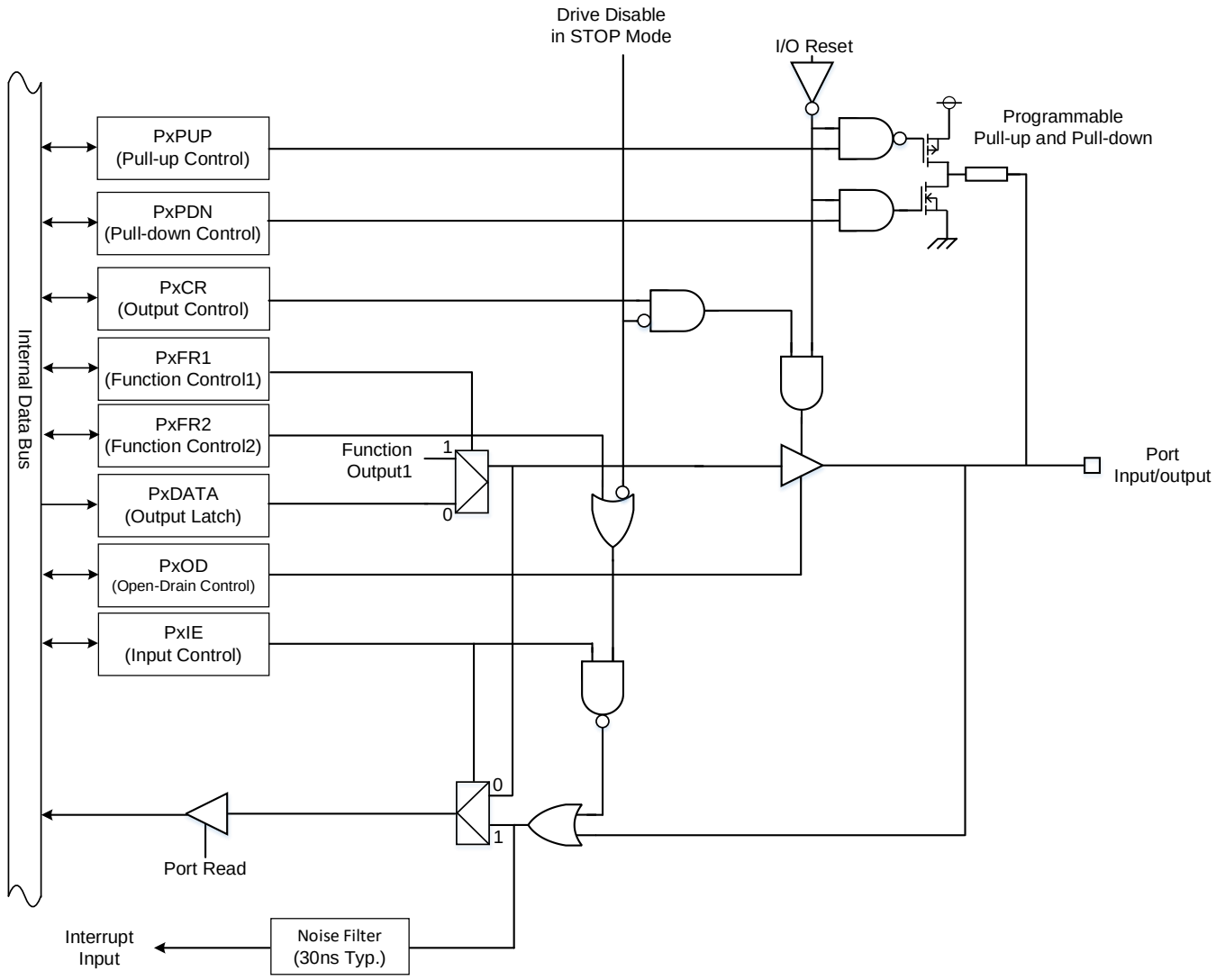


Figure 8.14 Port Type T14

8.4.15. Type T15

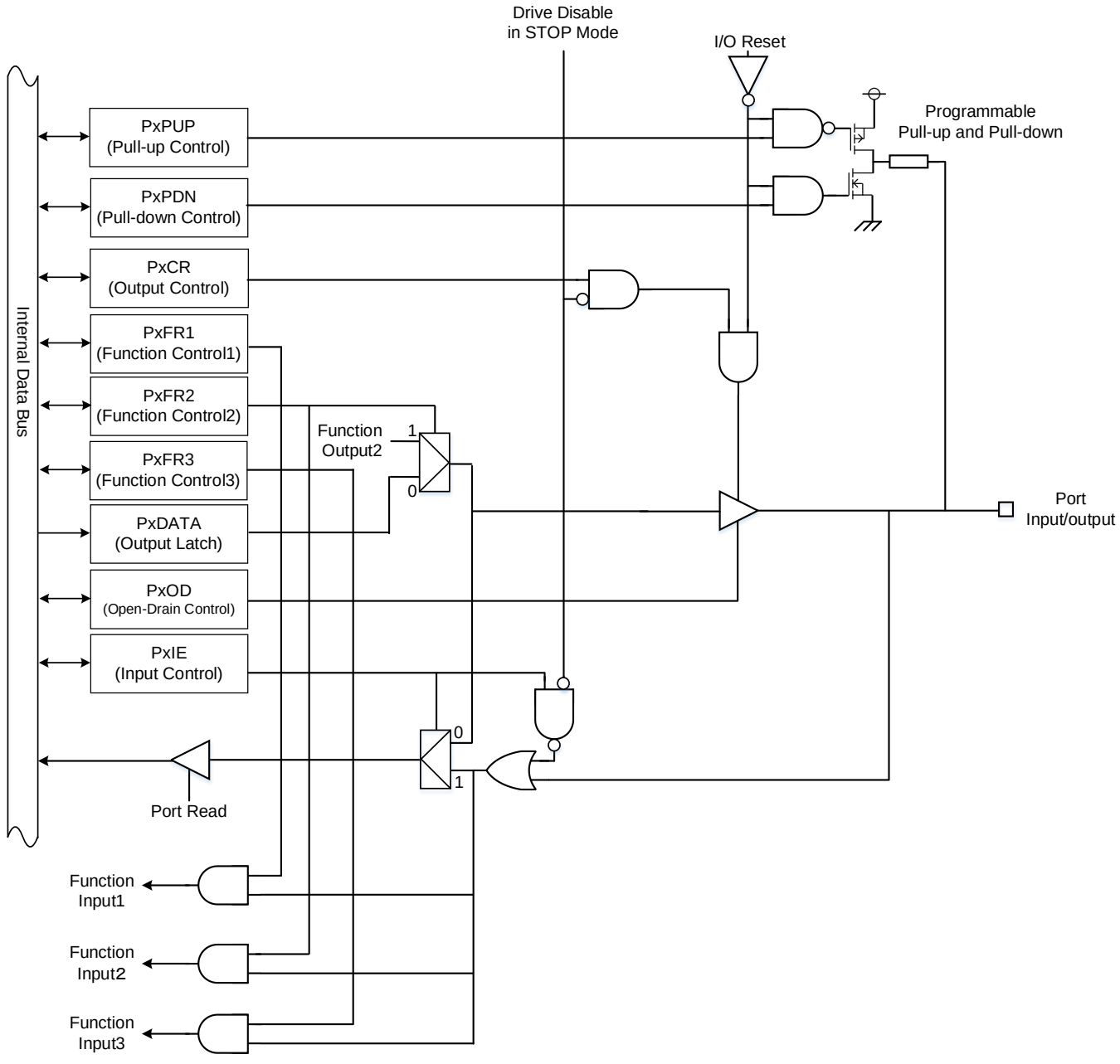


Figure 8.15 Port Type T15

8.4.16. Type T16

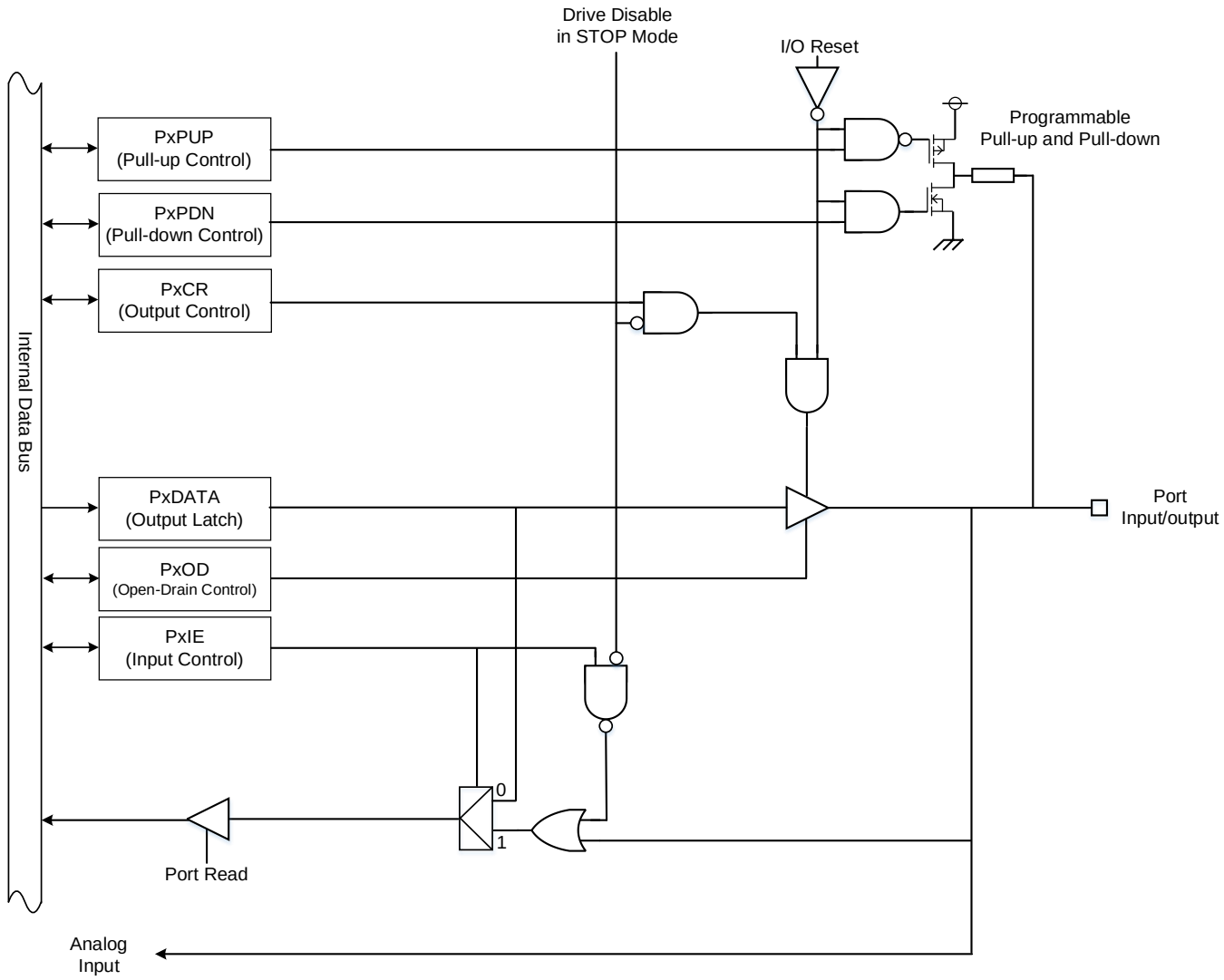


Figure 8.16 Port Type T16

8.4.17. Type T17

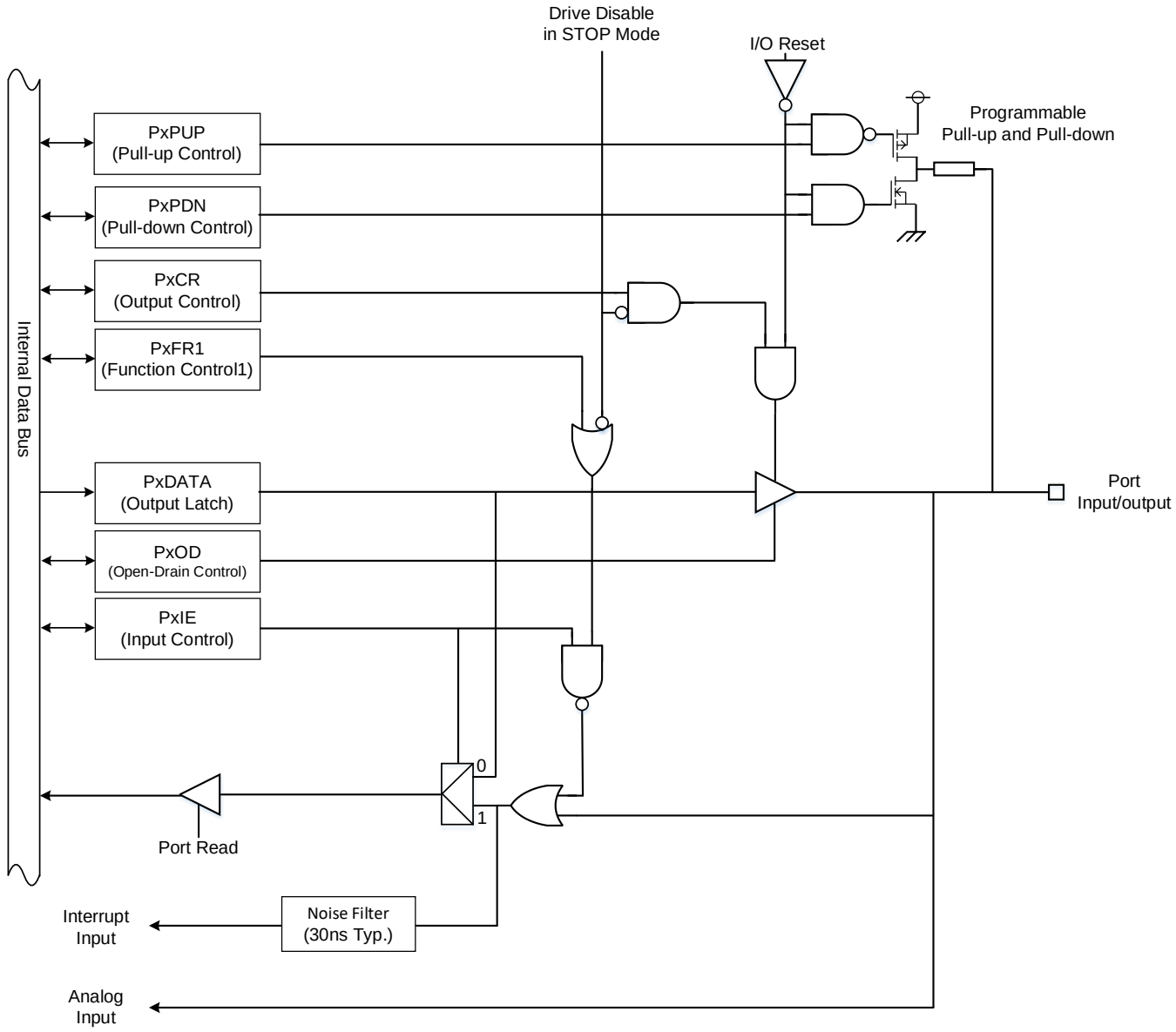


Figure 8.17 Port Type T17

8.4.18. Type T18

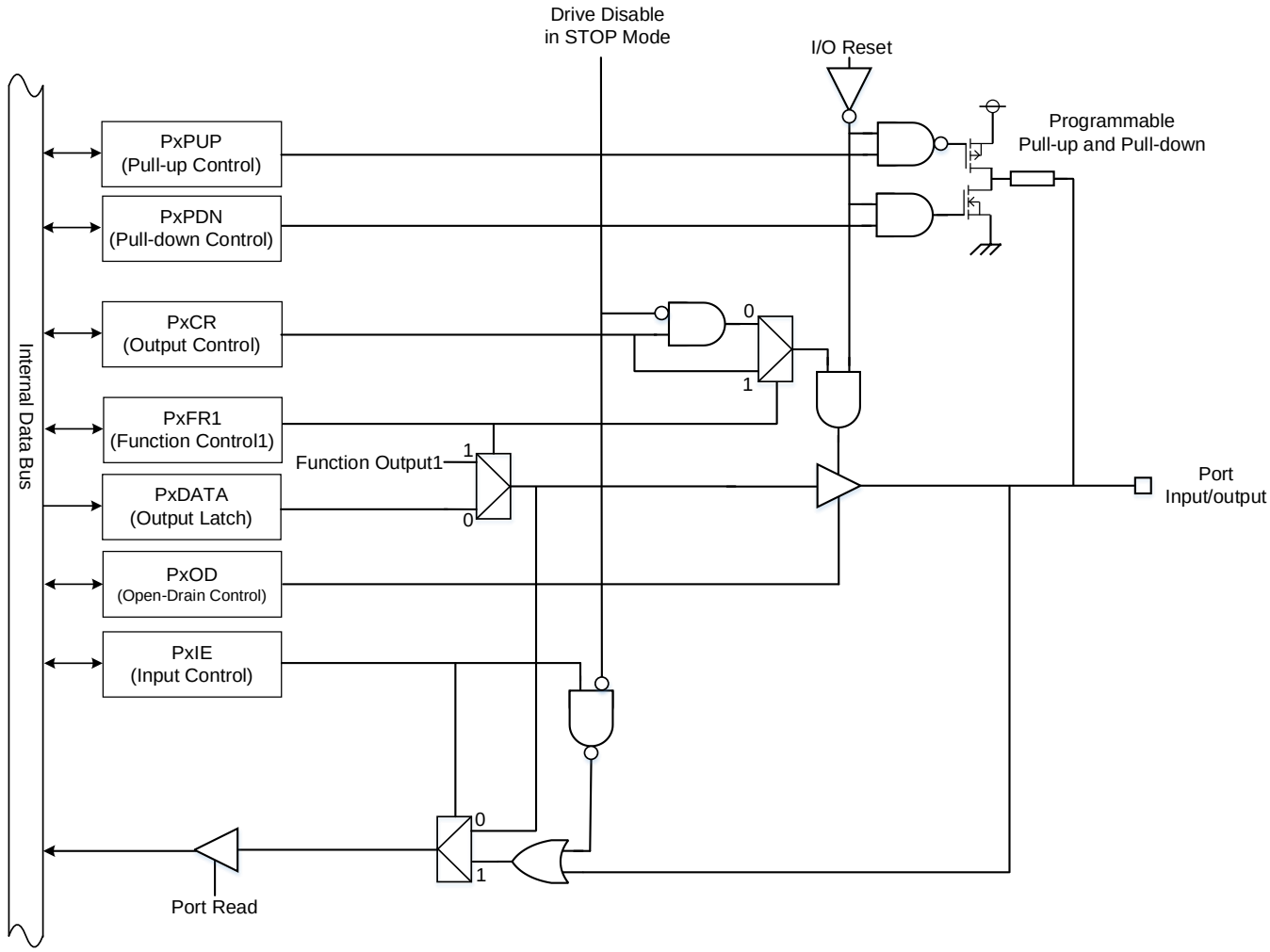


Figure 8.18 Port Type T18

8.4.19. Type T19

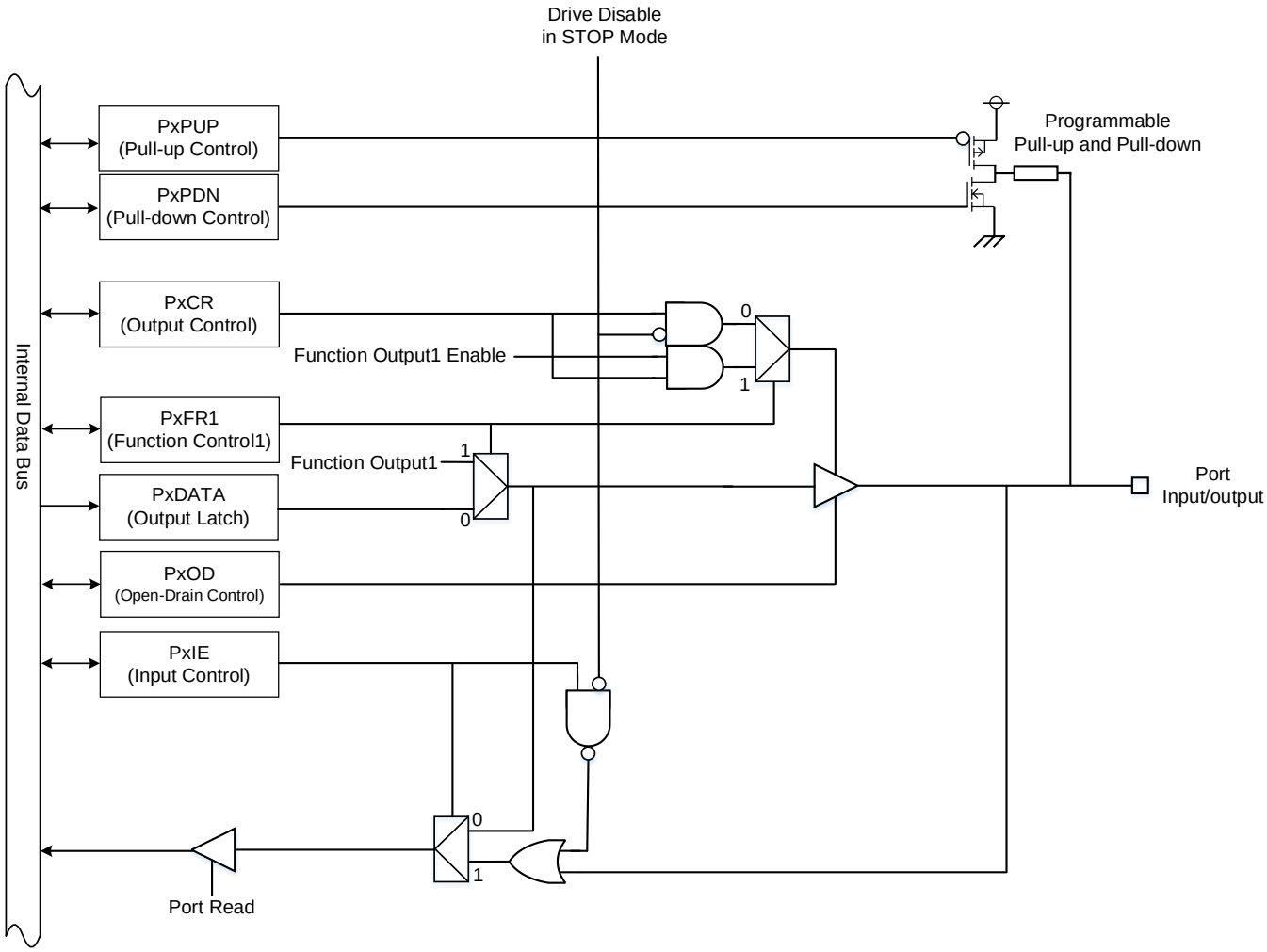


Figure 8.19 Port Type T19

8.4.20. Type T20

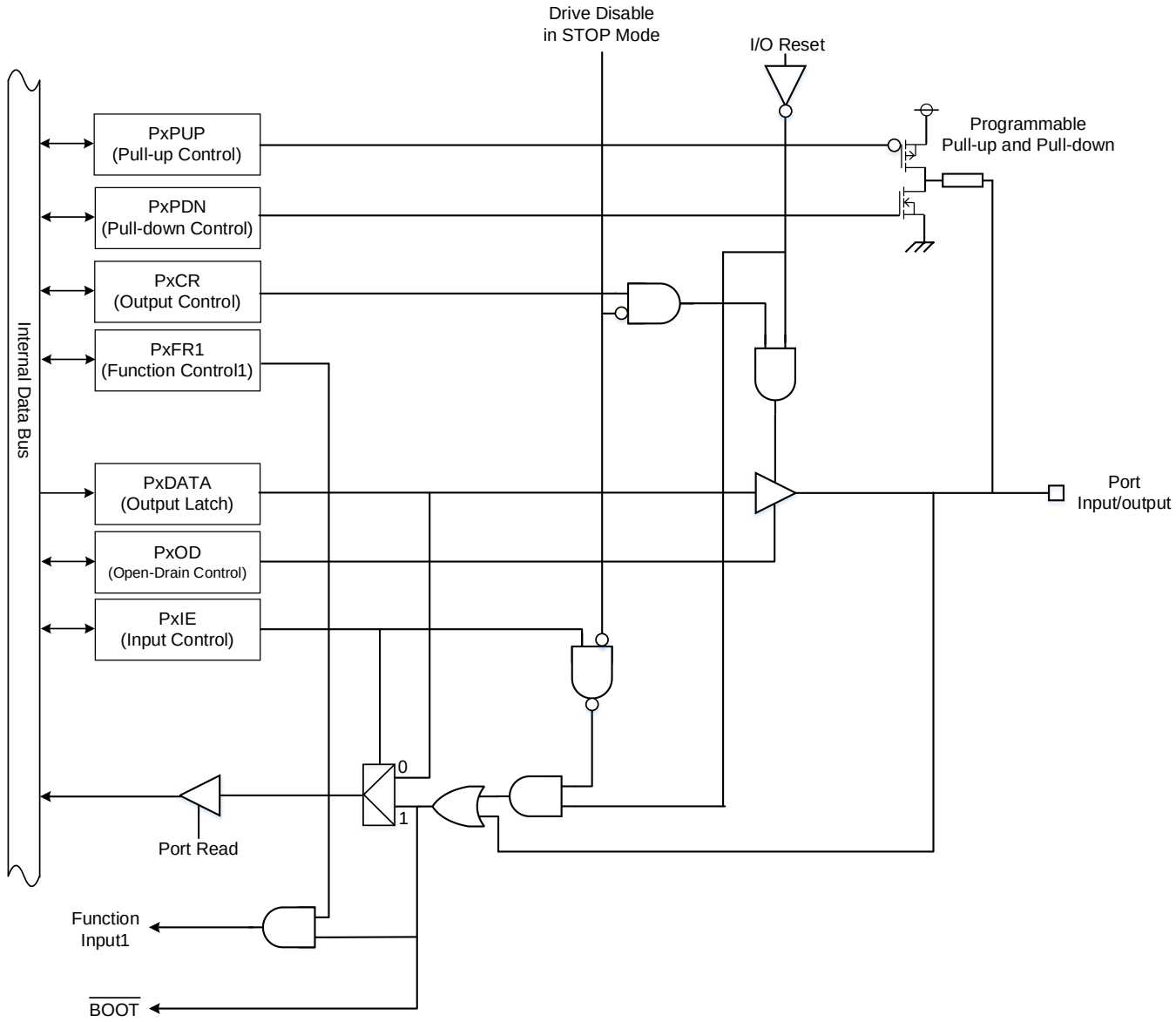


Figure 8.20 Port Type T20

8.4.21. Type T21

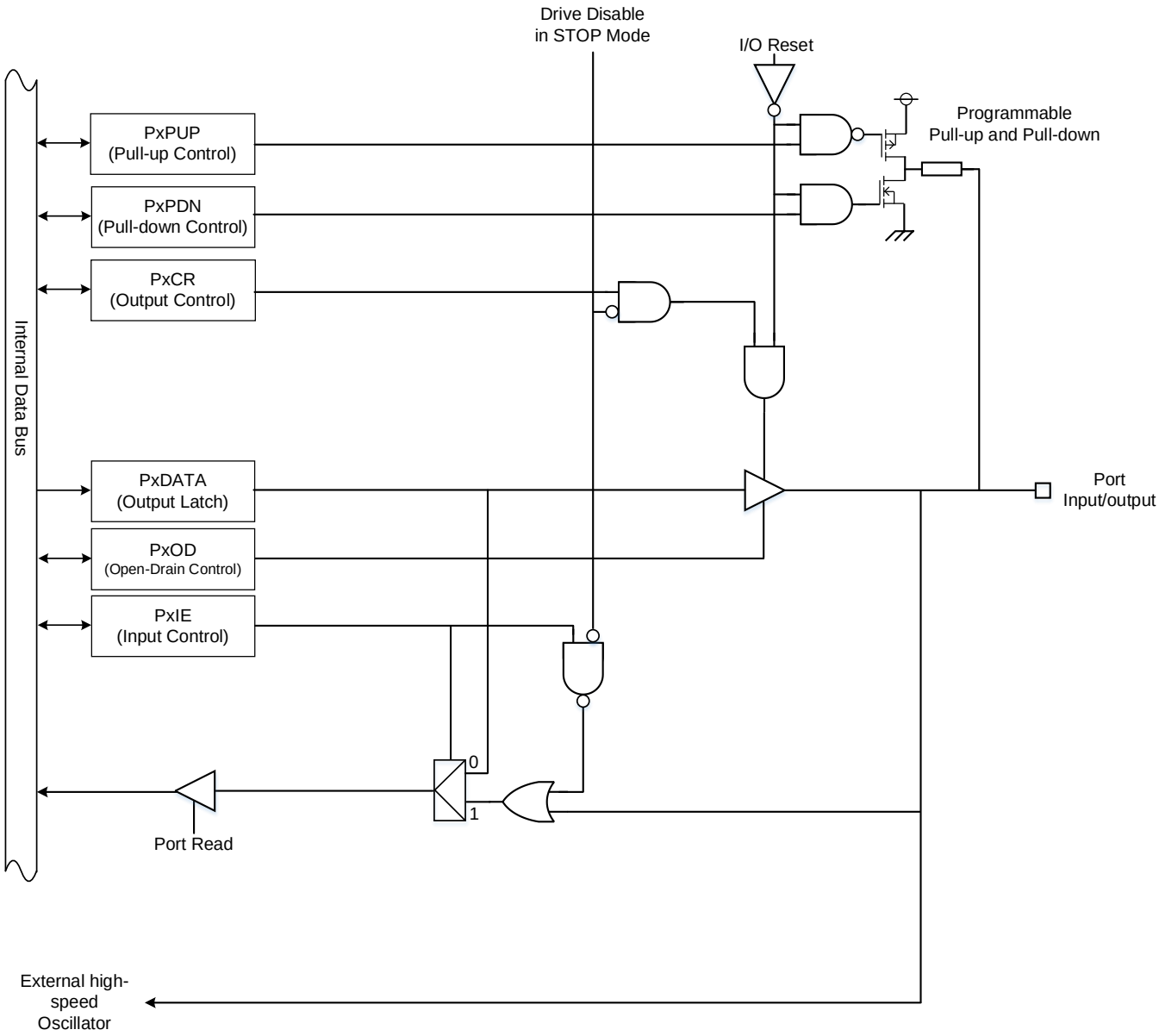


Figure 8.21 Port Type T21

8.5. Precaution

8.5.1. Pin status during a reset

During the reset, the pin status is high impedance except for below pins. And, the pull-up/pull-down is invalid.

- The debug interface pins (PB3 to PB7) are debug pin.
- PF0 ($\overline{\text{BOOT}}$) is used as a BOOT function. It is enabled to be input and pulled-up during pin reset. At the rising edge of the reset signal, when PF0 is "High" level, the MCU enters single chip mode and boots from the built-in flash memory. When PF0 is "Low" level, the MCU enters single BOOT mode and boots from the built-in BOOT ROM.

8.5.2. Unused pins

We recommend that each unused pin should be connected to the power supply pins or GND pins via resistors. Generally, when MCU operates while the high-impedance pins left open, electrostatic damage or latch-up may occur in the internal LSI due to induced voltage influenced from external noise.

8.5.3. Notice of using debug interface pins used as general-purpose ports

After releasing reset, when the debug interface pins are used as the general-purpose ports by the user program, the debug tool cannot be connected to MCU and cannot be controlled.
When the debug tool cannot be used for debugging, erase the flash memory by using UART connection and set single BOOT mode to connect it to MCU again. For details, refer to "20. Flash Operations".

9. 16-bit Timer/Event Counter (TMRB)

9.1. Outline

TMRB has the following features:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square wave output (PPG) mode
- External trigger programmable square wave output (PPG) mode

By using the capture function, TMRB can be used for the following purposes.

- One-shot pulse output from an external trigger
- Pulse width measurement

In the description below, "x" means the channel number.

9.2. Block Diagram

TMRB consists of a 16-bit up counter, two 16-bit timer registers (double buffer configuration), two 16-bit capture registers, two comparators, and a capture input control, a timer flip-flop and a timer flip-flop control circuit. Timer operation mode and timer flip-flop are controlled by registers.

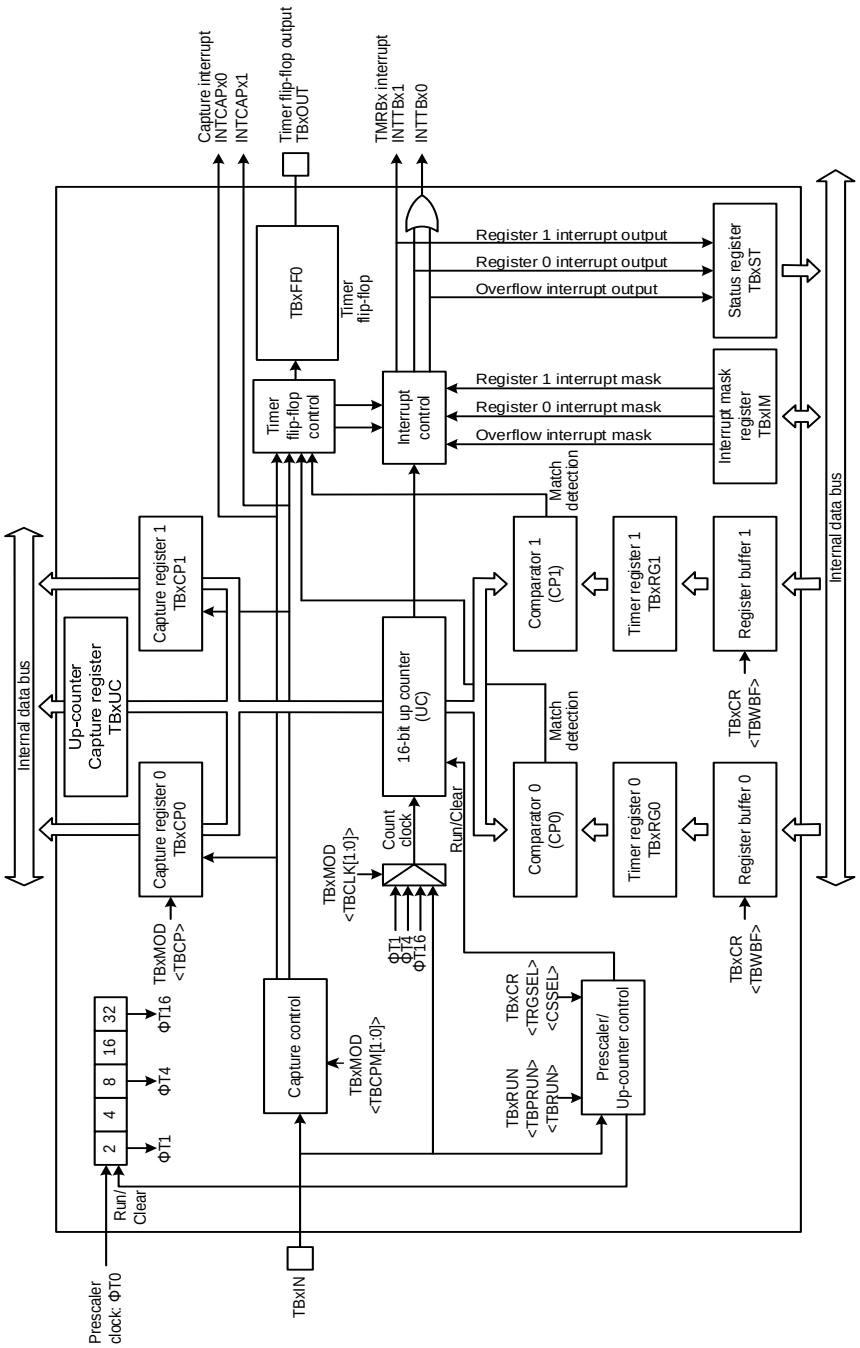


Figure 9.1 TMRB Block Diagram

9.3. Registers

9.3.1. List of Registers

The control registers and addresses are listed below.

Register name		Base+ (Address)
Enable Register	TBxEN	0x0000
RUN Register	TBxRUN	0x0004
Control Register	TBxCR	0x0008
Mode Register	TBxMOD	0x000C
Flip-flop Control Register	TBxFFCR	0x0010
Status Register	TBxST	0x0014
Interrupt Mask Register	TBxIM	0x0018
Up Counter Capture Register	TBxUC	0x001C
Timer Register 0	TBxRG0	0x0020
Timer Register 1	TBxRG1	0x0024
Capture Register 0	TBxCP0	0x0028
Capture Register 1	TBxCP1	0x002C

9.3.2. TBxEN (Enable Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBEN	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	TBEN	R/W	TMRB operation control 0: Disabled 1: Enabled Specifies the operation of TMRB. The clock is not supplied to other registers of TMRB in the operation disabled state, so power consumption can be reduced. (In this state, read and write to registers except TBxEN cannot be performed.) When using TMRB, set <TBEN> to "1" before setting the registers of TMRB. When operation is disabled after TMRB is operated, the settings of the registers are retained.
6:0	-	R	Read as "0".

9.3.3. TBxRUN (RUN Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	TBPRUN	-	TBRUN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:3	-	R	Read as "0".
2	TBPRUN	R/W	Prescaler operation control 0: Stop and clear 1: Start counting
1	-	R	Read as "0".
0	TBRUN	R/W	Up counter operation control 0: Stop and clear 1: Start counting

Note1: When starting counting by an external trigger, be sure to set TBxRUN<TBRUN> to "1".

Note2: If TBxUC<TBxUC[15:0]> is read while the up counter stops(TBxRUN<TBRUN> is set to "0"), the last captured value in operation of the up counter is read.

9.3.4. TBxCR (Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBWBF	-	-	-	I2TB	-	TRGSEL	CSSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	TBWBF	R/W	Double buffer control 0: Disabled 1: Enabled
6:5	-	R/W	Write as "0".
4	-	R	Read as "0".
3	I2TB	R/W	Operation in IDLE mode control 0: Stop 1: Operation
2	-	R	Read as "0".
1	TRGSEL	R/W	External trigger select 0: Rising edge 1: Falling edge
0	CSSEL	R/W	Selecting the start method of the up counter 0: Software start 1: External trigger start

Note1: Do not modify TBxCR while TMRB is operating.

Note2: When the external trigger start is selected as the method to start up counter, after set TBxCR<CSSEL> and TBxCR<TRGSEL>, set TBxRUN<TBRUN> and TBxRUN<TBPRUN> to "1".

9.3.5. TBxMOD (Mode Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	TBRSWR	TBCP	TBCPM		TBCLE	TBCLK	
After reset	0	0	1	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:7	-	R	Read as "0".
6	TBRSWR	R/W	Write timing control to timer registers 0 and 1 when double buffer is used 0: Even if only one side is ready to write to timer register 0 and timer register 1, one side at a time can write to the timer register. 1: If neither timer register 0 nor timer register 1 is ready to be written, the timer register cannot be written.
5	TBCP	R	Read as "1".
		W	Software capture control 0: Software Capture 1: Don't care Writing "0" to this bit captures the count value into the capture register 0 (TBxCP0).
4:3	TBCPM[1:0]	R/W	Capture control 00: Disabled 01: Capture register 0 (TBxCP0) captures the count value at the rising edge of TBxIN pin. 10: Capture register 0 (TBxCP0) captures the count value at the rising edge of TBxIN pin input, and capture register 1 (TBxCP1) captures the count value at the falling edge of TBxIN pin input. 11: Disabled
2	TBCLE	R/W	Up counter clear control 0: Clear disabled 1: Clear enabled Controls clear control of the up counter. When this bit is set to "0", clearing is disabled. When this bit is set to "1", clearing is performed when up counter matches the timer register 1 (TBxRG1).
1:0	TBCLK[1:0]	R/W	TMRB source clock select 00: TBxIN pin input 01: ΦT1 10: ΦT4 11: ΦT16

Note: Do not modify TBxMOD while the TMRB is operating.

9.3.6. TBxFFCR (Flip-Flop Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	TBC1T1	TBC0T1	TBE1T1	TBE0T1	TBFF0C	
After reset	1	1	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:6		R	Read as "1".
5	TBC1T1	R/W	TBxFF0 inversion control when the up counter value is captured to TBxCP1 0: Invert disabled 1: Invert enabled When this bit is set to "1", TBxFF0 is inverted when the up counter value is captured to TBxCP1. When "0" is set, it is not inverted.
4	TBC0T1	R/W	TBxFF0 inversion control when the up counter value is captured to TBxCP0 0: Invert disabled 1: Invert enabled When this bit is set to "1", TBxFF0 is inverted when the up counter value is captured to TBxCP0. When "0" is set, it is not inverted.
3	TBE1T1	R/W	TBxFF0 inversion control when up counter matches TBxRG1 0: Invert disabled 1: Invert enabled When this bit is set to "1", TBxFF0 is inverted when the up counter matches TBxRG1. When "0" is set, it is not inverted.
2	TBE0T1	R/W	TBxFF0 inversion control when up counter matches TBxRG0 0: Invert disabled 1: Invert enabled When this bit is set to "1", TBxFF0 is inverted when the up counter matches TBxRG0. When "0" is set, it is not inverted.
1:0	TBFF0C[1:0]	R	Read as "11".
		W	TBxFF0 control 00: Invert TBxFF0. 01: Set TBxFF0 to "1". 10: Clear TBxFF0 to "0". 11: Don't care

Note: Do not change TBxFFCR while TMRB is operating.

9.3.7. TBxST (Status Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	INTTBOF	INTTB1	INTTB0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:3	-	R	Read as "0".
2	INTTBOF	R	Overflow flag 0: No overflow occurs. 1: Overflow occurs. When the up counter overflows, this flag is set to "1".
1	INTTB1	R	Flag matching up counter and TBxRG1 0: No match is detected 1: Matching with TBxRG1 is detected. When the up counter matches TBxRG1, this flag is set to "1".
0	INTTB0	R	Flag matching up counter and TBxRG0 0: No match is detected 1: Matching with TBxRG0 was detected. When the up counter matches TBxRG0, this flag is set to "1".

Note1: Even if the interrupt request is masked by TBxIM, the flag is set when an interrupt request occurs.

Note2: The flags are automatically cleared when the register is read.

9.3.8. TBxIM (Interrupt Mask Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	TBIMOF	TBIM1	TBIM0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:3	-	R	Read as "0".
2	TBIMOF	R/W	Overflow interrupt request mask control 0: Do not mask interrupt request 1: Mask the interrupt request Setting this bit to "0" does not mask the overflow interrupt request. Masks when "1" is set.
1	TBIM1	R/W	Matching interrupt request masking control for the up counter and TBxRG1 0: Do not mask interrupt request 1: Mask the interrupt request When this bit is set to "0", the up counter and TBxRG1 matching interrupt request is not masked. Masks when "1" is set.
0	TBIM0	R/W	Matching interrupt request masking control for the up counter and TBxRG0 0: Do not mask interrupt request 1: Mask the interrupt request When this bit is set to "0", the up counter and TBxRG0 matching interrupt request is not masked. Masks when "1" is set.

Note: When an interrupt request is generated even if the interrupt request is masked by TBxIM, the each flag in TBxST is set.

9.3.9. TBxUC (Up Counter Capture Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TBUC							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBUC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TBUC[15:0]	R	Capture value of the up counter When TBxUC<TBxUC[15:0]> is read during up counter operation, the value of the up counter at reading is captured and read.

Note: When the up counter is stopped (TBxRUN<TBRUN>= "0"), TBxUC<TBxUC[15:0]> is read the last value captured during up counter operation.

9.3.10. TBxRG0 (Timer Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TBRG0							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TBRG0[15:0]	R/W	Set the value to be compared with the up counter.

9.3.11. TBxRG1 (Timer Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TBRG1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TBRG1[15:0]	R/W	Set the value to be compared with the up counter.

9.3.12. TBxCP0 (Capture Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TBCP0							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBCP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TBCP0[15:0]	R	The captured value of the up counter can be read.

9.3.13. TBxCP1 (Capture Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TBCP1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBCP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TBCP1[15:0]	R	The captured value of the up counter can be read.

9.4. Operations

9.4.1. Prescaler

The prescaler is a 4-bit prescaler that generates the source clock for the up counter (UC).

The input clock ΦT0 to the prescaler is fperiph / 1, fperiph / 2, fperiph / 4, fperiph / 8, fperiph / 16, or fperiph / 32 selected by CGSYSCR<PRCK[2:0]>.

The fperiph is the clock for either fgear or fc selected by CGSYSCR<FPSEL>.

The fgear is one of the clocks for fc / 1, fc / 2, fc / 8, fc / 16 selected by CGSYSCR<GEAR[2:0]>.

The operation of the prescaler is controlled by TBxRUN<TBPRUN>. Writing "1" starts counting, and writing "0" stops counting and clears the prescaler.

Table 9.1 shows the resolution of the output clock from the prescaler.

Table 9.1 Resolution of Output Clock from Prescaler (fc = 80MHz)

CGSYSCR <FPSEL>	CGSYSCR <GEAR[2:0]>	CGSYSCR <PRCK[2:0]>	Resolution of output clock from prescaler		
			ΦT1	ΦT4	ΦT16
0 (fgear)	000 (fc)	000 (fperiph/1)	0.025μs (fc/2^1)	0.1μs (fc/2^3)	0.4μs (fc/2^5)
		001 (fperiph/2)	0.05μs (fc/2^2)	0.2μs (fc/2^4)	0.8μs (fc/2^6)
		010 (fperiph/4)	0.1μs (fc/2^3)	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		011 (fperiph/8)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		100 (fperiph/16)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		101 (fperiph/32)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
	100 (fc / 2)	000 (fperiph/1)	0.05μs (fc/2^2)	0.2μs (fc/2^4)	0.8μs (fc/2^6)
		001 (fperiph/2)	0.1μs (fc/2^3)	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		010 (fperiph/4)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		011 (fperiph/8)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		100 (fperiph/16)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
		101 (fperiph/32)	1.6μs (fc/2^7)	6.4μs (fc/2^9)	25.6μs (fc/2^{11})
	101 (fc / 4)	000 (fperiph/1)	0.1μs (fc/2^3)	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		001 (fperiph/2)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		010 (fperiph/4)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		011 (fperiph/8)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
		100 (fperiph/16)	1.6μs (fc/2^7)	6.4μs (fc/2^9)	25.6μs (fc/2^{11})
		101 (fperiph/32)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})	51.2μs (fc/2^{12})
	110 (fc / 8)	000 (fperiph/1)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		001 (fperiph/2)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		010 (fperiph/4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
		011 (fperiph/8)	1.6μs (fc/2^7)	6.4μs (fc/2^9)	25.6μs (fc/2^{11})
		100 (fperiph/16)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})	51.2μs (fc/2^{12})
		101 (fperiph/32)	6.4μs (fc/2^9)	25.6μs (fc/2^{11})	102.4μs (fc/2^{13})
	111 (fc /16)	000 (fperiph/1)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		001 (fperiph/2)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
		010 (fperiph/4)	1.6μs (fc/2^7)	6.4μs (fc/2^9)	25.6μs (fc/2^{11})
		011 (fperiph/8)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})	51.2μs (fc/2^{12})
		100 (fperiph/16)	6.4μs (fc/2^9)	25.6μs (fc/2^{11})	102.4μs (fc/2^{13})
		101 (fperiph/32)	12.8μs (fc/2^{10})	51.2μs (fc/2^{12})	204.8μs (fc/2^{14})

CGSYSCR <FPSEL>	CGSYSCR <GEAR[2:0]>	CGSYSCR <PRCK[2:0]>	Resolution of output clock from prescaler		
			ΦT1	ΦT4	ΦT16
1 (fc)	000 (fc)	000 (fperiph/1)	0.025μs (fc/2^1)	0.1μs (fc/2^3)	0.4μs (fc/2^5)
		001 (fperiph/2)	0.05μs (fc/2^2)	0.2μs (fc/2^4)	0.8μs (fc/2^6)
		010 (fperiph/4)	0.1μs (fc/2^3)	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		011 (fperiph/8)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		100 (fperiph/16)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		101 (fperiph/32)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
	100 (fc / 2)	000 (fperiph/1)	-	0.1μs (fc/2^3)	0.4μs (fc/2^5)
		001 (fperiph/2)	0.05μs (fc/2^2)	0.2μs (fc/2^4)	0.8μs (fc/2^6)
		010 (fperiph/4)	0.1μs (fc/2^3)	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		011 (fperiph/8)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		100 (fperiph/16)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		101 (fperiph/32)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
	101 (fc / 4)	000 (fperiph/1)	-	0.1μs (fc/2^3)	0.4μs (fc/2^5)
		001 (fperiph/2)	-	0.2μs (fc/2^4)	0.8μs (fc/2^6)
		010 (fperiph/4)	0.1μs (fc/2^3)	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		011 (fperiph/8)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		100 (fperiph/16)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		101 (fperiph/32)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
	110 (fc / 8)	000 (fperiph/1)	-	-	0.4μs (fc/2^5)
		001 (fperiph/2)	-	0.2μs (fc/2^4)	0.8μs (fc/2^6)
		010 (fperiph/4)	-	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		011 (fperiph/8)	0.2μs (fc/2^4)	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		100 (fperiph/16)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		101 (fperiph/32)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})
	111 (fc /16)	000 (fperiph/1)	-	-	0.4μs (fc/2^5)
		001 (fperiph/2)	-	-	0.8μs (fc/2^6)
		010 (fperiph/4)	-	0.4μs (fc/2^5)	1.6μs (fc/2^7)
		011 (fperiph/8)	-	0.8μs (fc/2^6)	3.2μs (fc/2^8)
		100 (fperiph/16)	0.4μs (fc/2^5)	1.6μs (fc/2^7)	6.4μs (fc/2^9)
		101 (fperiph/32)	0.8μs (fc/2^6)	3.2μs (fc/2^8)	12.8μs (fc/2^{10})

Note1: Be sure to select the output clock ΦTn from the prescaler so that ΦTn < fsys is satisfied (ΦTn becomes slower than fsys).

Note2: Do not switch the clock gear while TMRB is running.

Note3: "-" in the cell means setting prohibited.

9.4.2. Up Counter (UC)

UC is 16-bit binary counter.

- Source clock

  The source clock is set by TBxMOD<TBCLK[1:0]>. The prescaler output clock ΦT1, ΦT4, ΦT16, or TBxIN inputs can be selected.

- Starting and stopping UC operation

  Clearing UC is set by TBxRUN<TBRUN>. When<TBRUN> is set to "1", counting starts. And when it is set to "0", counting stops and UC is cleared.

- UC clear timing

  (a) When compare match

  By setting TBxMOD<TBCLE> to "1", UC is cleared with matching of UC and TBxRG1. When TBxMOD<TBCLE> is set to "0", no match between UC and TBxRG1 is detected. Therefore, UC operates as a free-running up counter.

  (b) When UC is stopped

  By setting TBxRUN<TBRUN> to "0", counting stops and UC is cleared.

- UC overflowing

  When UC overflows, TBxST<INTTBOF> is set to "1" and an overflow interrupt (INTTBx0) is generated.

9.4.3. Timer Register (TBxRG0, TBxRG1)

There are two registers which set the value to be compared with UC. The value set in the timer register is compared with the value of UC by the comparator (CPn), and when it matches, CPn outputs a match detection signal.

TBxRG0 and TBxRG1 are double buffer configuration and paired with buffer registers. The double buffer is disabled in the initial state.

Double buffer control is set by TBxCR<TBWBF>. When TBxCR<TBWBF> is set to "0", double buffer is disabled, and when TBxCR<TBWBF> is set to "1", it is enabled.

When double buffer is enabled, data is transferred from register buffer 0/1 to the timer register TBxRG0/1, respectively, when UC matches TBxRG1. When UC is stopped even if the double buffer is enabled, the double buffer is disabled and the value transferred to TBxRG0/1 as soon as the value is written to the buffer register.

9.4.4. Capture Control

This circuit controls the timing at which UC value is captured into the capture register TBxCP0, TBxCP1 according to rising or falling edge of TBxIN pin input. The timing for capturing to the capture register is set by TBxMOD<TBCPM[1:0]>.

UC value can also be captured to the capture register at any timing by software. When TBxMOD<TBCP> is set to "0", the value of UC at the set timing is captured into TBxCP0.

9.4.5. Capture Register (TBxCP0, TBxCP1)

These registers store the captured value of UC. There are two registers. For details about which capture register stores the captured value, refer to "9.4.4. Capture Control".

To read TBxCP0 or TBxCP1, use a 16-bit data transfer instruction or read in the order of lower and higher.

9.4.6. Up Counter Capture Register (TBxUC)

In addition to capture by capture control, the present value of UC can be captured and read by reading TBxUC.

9.4.7. Comparator (CP0, CP1)

CP0 and CP1 compare UC with the value of TBxRG0 and TBxRG1. When they match, a match signal is output and INTTBx0 and INTTBx1 are generated.

9.4.8. Timer Flip-Flop (TBxFF0)

A timer flip-flop (TBxFF0) is a flip-flop whose output is inverted by a match signal from CPn or a signal which captures the value of UC to TBxCPn. Enable/disable inversion is set by TBxFFCR<TBC1T1>, <TBC0T1>, <TBE1T1>, <TBE0T1>.

After releasing reset, TBxFF0 is undefined.

TBxFF0 inverts when TBxFFCR<TBFF0C[1:0]> is set to "00", and is set to "1" and "0" when TBxFFCR <TBFF0C[1:0]> is set to "01" and "10", respectively.

The value of TBxFF0 is output to the timer flip-flop output pin (TBxOUT). When it is output to TBxOUT, it is necessary to set the corresponding port beforehand.

9.4.9. Capture Interrupt (INTCAPx0, INTCAPx1)

INTCAPx0 and INTCAPx1 occur when the value of UC is captured into the capture register TBxCP0 and TBxCP1.

9.5. Explanation of Each Mode

9.5.1. 16-bit Interval Timer Mode

To generate an interrupt with a constant cycle, set an interval time in TBxRG0 and TBxRG1 to generate a INTTBx0 and INTTBx1, respectively.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	x	x	x	x	x	x	x	Enables operation of TMRB.
TBxRUN	←	x	x	x	x	x	0	x	0	Stops and clears the operation of the prescaler and UC.
Interrupt enable set Register	←	*	*	*	*	*	*	*	*	Set the bit corresponding to INTTBx1 interrupt to "1", enables interrupts.
TBxFFCR	←	x	x	0	0	0	0	1	1	Disables TBxFF0 invert.
TBxMOD	←	x	0	1	0	0	1	Y	Y	Prescaler output is used as source clock of UC.
					YY = "01", "10" or "11"					Does not capture by software and hardware.
TBxRG1	←	*	*	*	*	*	*	*	*	Sets the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TBxRUN	←	x	x	x	x	x	1	x	1	Starts operation of the prescaler and counting of the UC.

Note: x: Don't care, *: Can be set to any value, -: Cannot be changed

9.5.2. 16-bit Event Counter Mode

Use TMRB as an event counter by setting the source clock of UC as the signal to be input to TBxIN pin.

UC counts up on the rising edge of the signal.

The value of UC can be also captured to TBxCP0 by setting TBxMOD<TBCP> to "0".

		7	6	5	4	3	2	1	0	
TBxEN	←	1	x	x	x	x	x	x	x	Enables operation of TMRB.
TBxRUN	←	x	x	x	x	x	0	x	0	Stops and clears the operation of the prescaler and UC.
Port register setting										Make settings for using the ports assigned to TBxIN pins as TBxIN pins.
TBxFFCR	←	x	x	0	0	0	0	1	1	Disables TBxFF0 invert.
TBxMOD	←	x	0	1	0	0	0	0	0	TBxIN pin input signal is used as the source clock of UC. Does not capture by software and hardware.
TBxRUN	←	x	x	x	x	x	1	x	1	Starts operation of the prescaler and counting of the UC.
	:									
	:									
TBxMOD	←	x	0	0	0	0	0	0	0	The value of UC is captured to TBxCP0.

Note: x: Don't care, *: Can be set to any value, -: Cannot be changed

9.5.3. 16-bit Programmable Square Wave Output (PPG) Mode

A square wave (programmable square wave) with an arbitrary duty cycle is output.

The programmable square wave supports either "Low" active or "High" active.

TBxRG1 determines the cycle of the programmable square wave. TBxRG0 determines the duty of the programmable square wave. Enable TBxFF0 invert at matching between UC and TBxRG0, TBxRG1. Set each parameter so that TBxRG1 is greater than TBxRG0.

To output a programmable square wave from TBxOUT pin, setting to use the port as an TBxOUT pin must be made beforehand.

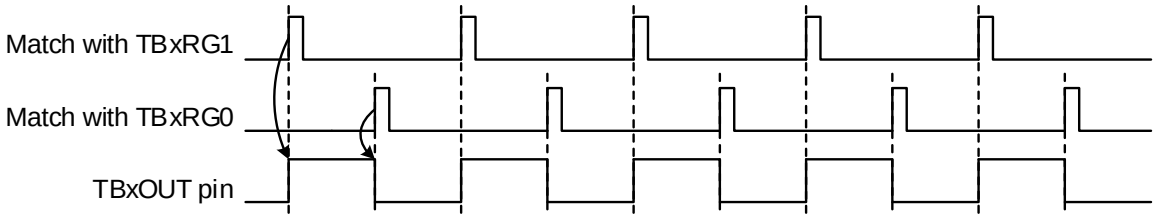


Figure 9.2 Output Waveforms in 16-bit Programmable Square Wave Output (PPG) Mode

Changing TBxRG0 may not be in time when reducing the programmable square wave duty.

By enabling double buffer, pre-configured register buffer 0 value is transferred to TBxRG0 for UC and TBxRG1 matching.

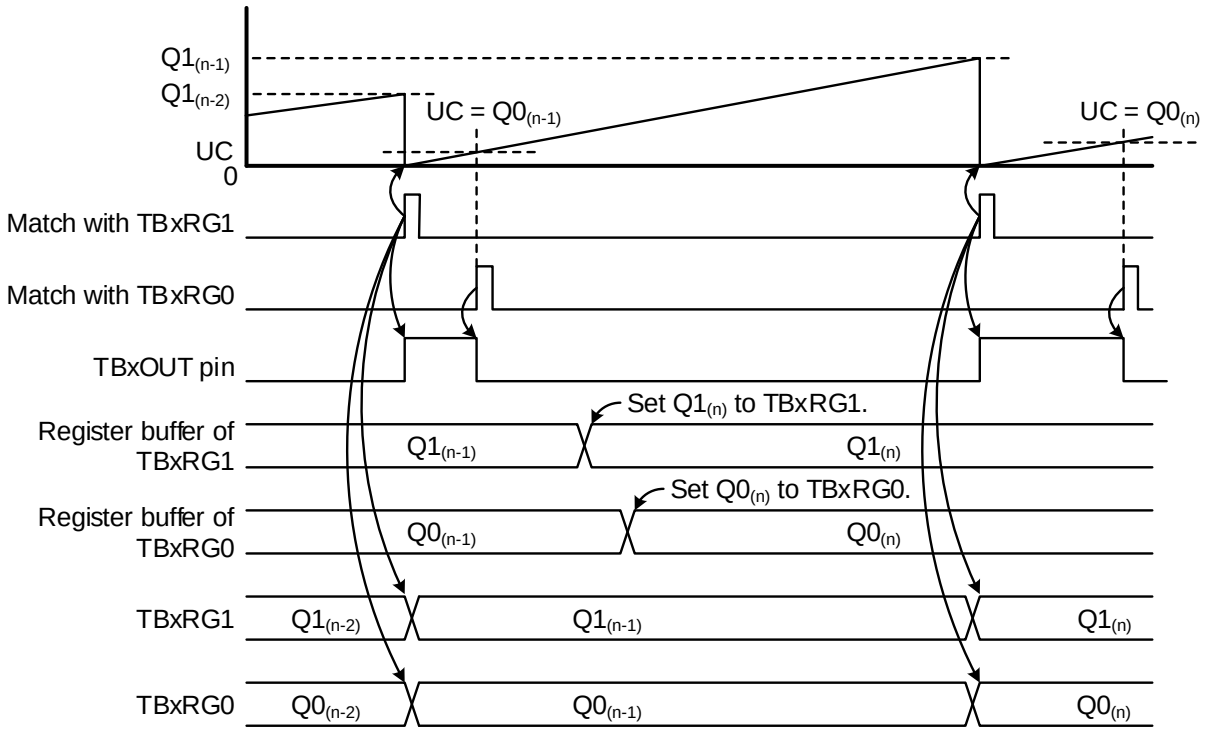


Figure 9.3 Register Buffer Operation

Figure 9.4 shows the output circuit in 16-bit programmable square wave output (PPG) mode.

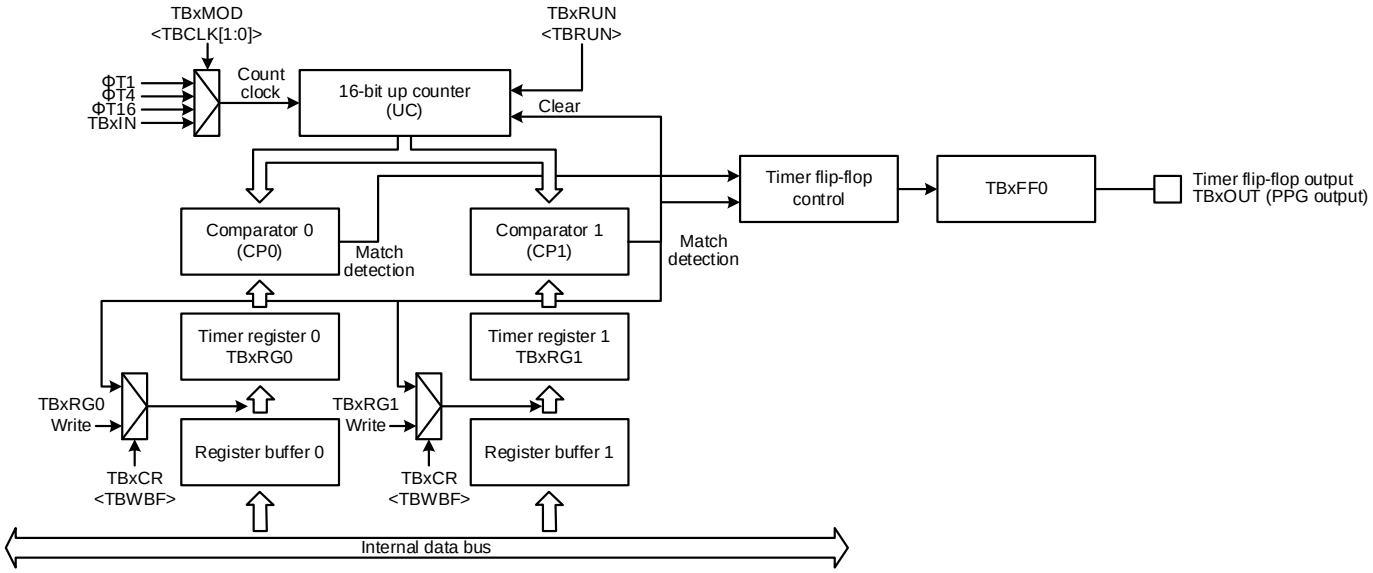


Figure 9.4 Output Circuit in 16-bit Programmable Square Wave Output (PPG) Mode

In 16-bit programmable square wave output (PPG) mode, each register is set as follows.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	x	x	x	x	x	x	x	Enables operation of TMRB.
TBxRUN	←	x	x	x	x	x	0	x	0	Stops and clears the operation of the prescaler and UC.
TBxCR	←	0	0	0	x	x	x	x	0	Disables the double buffer. Set counter start method to software start.
TBxRG0	←	*	*	*	*	*	*	*	*	Sets the duty (16 bits).
		*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Sets the cycle (16 bits)
		*	*	*	*	*	*	*	*	
TBxCR	←	1	0	0	x	x	x	x	-	Enables double Buffer.
TBxFFCR	←	x	x	0	0	1	1	1	0	Sets TBxFF0 to "0". Enables TBxFF0 invert by matching with UC and TBxRG0, TBxRG1.
TBxMOD	←	x	0	1	0	0	1	Y	Y	Enables UC clearing when UC matches TBxRG1.
				YY = "01", "10" or "11"						Disables capture. Makes UC source clock to prescaler output clock.
Port register setting										Sets the port assigned to TBxOUT pin for use as TBxOUT pin.
TBxRUN	←	x	x	x	x	x	1	x	1	Starts operation of the prescaler and counting of the UC.

Note: x: Don't care, *: Can be set to any value.-: Cannot be changed

9.5.4. External Trigger Programmable Square Wave (PPG) Output Mode

A programmable square wave with a delay for the external signal input to TBxIN pin is output.

How to output a "High" pulse-width (p) programmable square wave with a delay of (d) once from the rising edge of an external signal is explained below.

Set TBxCR<CSSEL> to "1" to start UC with an external trigger. Also, set TBxCR<TRGSEL> to "0" to set the rising edge of the external signal as an external trigger.

Set a delay (d) in TBxRG0. Set TBxRG1 to (d) + (p) that is added the delay (d) and pulse width (p).

Set TBxFFCR<TBFF0C[1:0]> to "10" to set TBxFF0 to "0". Set TBxFFCR<TBE1T1> and <TBE0T1> to "1" to enable TBxFF0 invert at matching UC and TBxRG0,TBxRG1.

UC starts at the rising edge of the external signal. TBxOUT pin changes to "High" level when UC matches TBxRG0. When UC and TBxRG1 are matched, TBxOUT pin changes to "Low" level and INTTBx1 occurs. In INTTBx1 interrupt service routine, set TBxFFCR<TBE1T1> and <TBE0T1> to "1" to disable TBxFF0 invert, or set TBxRUN<TBRUN> to "0" to stop and clear UC.

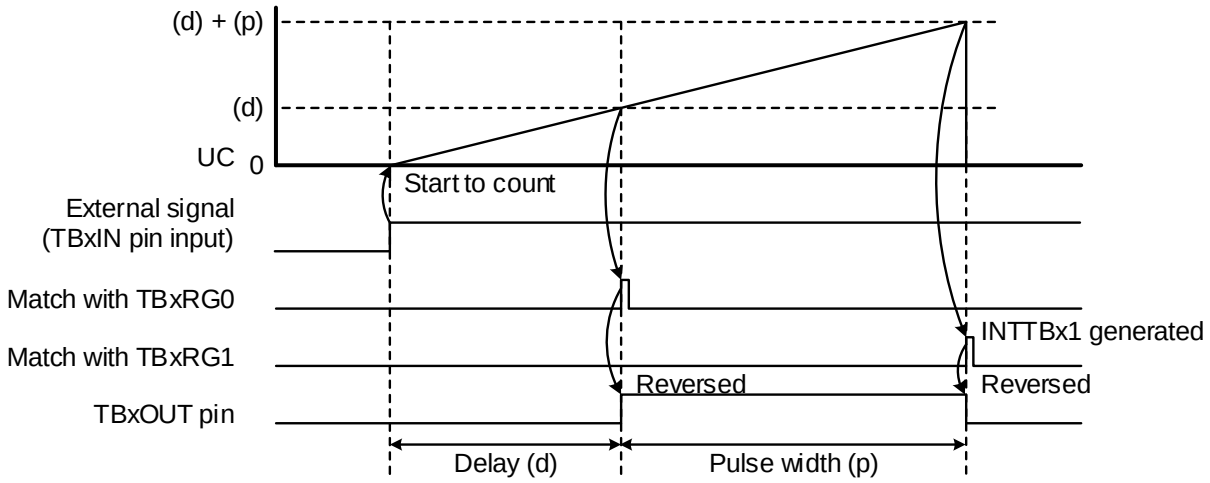


Figure 9.5 Square Wave Output in External Trigger Programmable Square Wave (PPG) Output Mode

9.5.5. Applications Using the Capture Function

By using the capture function, many applications are possible, including the following examples.

(5) Square wave output from external signal

(6) Pulse width measurement

9.5.5.1. Square Wave Output from External Signal

How to output a "High" pulse-width (p) programmable square wave with a delay of (d) once from the rising edge of an external signal is explained below when UC is operated as a free-running up comer by using the capture function.

Set TBxMOD<TBCLE> to "0" to disable clearing of UC. UC operates as a free-running up counter.
Set TBxMOD<TBCPM[1:0]> to "01" to capture the value of UC to TBxCAP0 at the rising edge of TBxIN pin input.
Set TBxFFCR<TBFF0C[1:0]> to "10" to set TBxFF0 to "0". Set TBxFFCR<TBE1T1> and <TBE0T1> to "1" to enable TBxFF0 invert at matching UC and TBxRG0,TBxRG1.
When the external signal rises, UC value is captured to TBxCAP0 and INTCAPx0 occurs. In INTCAPx0 interrupt service routine, read the value (c) of TBxCAP0 and set TBxRG0 to the value (c) + (d) obtained by adding the delay (d). Also, set TBxRG1 to the value (c) + (d) + (p) obtained by adding "High" pulse width (p) to (c) + (d).
TBxOUT pin changes to "High" level when UC and TBxRG0 are matched. When UC and TBxRG1 are matched, TBxOUT pin changes to "Low" level, and INTTBx1 occurs. In INTTBx1 interrupt service routine, set TBxFFCR<TBE1T1> and <TBE0T1> to "1" to disable TBxFF0 invert, or set TBxRUN<TBRUN> to "0" to stop and clear UC.

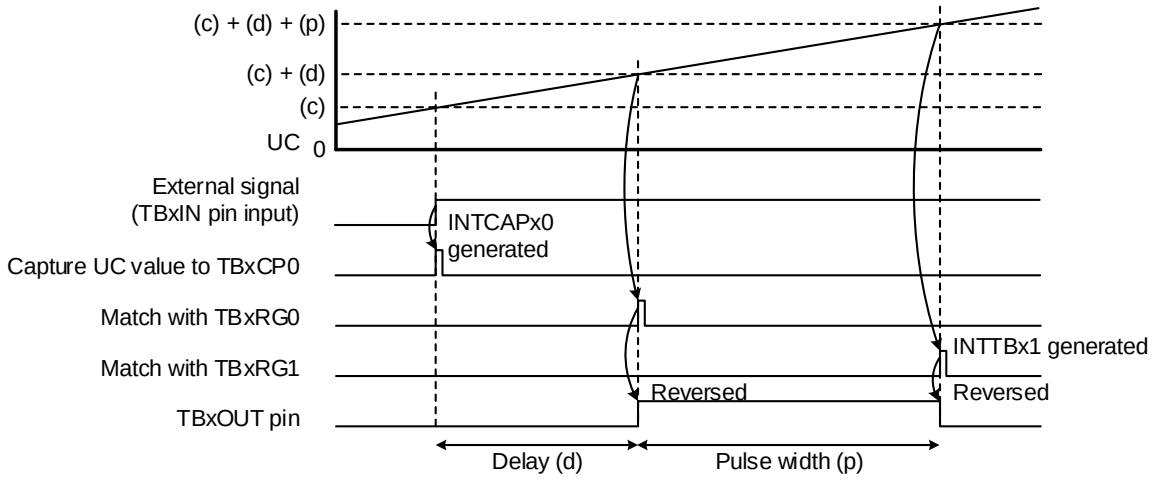


Figure 9.6 Square Wave Output from External Signal Using Free-running Counter (with Delay)

When using the rising edge of TBxIN pin input signal as a trigger, the following shows a setting for outputting a 2ms width square wave after 3ms from trigger.

(d) = 3ms / ΦT1、(p) = 2ms / ΦT1

		7	6	5	4	3	2	1	0	
[Main Processing] Capture by TBxIN pin Setting										
TBxEN	←	1	x	x	x	x	x	x	x	Enables operation of TMRB.
TBxRUN	←	x	x	x	x	x	0	x	0	Stops and clears the operation of the prescaler and UC.
TBxCR	←	0	0	0	x	x	x	x	0	Disables the double buffer. The counter start method is software start.
TBxCR	←	1	0	0	x	x	x	x	-	Enables double buffer.
TBxFFCR	←	x	x	0	0	0	0	1	0	Sets TBxFF0 to "0". Disables TBxFF0 invert.
TBxMOD	←	x	0	1	0	1	0	0	1	Captures the value of UC to TBxCP0 at the rising edge of TBxIN pin. Disables UC clear. Set UC source clock to ΦT1.
Port register setting										Sets the port assigned to TBxIN pin for use as TBxIN pin.
Port register setting										Sets the port assigned to TBxOUT for use as TBxOUT pin.
Interrupt Enable Setting										Enables INTCAPx0.
TBxRUN	←	x	x	x	x	x	1	x	1	Starts counting of the prescaler and UC.
	:									
	:									
[INTCAPx0 Interrupt Service Routine Handling] Delay and Pulse Width Setting										
TBxRG0	←	*	*	*	*	*	*	*	*	Sets TBxCAP0 + (d) as delay.
		*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Sets TBxCAP0 + (d) + (p) as pulse width.
		*	*	*	*	*	*	*	*	
TBxFFCR	←	x	x	0	0	1	1	1	1	Enables TBxFF0 invert at matching UC and TBxRG0, TBxRG1.
Interrupt Enable Setting										Enables INTTBx1.
	:									
	:									
[INTTBx1 Interrupt Service Routine Handling] TBxFF0 invert disabled										
TBxFFCR	←	x	x	0	0	0	0	1	1	Disables TBxFF0 invert.
Interrupt disable setting										Disables INTTBx1.
	:									
	:									

Note: x: Don't care, *: Any number can be set.

9.5.5.2. Measuring "High" Level Width of TBxIN Pin Input Pulses

How to measure the "High" level width of TBxIN pin input pulse by the capture function when UC is operated as a free-running up comer is explained below.

Set TBxMOD<TBCLE> to "0" to disable clearing of UC. UC operates as a free-running up counter.
Set TBxMOD<TBCPM[1:0]> to "10" to capture the value of UC to TBxCAP0 at the rising edge of TBxIN pin input and to TBxCAP1 at the falling edge.
Set the interrupt enable register to generate an interrupt on INTCAPx1.
When the external signal rises, the value of UC is captured to TBxCAP0.
When the external signal falls, the value of UC is captured to TBxCAP1 and INTCAPx1 occurs.
In INTCAPx1 interrupt service routine, subtracting the value of TBxCAP0 from the value of TBxCAP1 and multiplying that value by the cycle of the source clock of UC, the "High" level width of TBxIN pin input pulse can be obtained.
For example, when the difference between TBxCP1 and TBxCP0 is 100 and the cycle of the source clock of UC is 0.5μs, the width of the "High" level is 100 × 0.5μs = 50μs.
When measuring the "High" level width which exceeds the maximum count cycle of UC, software-based processing, such as adding the maximum count cycle of UC, is required.

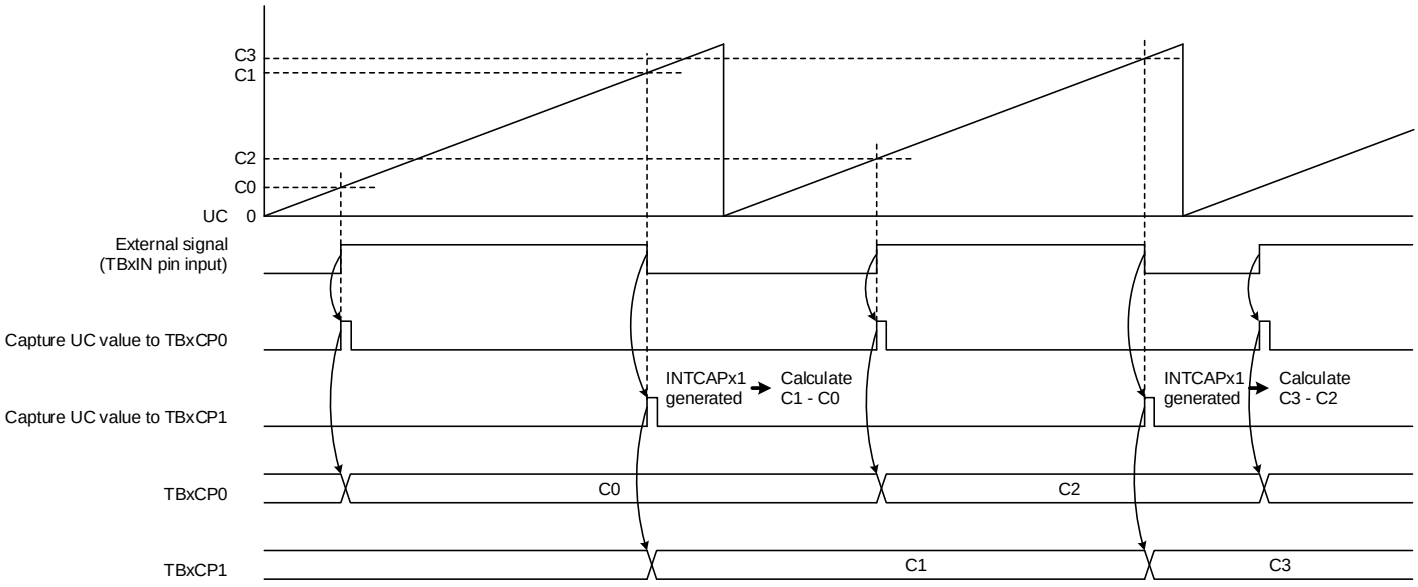


Figure 9.7 Measuring the "High" Level-Width of TBxIN Pin Input Pulses

10. Serial Channel (SIO/UART)

10.1. Outline

The serial channel (SIO/UART) has two modes: synchronous communication mode (SIO mode) and asynchronous communication mode (UART mode).

The following features are available.

- Transfer clock
  - Peripheral clock (ΦT0) can be divided by 2, 8, 32, or 128 with prescaler
  - The prescaler output clock can be divided by 1 to 16.
  - The prescaler output clock can be divided by $N + \frac{(16 - K)}{16}$ (N = 2 to 15, K = 1 to 15) (UART mode only)
  - System clock fsys available (UART mode only)
- Transmission/reception double buffer and FIFO

  Transmission/reception double buffer and up to 4-byte FIFO are available.
- SIO mode
  - Transfer mode: half duplex (transmission/reception), full duplex
  - Data input/output timing:

    SCLKx pin output: input is the rising edge of SCLKx pin, and output is the falling edge of SCLKx pin.

    SCLKx pin input: edge-selectable for SCLKx pin
  - Interval time for continuous transfer can be set.
- UART mode
  - Data length: 7, 8, 9 bits
  - Addition of parity bit (7-or 8-bit data length)
  - Handshake function using $\overline{\text{CTSx}}$ pin
  - Serial link by wake-up function (9-bit data length)

In the following description, "x" means the channel number.

10.2. Block Diagram

Figure 10.1 shows a block diagram of SIO/UART.

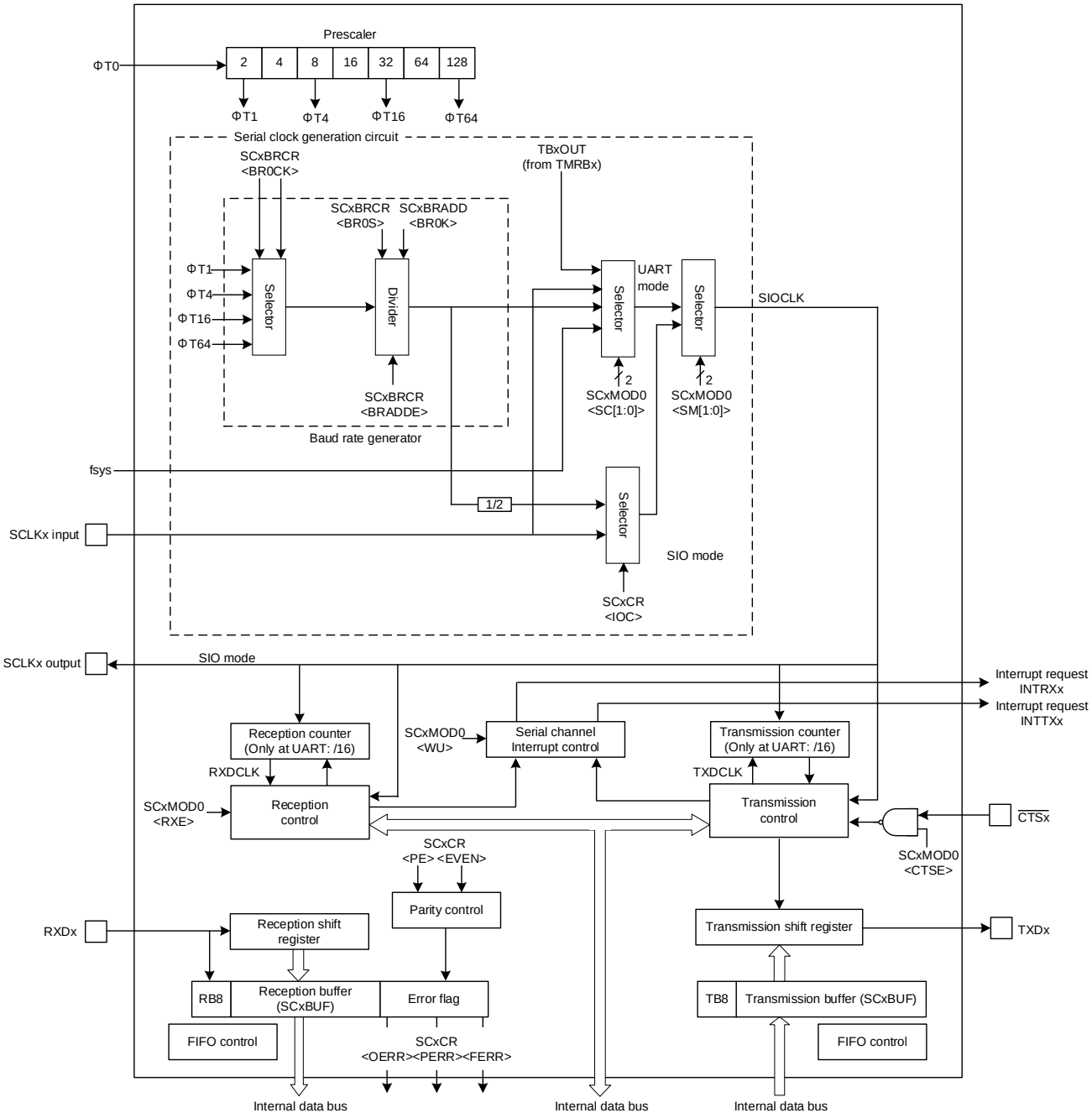


Figure 10.1 SIO/UART Block Diagram

10.3. Registers

10.3.1. Register List

The control registers and addresses are listed below.

Register name		Address (Base+)
Enable Register	SCxEN	0x0000
Transmission/Reception Buffer Register	SCxBUF	0x0004
Control Register	SCxCR	0x0008
Mode Control Register 0	SCxMOD0	0x000C
Baud Rate Generator Control Register	SCxBRCR	0x0010
Baud Rate Generator Control Register 2	SCxBRADD	0x0014
Mode Control Register 1	SCxMOD1	0x0018
Mode Control Register 2	SCxMOD2	0x001C
Reception FIFO Configuration Register	SCxRFC	0x0020
Transmission FIFO Configuration Register	SCxTFC	0x0024
Reception FIFO Status Register	SCxRST	0x0028
Transmission FIFO Status Register	SCxTST	0x002C
FIFO Configuration Register	SCxFCNF	0x0030

Note: Do not rewrite the control registers during transmission or reception.

10.3.2. SCxEN (Enable Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	-	SIOE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
0	SIOE	R/W	SIO/UART operation control 0: Disabled 1: Enabled Specifies the operation of SIO/UART. When using SIO/UART, set SCxEN<SIOE> to "1" first. When operation is disabled after SIO/UART is enabled, all register's values except SCxTFC<TIL[1:0]> are retained. When operation is disabled, clocking to SIO/UART except SCxEN is stopped, so power consumption can be reduced.

Note: When SCxEN<SIOE> is set to "0" (disabling SIO/UART operation) or SCxMOD1<I2S0> is set to "0" and operation mode enters to IDLE mode (disabling operation in IDLE mode), re-set SCxTFC after releasing IDLE mode.

10.3.3. SCxBUF (Buffer Register)

SCxBUF is used as a transmission buffer or FIFO when writing, and as a reception buffer or FIFO when reading.

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TB/RB							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:0	RB[7:0]	R	Reception buffer/FIFO
	TB[7:0]	W	Transmission buffer/FIFO

10.3.4. SCxCR (Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	RB8	R	Reception data bit 8 (UART mode) This is the 9th bit of reception data in 9-bit UART mode.
6	EVEN	R/W	Parity bit setting (UART mode) 0: Odd 1: Even Sets the parity bit odd/even. The parity bit is available in 7-bit UART mode and 8-bit UART mode.
5	PE	R/W	Parity bit addition control (UART mode) 0: Disabled 1: Enabled This bit is used to enable/disable parity bit addition. The parity bit is available in 7-bit UART mode and 8-bit UART mode.
4	OERR	R	Overrun error (Note) 0: No error occurs. 1: Error occurs.
3	PERR	R	Parity/underrun error (Note) 0: No error occurs. 1: Error occurs.
2	FERR	R	Framing error (Note) 0: No error occurs. 1: Error occurs.
1	SCLKS	R/W	Clock edge selection (SIO mode) 0: The data of the transmission shift register is output to TXDx pin one bit at a time on the falling edge of SCLKx pin. At the rising edge of SCLKx pin, the data of RXDx pin is input in the reception shift register one bit at a time. At this time, SCLKx pin starts at "High" level. 1: At the rising edge of SCLKx pin, the data of the transmission shift register is output to TXDx pin one bit at a time. At the falling edge of SCLKx pin, the data of RXDx pin is input in reception shift register one bit at a time. At this time, SCLKx pin starts from the "Low" level. When SCxCR <IOC> is set to "0", set SCxCR<SCLKS> to "0".
0	IOC	R/W	Transfer clock input/output selection (SIO mode) 0: SCLKx pin output 1: SCLKx pin input

Note: The error flags (OERR, PERR, and FERR) are cleared when they are read.

10.3.5. SCxMOD0 (Mode Control Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TB8	CTSE	RXE	WU	SM		SC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	TB8	R/W	Transmission data bit 8 (UART mode) This bit is used to set the 9th bit of transmission data in 9-bit UART mode.
6	CTSE	R/W	Handshake function control (UART mode) 0: Handshake function disabled 1: Handshake function enabled Controls the handshake function. When enabled, the handshake function using $\overline{\text{CTSx}}$ pins can be used.
5	RXE	R/W	Reception operation control (Note1) (Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function control (UART mode) 0: Disabled 1: Enabled This bit is valid only in 9-bit UART mode. It has no meaning in other modes. If enabled, an interrupt is generated when the reception data of the 9th bit is "1".
3:2	SM[1:0]	R/W	Serial transfer mode 00: SIO mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode
1:0	SC[1:0]	R/W	SIOCLK clock select (UART mode) 00: TMRB TBxOUT 01: Baud rate generator 10: fsys 11: SCLKx pin input

Note1: After setting SCxMOD0, SCxMOD1, SCxMOD2, set SCxMOD0<RXE>.

Note2: Do not set SCxMOD0<RXE> to "0" during reception.

10.3.6. SCxBRCR (Baud Rate Generator Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	BRADDE	BR0CK		BR0S			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	-	R/W	Write as "0".
6	BRADDE	R/W	N + $\frac{(16 - K)}{16}$ divide control (UART mode) 0: Disabled 1: Enabled In SIO mode, set SCxBRCR<BRADDE> to "0".
5:4	BR0CK[1:0]	R/W	Baud rate generator input clock selection 00: ΦT1 01: ΦT4 10: ΦT16 11: ΦT64
3:0	BR0S[3:0]	R/W	Divide value "N" setting 0000: 16 0001: 1 0010: 2 : 1111: 15

10.3.7. SCxBRADD (Baud Rate Generator Control Register 2)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	BR0K			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:4	-	R	Read as "0".
3:0	BR0K[3:0]	R/W	Divide value "K" setting (UART mode) 0000: Prohibited 0001: 1 0010: 2 : 1111: 15

The divide value settings are shown in Table 10.1.

Table 10.1 Division Value Setting

Parameter	When SCxBRCR<BRADDE>= "0"	When SCxBRCR<BRADDE>= "1" (Note1) (Available only in UART mode.)
SCxBRCR<BR0S> setting	Division value "N" setting (Note2)(Note3)	
SCxBRADD<BR0K> setting	No setting required	Set "K"
Division value	Divided by N	Divided by $N + \frac{(16 - K)}{16}$

Note1: When dividing by $N + \frac{(16 - K)}{16}$, be sure to set SCxBRADD<BR0K>to "K" and then set SCxBRCR <BRADDE> to "1".

Note2: When dividing by $N + \frac{(16 - K)}{16}$ in UART mode, the dividing value "N" cannot be set to "1" and "16".

Note3: Division value "N" can be set to "1" only when double buffer is enabled in SIO mode.

10.3.8. SCxMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	I2S0	FDPX		TXE	SINT			-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	I2S0	R/W	Operation in IDLE mode 0: Stop 1: Operation
6:5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer disabled 01: Half duplex reception 10: Half duplex transmission 11: Full duplex Specifies the transfer mode and FIFO configuration in SIO mode. In UART mode, specify only FIFO configuration.
4	TXE	R/W	Transmission operation control (Note1) (Note2) 0: Disabled 1: Enabled
3:1	SINT[2:0]	R/W	Interval time during continuous transfer (SIO mode) 000: None 001: 1 × SCLKx cycle 010: 2 × SCLKx cycle 011: 4 × SCLKx cycle 100: 8 × SCLKx cycle 101: 16 × SCLKx cycle 110: 32 × SCLKx cycle 111: 64 × SCLKx cycle This bit is enabled when SCLKx pin output is selected as the transfer clock in SIO mode. It does not mean in other modes. Specifies the interval between continuous transfers when double buffer or FIFO is enabled in SIO mode.
0	-	R/W	Write as "0".

Note1: After setting SCxMOD0, SCxMOD1, SCxMOD2, set SCxMOD1<TXE>.

Note2: Do not set SCxMOD1<TXE> to "0" during transmission.

Note3: When SCxEN<SIOE> is set to "0" (disabling SIO/UART operation) or SCxMOD1<I2S0> is set to "0" and operation mode enters to IDLE mode (disabling operation in IDLE mode), re-set SCxTFC after releasing IDLE mode.

10.3.9. SCxMOD2 (Mode Control Register 2)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST	
After reset	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	TBEMP	R	Transmission buffer empty flag 0: Buffer full 1: Buffer empty This flag indicates the buffer empty state of the transmission buffer. When data in the transmission buffer is transferred to the transmission shift register and transmission buffer becomes buffer empty state, this bit is set to "1". And when transmission data is written to the transmission buffer, this bit is set to "0". When double buffer is disabled, this flag has no meaning.
6	RBFLL	R	Reception buffer full flag 0: Buffer empty 1: Buffer full This flag indicates the buffer full state of the reception buffer. When the reception operation is completed and data is stored in the reception buffer from the reception shift register, this bit is set to "1". And when the received data is read from the reception buffer, this bit is set to "0". When double buffer is disabled, this flag has no meaning.
5	TXRUN	R	Transmission operation flag 0: Stop 1: Operation This flag indicates transmission operation. SCxMOD2<TXRUN> and SCxMOD2<TBEMP> indicate the following conditions.
4	SBLEN	R/W	Transmission STOP bit length selection (UART mode) 0: 1 bit 1: 2 bits Selects the transmission STOP bit length. The reception STOP bit length is only 1 bit regardless of SCxMOD2<SBLEN>.
3	DRCHG	R/W	Transfer direction selection 0: LSB first 1: MSB first Selects the transfer direction in SIO mode. In UART mode, set SCxMOD2<DRCHG> to "0".

SCxMOD2 <TXRUN>	SCxMOD2 <TBEMP>	Transmission status
1	-	Transmission operation in progress
0	1	End of transmission
	0	Transmission buffer contains the next data and waits for transmission

<table>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Double buffer control
0: Disabled
1: Enabled
This bit is available when SCLKx pin output is selected as the transfer clock or when SCLKx pin input is selected as the transfer clock for transmission in SIO mode.
In UART mode, it is available when Transmitting.
When reception with SCLKx pin input selected in SIO mode or reception in UART mode, double buffer is enabled regardless of SCxMOD2<WBUF>.</td></tr>
<tr><td>1:0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Writing "10" → "01" in this order generates a software reset.
The following bits are initialized by a software reset: In addition, the transmission/reception circuit and FIFO are initialized. (Note1) (Note2)
<table>
<tr><th>Register name</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td><RXE></td></tr>
<tr><td>SCxMOD1</td><td><TXE></td></tr>
<tr><td>SCxMOD2</td><td><TBEMP>, <RBFLL>, <TXRUN></td></tr>
<tr><td>SCxCR</td><td><OERR>, <PERR>, <FERR></td></tr>
</table></td></tr>
</table>

Note1: To perform a software reset during transfer operation, perform this procedure twice consecutively.

Note2: Two clocks are required after the instruction is executed to complete the software reset operation.

10.3.10. SCxRFC (Reception FIFO Configuration Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	RFCS	RFIS	-	-	-	-	RIL	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	RFCS	R	Read as "0".
		W	FIFO for reception clearing control (Note) 0: Don't care 1: Clear Writing "1" initializes the used FIFO and read pointer. And initializes SCxRST<RLVL[2:0]> to "000".
6	RFIS	R/W	Interrupt generation condition select when FIFO is enabled 0: FIFO level reaches to FILL level. 1: FIFO level reaches to FILL level or exceeds FILL level when new data is read.
5:2	-	R	Read as "0".
1:0	RIL[1:0]	R/W	When FIFO is enabled, FILL level at which an interrupt is generated is set.

SCxRFC <RIL[1:0]>	Half duplex	Full duplex
00	Full	Full
01	1 byte	1 byte
10	2 bytes	Full
11	3 bytes	1 byte

Note: When FIFO is enabled, clear FIFO for transmission/reception after setting the transfer mode of the SIO/UART (SCxMOD1<FDPX[1:0]>) and FIFO is enabled (set SCxFCNF<CNFG> to "1").

10.3.11. SCxTFC (Transmission FIFO Configuration Register) (Note2)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TFCS	TFIS	-	-	-	-	TIL	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	TFCS	R	Read as "0".
		W	FIFO for transmission clearing control (Note1) 0: Don't care 1: Clear Writing "1" initializes the used FIFO and the write pointer. And initializes SCxTST<TLVL[2:0]> to "000".
6	TFIS	R/W	Interrupt generation condition selection when FIFO is enabled 0: FIFO level reaches to FILL level. 1: FIFO level reaches to FILL level or does not reach to FILL level when new data is written.
5:2	-	R	Read as "0".
1:0	TIL[1:0]	R/W	When FIFO is enabled, FILL level at which an interrupt is generated is set.

SCxTFC <TIL[1:0]>	Half duplex	Full duplex
00	Empty	Empty
01	1 byte	1 byte
10	2 bytes	Empty
11	3 bytes	1 byte

Note1: When FIFO is used, clear FIFO for transmission/reception after setting the transfer mode of the SIO/UART (SCxMOD1<FDPX[1:0]>) and FIFO is enabled (set SCxFCNF<CNFG> to "1").

Note2: When SIO/UART operation is disabled (set SCxEN<SIOE> to "0") or SCxMOD1<I2S0> is set to "0", re-set SCxTFC after operation mode enters to IDLE mode (operation disabled during IDLE mode).

10.3.12. SCxRST (Reception FIFO Status Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	ROR	-	-	-	-	RLVL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	ROR	R	Overrun error when FIFO is enabled (Note) 0: No error occurs. 1: Error occurs.
6:3	-	R	Read as "0".
2:0	RLVL[2:0]	R	FILL level of FIFO for reception 000: 0 bytes 001: 1 byte 010: 2 bytes 011: 3 bytes 100: FULL

Note: SCxRST<ROR> is set to "0" when SCxBUF is read.

10.3.13. SCxTST (Transmission FIFO Status Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TUR	-	-	-	-	TLVL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	TUR	R	Underrun error when FIFO is enabled (Note) 0: No error occurs. 1: Error occurs.
6:3	-	R	Read as "0".
2:0	TLVL[2:0]	R	FILL level of FIFO for transmission 000: Empty 001: 1 byte 010: 2 bytes 011: 3 bytes 100: 4 bytes

Note: SCxTST<TUR> is set to "0" when SCxBUF is written.

10.3.14. SCxFCNF(FIFO Configuration Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	RFST	TFIE	RFIE	RXTXCNT	CNFG
After reset	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31:8</td><td>-</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>7:5</td><td>-</td><td>R/W</td><td>Write as "0".</td></tr>
<tr><td>4</td><td>RFST</td><td>R/W</td><td>Setting the number of byte of FIFO which is used during reception (Note1)
0: Maximum
1: Same as FILL level of FIFO for reception
When this bit is set to "0", the largest number of bytes of FIFO which is configured can be used.
When this bit is set to "1", the number of bytes of FILL level which is set by SCxRFC<RIL[1:0]> can be used.
Refer to SCxFCNF<CNFG> for the number of bytes of FIFO which can be used.</td></tr>
<tr><td>3</td><td>TFIE</td><td>R/W</td><td>Transmission interrupt generation control when FIFO is enabled during transmission
0: Disabled
1: Enabled
This bit disables or enables transmission interrupt generation when FIFO is enabled.</td></tr>
<tr><td>2</td><td>RFIE</td><td>R/W</td><td>Reception interrupt generation control when FIFO is enabled during reception
0: Disabled
1: Enabled
This bit disables or enables reception interrupt generation when FIFO is enabled.</td></tr>
<tr><td>1</td><td>RXTXCNT</td><td>R/W</td><td>Automatic disabling control of SCxMOD0<RXE>/SCxMOD1<TXE>
0: Disabled
1: Enabled
The SIO/UART operates as shown below according to the transfer mode.
<table>
<tr><th>Transfer Mode</th><th>Operation</th></tr>
<tr><td>Half duplex reception</td><td>When the specified number of bytes of data is stored in the reception shift register, reception buffer, and FIFO, SCxMOD0<RXE> is automatically set to "0" and reception operation is disabled.</td></tr>
<tr><td>Half duplex transmission</td><td>When all data in FIFO, transmission buffer, and transmission shift register are transmitted, SCxMOD1<TXE> is automatically set to "0" and the transmission operation is disabled.</td></tr>
<tr><td>Full duplex</td><td>When the condition either half-duplex reception or half-duplex transmission is satisfied, both SCxMOD1<TXE> and SCxMOD0<RXE> are automatically set to "0" and transmission/reception operation is disabled.</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td>0</td>
<td>CNFG</td>
<td>R/W</td>
<td>Enable FIFO control (Note2)
0: Disabled
1: Enabled
According to the transfer mode, FIFO configuration is as follows:
<table>
<tr><th>Transfer Mode</th><th>FIFO configuration</th></tr>
<tr><td>Half duplex reception</td><td>Reception: 4 bytes</td></tr>
<tr><td>Half duplex transmission</td><td>Transmission: 4 bytes</td></tr>
<tr><td>Full duplex</td><td>Reception: 2 bytes + Transmission: 2 bytes</td></tr>
</table>
</td>
</tr>
</table>

Note1: The number of bytes of FIFO which is used during transmission is always the largest byte.

Note2: The FIFO cannot be used in 9-bit UART mode.

10.4. Mode

Table 10.2 shows modes and data formats.

Table 10.2 Modes and Data Formats

Mode	Type	Data length	Transfer direction	Parity bit addition	STOP bit length
Mode 0	Synchronous communication mode (SIO mode)	8 bits	LSB first or MSB first	-	-
Mode 1	Asynchronous communication mode (UART mode)	7 bits	LSB first	✓	Transmission: 1 bit or 2 bits Reception: 1 bit
Mode 2		8 bits		✓	
Mode 3		9 bits		-	

✓: Available, -: Not available

Mode 0 is SIO mode and is used to extend the I/O.

Transmission and reception data is in synchronization with SCLKx pin. SCLKx pin can be used for input/output.

The transfer direction is selected from LSB first or MSB first.

Both parity bit addition and STOP bit addition are not supported.

Modes 1 to 3 are UART modes.

Transfer direction is LSB first only.

In mode 1 and mode 2, parity bit can be added. In mode 3, the parity bit cannot be added because the data length is 9 bits.

STOP bit can be added. STOP bit for transmission can be selected from 1 bit or 2 bits. STOP bit length for reception is 1 bit only.

Mode 3 supports the wake-up function for the master controller to activate the slave controller in serial link (multi-controller system).

10.5. Data Format

10.5.1. Data Format List

Figure 10.2 shows a data format.

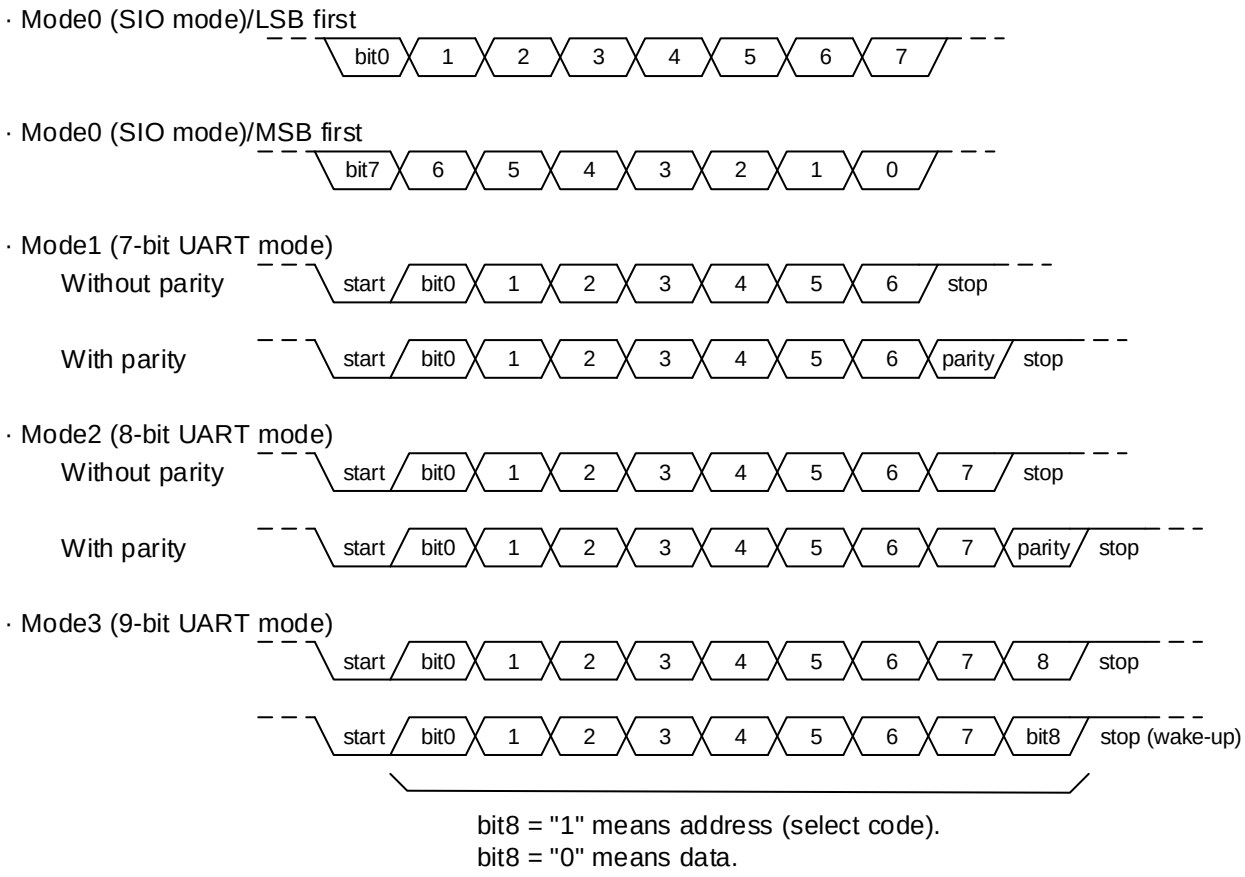


Figure 10.2 Data Format

10.5.2. Parity Bit Control

In mode 1 and mode 2, parity bit can be added to the transfer data.

When SCxCR<PE> is set to "1", parity bit addition is enabled.

A parity bit odd or even is selected by SCxCR<EVEN>.

10.5.2.1. During Transmission

During transmission, the parity control circuit automatically generates a parity bit for the buffer register data.

In mode 1, parity bit is stored in SCxBUF<TB7>.

In mode 2, parity bit is stored in SCxMOD0<TB8>.

SCxCR<PE> and SCxCR<EVEN> should be set before writing transmission data to the buffer registers.

10.5.2.2. During Reception

During reception, the parity control circuit automatically generates a parity bit when data is stored in the reception buffer register from the reception shift register.

In mode 1, SCxBUF<RB7> is compared with the generated parity bit.

In mode 2, SCxCR<RB8> is compared with the generated parity bit.

A parity error occurs when the compared result is not matched, and SCxCR<PERR> is set to "1".

When using FIFO, SCxCR<PERR> indicates that a parity error has occurred on any of the reception data.

10.5.3. STOP Bit Length

When the serial transfer mode is UART mode, STOP bit length for transmission is selected from 1 bit or 2 bits by SCxMOD2<SBLEN>.

STOP bit length for reception is only 1 bit regardless of SCxMOD2<SBLEN>.

10.6. Clock Control

10.6.1. Prescaler

A 7-bit prescaler is built-in to generate a 2/8/32/128 divided clock of ΦT0.

The input clock ΦT0 of the prescaler is selected from fgear and fc by CGSYSCR in the clock/mode control circuit.

The conditions to use the clock that is generated by the prescaler are as follows.

- SIO mode

  Set SCxCR<IOC> to "0" to select SCLKx pin output as the transfer clock.

- UART mode

  Set SCxMOD0<SC[1:0]> to "01" to select the baud rate generator as the serial transfer clock.

Table 10.3 and Table 10.4 show the resolution of the output clock from the prescaler.

Table 10.3 Resolution of Output Clock from Prescaler (fc = 80MHz) (1/2)

CGSYSCR <FPSEL>	CGSYSCR <GEAR[2:0]>	CGSYSCR <PRCK[2:0]>	Resolution of output clock from prescaler			
			ΦT1	ΦT4	ΦT16	ΦT64
0 (fgear)	000 (fc)	000 (fperiph / 1)	-	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)
		001 (fperiph / 2)	0.05μs (fc / 2^2)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		010 (fperiph / 4)	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		011 (fperiph / 8)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		100 (fperiph / 16)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		101 (fperiph / 32)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
	100 (fc / 2)	000 (fperiph / 1)	-	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		001 (fperiph / 2)	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		010 (fperiph / 4)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		011 (fperiph / 8)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		100 (fperiph / 16)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
		101 (fperiph / 32)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})	102.4μs (fc / 2^{13})
	101 (fc / 4)	000 (fperiph / 1)	-	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		001 (fperiph / 2)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		010 (fperiph / 4)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		011 (fperiph / 8)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
		100 (fperiph / 16)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})	102.4μs (fc / 2^{13})
		101 (fperiph / 32)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})	204.8μs (fc / 2^{14})
	110 (fc / 8)	000 (fperiph / 1)	-	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		001 (fperiph / 2)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		010 (fperiph / 4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
		011 (fperiph / 8)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})	102.4μs (fc / 2^{13})
		100 (fperiph / 16)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})	204.8μs (fc / 2^{14})
		101 (fperiph / 32)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})	102.4μs (fc / 2^{13})	409.6μs (fc / 2^{15})
	111 (fc /16)	000 (fperiph / 1)	-	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		001 (fperiph / 2)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
		010 (fperiph / 4)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})	102.4μs (fc / 2^{13})
		011 (fperiph / 8)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})	204.8μs (fc / 2^{14})
		100 (fperiph / 16)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})	102.4μs (fc / 2^{13})	409.6μs (fc / 2^{15})
		101 (fperiph / 32)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})	204.8μs (fc / 2^{14})	819.2μs (fc / 2^{16})

Note1: Always select the output clock ΦTn from the prescaler so that ΦTn < fsys / 2 is satisfied (ΦTn becomes slower than fsys / 2).

Note2: Do not change CGSYSCR<FPSEL>, CGSYSCR<GEAR[2:0]>, CGSYSCR<PRCK[2:0]> during SIO/UART operation.

Note3: "-" means the prohibited setting.

Table 10.4 Resolution of Output Clock from Prescaler (fc = 80MHz) (2/2)

CGSYSCR <FPSEL>	CGSYSCR <GEAR[2:0]>	CGSYSCR <PRCK[2:0]>	Resolution of output clock from prescaler			
			ΦT1	ΦT4	ΦT16	ΦT64
1 (fc)	000 (fc)	000 (fperiph / 1)	-	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)
		001 (fperiph / 2)	0.05μs (fc / 2^2)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		010 (fperiph / 4)	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		011 (fperiph / 8)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		100 (fperiph / 16)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		101 (fperiph / 32)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
	100 (fc / 2)	000 (fperiph / 1)	-	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)
		001 (fperiph / 2)	-	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		010 (fperiph / 4)	0.1μs (fc / 2^3)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		011 (fperiph / 8)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		100 (fperiph / 16)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		101 (fperiph / 32)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
	101 (fc / 4)	000 (fperiph / 1)	-	-	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)
		001 (fperiph / 2)	-	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		010 (fperiph / 4)	-	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		011 (fperiph / 8)	0.2μs (fc / 2^4)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		100 (fperiph / 16)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		101 (fperiph / 32)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
	110 (fc / 8)	000 (fperiph / 1)	-	-	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)
		001 (fperiph / 2)	-	-	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		010 (fperiph / 4)	-	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		011 (fperiph / 8)	-	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		100 (fperiph / 16)	0.4μs (fc / 2^5)	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		101 (fperiph / 32)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})
	111 (fc /16)	000 (fperiph / 1)	-	-	-	1.6μs (fc / 2^7)
		001 (fperiph / 2)	-	-	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)
		010 (fperiph / 4)	-	-	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)
		011 (fperiph / 8)	-	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})
		100 (fperiph / 16)	-	1.6μs (fc / 2^7)	6.4μs (fc / 2^9)	25.6μs (fc / 2^{11})
		101 (fperiph / 32)	0.8μs (fc / 2^6)	3.2μs (fc / 2^8)	12.8μs (fc / 2^{10})	51.2μs (fc / 2^{12})

Note1: Always select the output clock ΦTn from the prescaler so that ΦTn < fsys / 2 is satisfied (ΦTn becomes slower than fsys / 2).

Note2: Do not change CGSYSCR<FPSEL>, CGSYSCR<GEAR[2:0]>, CGSYSCR<PRCK[2:0]> during SIO/UART operation.

Note3: "-" means the prohibited setting.

10.6.2. Serial Clock Generation Circuit

This circuit generates the transmission/reception clock (SIOCLK).

It consists of a baud rate generator and a clock selection circuit.

10.6.2.1. Baud Rate Generator

The baud rate generator is the circuit which is used to generate clocks that determine SIO/UART transfer rate.

The baud rate generator input clock is divided by the divider circuit to output the baud rate generator output clock.

(1) Baud rate generator input clock

The input clock of the baud rate generator is selected from the 2/8/32/128 divided clock output from the prescaler. The divided clocks are selected by SCxBRCR<BR0CK[1:0]>.

(2) Baud rate generator output clock

The divider circuit consists of an N divider circuit and an $N + \frac{(16 - K)}{16}$ divider circuit.

The divide value is set by SCxBRCR<BRADDE>, <BR0S[3:0]>, SCxBRADD<BR0K>.

Table 10.5 shows the division setting and divider values which can be used for each serial transfer mode.

Table 10.5 Settable Division Values

Serial Transfer Mode	Division setting SCxBRCR<BRADDE>	Divide value "N" SCxBRCR<BR0S[3:0]>	Divide value "K" SCxBRADD<BR0K>
SIO mode	0: $N + \frac{(16 - K)}{16}$ division disabled (N division is used)	1 to 16 (Note)	No meaning
UART mode	0: $N + \frac{(16 - K)}{16}$ division disabled (N division is used)	1 to 16	No meaning
	1: $N + \frac{(16 - K)}{16}$ division enable	2 to 15	1 to 15

Note: The division by 1 can be used only when the double buffer is enabled.

10.6.2.2. Clock Selection Circuit

This circuit selects the clock depending on the serial transfer mode.

Serial transfer mode is specified by SCxMOD0<SM[1:0]>.

The baud rate generator output clock in SIO mode is specified by SCxCR<IOC> and SCxCR<SCLKS>.

The baud rate generator output clock in UART mode is specified by SCxMOD0<SC[1:0]>.

(1) Transfer clock in SIO mode

Table 10.6 shows register setting for transfer clocks that can be used in SIO mode.

Table 10.6 Register Settings for Transfer Clock that Can be Used in SIO Mode

Serial transfer Mode SCxMOD0<SM[1:0]>	Transfer clock input/output selection SCxCR <IOC>	Clock edge selection SCxCR<SCLKS>	Transfer clock
00: SIO mode	0: SCLKx pin output	0: Transmission: Falling edge Reception: Rising edge	Clock which is divided baud rate generator output clock by 2
	1: SCLKx pin input	0: Transmission: Falling edge Reception: Rising edge	SCLKx pin input clock
		1: Transmission: Rising edge Reception: Falling edge	

The maximum frequency of the transfer clock is shown below.

Note: Set the clock so that the AC electrical characteristics in chapter "24.7. AC Electrical Characteristics" are satisfied.

– Clock/mode control circuit setting
  - fc = 80MHz
  - fgear = 80MHz (CGSYSCR<GEAR[2:0]>= "000", selects fc)
  - fperiph = 80MHz (CGSYSCR<FPSEL>= "0", selects fgear)
  - ΦT0 = 80MHz (CGSYSCR<PRCK[2:0]>= "000", selects the clock which is divided fperiph by one)

- When using SCLKx pin output for the transfer clock
  – Setting SIO/UART when double buffer is enabled
    - Prescaler output clock selection (SCxBRCR<BR0CK[1:0]>="00", selects ΦT1) = 40MHz
    - Divide value "N" (SCxBRCR<BR0S[3:0]>= "0001", divided by 1), divider output = 40MHz
      When double buffer is enabled, the divide value "N" can be set to "1".
    - SIOCLK (divider output frequency / 2) = 20MHz

  – Setting SIO/UART when double buffer is disabled
    - Prescaler output clock (SCxBRCR<BR0CK[1:0]>="00", selects ΦT1) = 40MHz
    - Divide vale "N" (SCxBRCR<BR0S[3:0]>= "0010", divided by 2), divider output = 20MHz
      When double buffer is disabled, the minimum value to which the divide value "N" can be set is "2".
    - SIOCLK (divider output frequency / 2) = 10MHz

- When using SCLKx pin input as the transfer clock
  – Setting SIO/UART when double buffer is enabled
    - SCLKx pin input transfer clock cycle > 6 / fsys
      The maximum frequency of the transfer clock is less than f_{SCLKx} = 80MHz / 6 = 13.3MHz@fsys = 80MHz.

  – Setting SIO/UART when double buffer is disabled
    - SCLKx pin input transfer clock cycle > 8 / fsys
      The maximum frequency of the transfer clock is less than f_{SCLKx} = 80MHz / 8 = 10MHz@fsys = 80MHz.

(2) Transfer clock in UART mode

Table 10.7 shows the register settings for SIOCLK clock which can be used in UART mode.

In UART mode, the reception/transmission counter divides SIOCLK clock selected in SCxMOD0<SC[1:0]> by 16.

Table 10.7 Register Settings for SIOCLK Clock which Can be Used in UART Mode

Mode SCxMOD0<SM[1:0]>	SIOCLK clock selection SCxMOD0<SC[1:0]>
01: UART mode 10: UART mode 11: UART mode	00: TMRB TBxOUT
	01: Baud rate generator
	10: fsys
	11: SCLKx pin input

The baud rate for each SIOCLK clock selected in SCxMOD0<SC[1:0]> is shown below.

- When using TMRB TBxOUT

  When using TMRB TBxOUT, select the UC source clock beforehand, and set TBxFF0 output invert when matching the UC and TBxRG1. The baud rate calculation formula when ΦT1 is selected as the UC source clock is shown below.

Baud rate calculation formula:

$$\text{Baud rate} = \frac{\Phi T1}{(TBxRG1 \times 2) \times 2 \times 16}$$

When timer prescaler clock ΦT1 (divided by 2) is selected

One clock is generated when TBxFF0 is inverted twice.

Table 10.8 shows examples of baud rate when using TMRB TBxOUT with the clock setting shown below.

- Clock/mode control circuit setting
  - fc = 80MHz, 9.8304MHz, 8MHz
  - fgear = 80MHz, 9.8304MHz, 8MHz (CGSYSCR<GEAR[2:0]>= "000", selects fc)
  - fperiph = 80MHz, 9.8304MHz, 8MHz (CGSYSCR<FPSEL>= "0", selects fgear)
  - ΦT0 = 80MHz, 9.8304MHz, 8MHz (CGSYSCR<PRCK[2:0]>= "000", selects the clock which is divided fperiph by one)

- TMRB setting
  - UC source clock = 40MHz, 4.9152MHz, 4MHz (TBxMOD<TBCLK[1:0]>= "01", selects ΦT1)

Table 10.8 Example of Baud Rate in UART Mode. (Unit: kbps)

TBxRG1 setting	fc (MHz)		
	80	9.8304	8
0x0001	625.0	76.8	62.5
0x0002	312.5	38.4	31.25
0x0003	-	25.6	-
0x0004	156.25	19.2	15.625
0x0005	125.0	15.36	12.5
0x0006	-	12.8	-
0x0008	78.125	9.6	-
0x000A	62.5	7.68	6.25
0x0010	39.025	4.8	-
0x0014	31.25	3.84	3.125

- When using the baud rate generator

The maximum baud rate when using the baud rate generator is shown below.

– Clock/mode control circuit setting
  - fc = 80MHz
  - fgear = 80MHz (CGSYSCR<GEAR[2:0]>= "000", selects fc)
  - fperiph = 80MHz (CGSYSCR<FPSEL>= "0", selects fgear)
  - ΦT0 = 80MHz (CGSYSCR<PRCK[2:0]>= "000", selects the clock which is divided fperiph by one)

– SIO/UART setting
  - ΦT1 = 40MHz (SCxBRCR<BR0CK[1:0]>= "00", selects ΦT1)
  - When SCxBRCR<BRADDE> is set to "0"
    When the divide value "N" is set to "1", SIOCLK is 40MHz. The baud rate is 2.5Mbps when SIOCLK is divided by 16.
  - When SCxBRCR<BRADDE> is set to "1"
    When the divide value "N" is set to "2" and the divide value "K" is set to "15", SIOCLK is 19.4MHz. The baud rate is 1.2Mbps when SIOCLK is divided by 16.

Table 10.9 shows examples of baud rate when the baud rate generator is used for below settings.

– Clock/mode control circuit setting
  - fc = 9.8304MHz
  - fgear = 9.8304MHz (CGSYSCR<GEAR[2:0]>= "000", selects fc)
  - fperiph = 9.8304MHz (CGSYSCR<FPSEL>= "0", selects fgear)
  - ΦT0 = 4.9152MHz (CGSYSCR<PRCK[2:0]>= "001", selects the clock which is divided fperiph by two)

– SIO/UART setting
  - When SCxBRCR<BRADDE> is set to "0"

Table 10.9 Example Baud Rate in UART Mode (Unit: kbps)

fc (MHz)	Divide value "N" (SCxBRCR<BRS[3:0]>)	ΦT1 (ΦT0 / 2)	ΦT4 (ΦT0 / 8)	ΦT16 (ΦT0 / 32)	ΦT64 (ΦT0 / 128)
9.830400	0010: 2	76.800	19.200	4.800	1.200
	0100: 4	38.400	9.600	2.400	0.600
	1000: 8	19.200	4.800	1.200	0.300
	0000: 16	9.600	2.400	0.600	0.150

- When using SCLKx pin
  - SCLKx pin input cycle > 2 / fsys

    Since SIOCLK is less than 40MHz, the maximum baud rate is 40MHz / 16 = less than 2.5Mbps.

- When using fsys

  Since fsys is up to 80MHz, the maximum baud rate is 80 / 16 = 5Mbps.

10.7. Transmission/Reception Buffers and FIFO

10.7.1. Block Diagram

The configuration of the transmission/reception buffers and FIFO is shown in Figure 10.3.

To use the transmission/reception buffers and FIFO, the registers of SIO/UART must be set beforehand. In addition, FIFO configuration depends on the transfer mode.

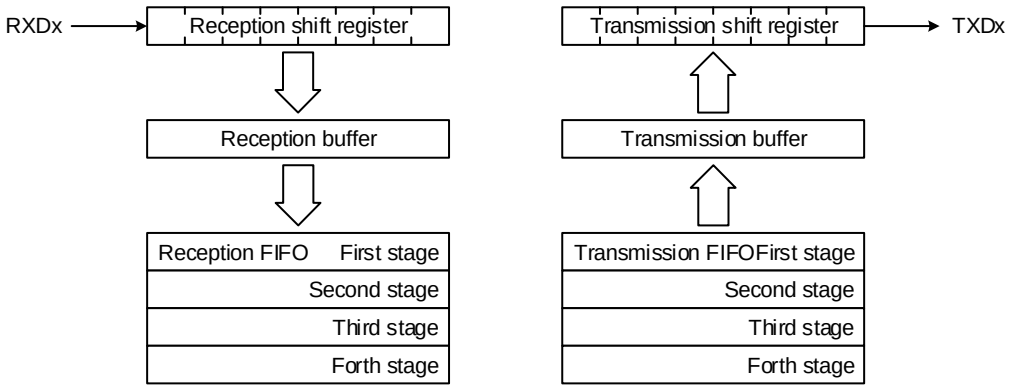


Figure 10.3 Configuration of Transmission/Reception Buffers and FIFO

10.7.2. Transmission/Reception Buffers

The transmission/reception buffers can be selected to enable or disable the double buffer. Enabling or disabling the double buffer is specified by SCxMOD2<WBUF>.

When SCLKx pin input is used as the transfer clock of reception in SIO mode, or when reception in UART mode, double buffer is enabled regardless of the setting of SCxMOD2<WBUF>.

Otherwise, enabling or disabling the double buffer depends on the setting of SCxMOD2<WBUF>.

Table 10.10 shows the relationship between serial transfer mode and buffer configuration.

Table 10.10 Serial Transfer Mode and Buffer Configuration

Serial Transfer Mode		SCxMOD2<WBUF>	
		0	1
UART mode	Transmission	Double buffer disabled	Double buffer enabled
	Reception	Double buffer enabled	Double buffer enabled
SIO mode (SCLKx pin input)	Transmission	Double buffer disabled	Double buffer enabled
	Reception	Double buffer enabled	Double buffer enabled
SIO mode (SCLKx pin output)	Transmission	Double buffer disabled	Double buffer enabled
	Reception	Double buffer disabled	Double buffer enabled

10.7.3. FIFO

In addition to double buffer, 4 bytes FIFO can be used.

To enable FIFO, set SCxMOD2<WBUF> to "1" and SCxFCNF<CNFG> to "1".

FIFO configuration is set in SCxMOD1<FDPX[1:0]>.

Note: When FIFO is used, be sure to clear FIFO for transmission/reception after setting the transfer mode of the SIO/UART (half duplex or full duplex) and FIFO is enabled (set SCxFCNF<CNFG> to "1").

Table 10.11 shows the relationship between the transfer mode and FIFO configuration.

Table 10.11 Transfer Mode and FIFO Configuration

Transfer mode	SCxMOD1<FDPX[1:0]>	Reception FIFO	Transmission FIFO
Half duplex reception	01	4 bytes	-
Half duplex transmission	10	-	4 bytes
Full duplex	11	2 bytes	2 bytes

10.8. Status Flag

SCxMOD2<RBFLL> is a flag which indicates the reception buffer full. This flag is set to "1" when data is stored in the reception buffer from the reception shift register after data reception is completed. This bit is set to "0" when the reception buffer is read.

SCxMOD2<TBEMP> is a flag that indicates transmission buffer empty. This flag is set to "1" when data is transferred from the transmission buffer to the transmission shift register. This flag is set to "0" when data is written to the transmission buffer.

These status flags have meaning only when double buffer is enabled.

10.9. Error Flags

These flags are cleared to "0" when SCxCR is read.

Table 10.12 Error Flags

Serial Transfer Mode	Flag		
	SCxCR<OERR>	SCxCR<PERR>	SCxCR<FERR>
UART mode	Overrun error	Parity error	Framing error
SIO mode (SCLKx pin input)	Overrun error	Underrun error (Double buffer enabled or Double buffer and FIFO enabled)	"0"
		"0" (Double buffer and FIFO are disabled.).	
SIO mode (SCLKx pin output)	No meaning	End of transmission	"0"

10.9.1. OERR Flag

An overrun error occurs in UART mode or SIO mode when the reception of the next data is completed before reading the data in the reception buffer, and OERR flag is set to "1".

When FIFO is enabled, the data in the reception buffer is automatically transferred to FIFO. Therefore, OERR flag is not set to "1" until FIFO becomes full.

When SCLKx pin output is used for the transfer clock in SIO mode, SCLKx pin output is stopped when all data is stored in the available reception buffer and FIFO, so OERR flag has no meaning.

Note: When SCLKx pin output is used for the transfer clock in SIO mode, read SCxCR and clear OERR flag before changing the transfer mode and transfer clock.

10.9.2. PERR Flag

PERR flag indicates a parity error in UART mode, or an underrun error or completion of transmission in SIO mode.

A parity error occurs when the parity bit generated from the received data differs from the received parity bit, and PERR flag is set to "1".

An underrun error occurs when there is no next data to be transmitted in the double buffer or FIFO, and when the next transfer clock is input, and PERR flag is set to "1".

When SCLKx pin output is used for the transfer clock in SIO mode, SCLKx pin output is automatically stopped when all data is transmitted, so an underrun error does not occur.

Note: When SCLKx pin output is used for the transfer clock in SIO mode, read SCxCR to clear PERR flag before changing the transfer mode and transfer clock.

10.9.3. FERR Flag

Framing error occurs when STOP bit is sampled near the middle of STOP bit timing in UART mode and it is "0", and FERR flags is set to "1".

In SIO mode, this bit is always "0".

10.10. Reception

10.10.1. Reception Counter

The reception counter is a 4-bit binary counter.

In SIO mode, it is incremented by SIOCLK clock.

In UART mode, it is increased by 16 SIOCLK clocks.

10.10.2. Reception Control

10.10.2.1. For SIO Mode

When SCLKx pin output is used as the transfer clock, RXDx pin is captured at the rising edge of SCLKx pin output.

When SCLKx pin input is used as the transfer clock, RXDx pin is captured at the rising or falling edge of SCLKx pin input depending on SCxCR<SCLKS> setting.

10.10.2.2. For UART Mode

The reception control circuit has a start bit detection circuit. It detects the correct start bit and starts reception operation.

For data reception, RXDx pin is sampled at the 7th, 8th, and 9th clock of SIOCLK, and the received data is determined by the majority decision.

10.10.3. Reception Operation

10.10.3.1. Reception Buffer Operation

The received data is stored in the reception shift register one by one, and a reception interrupt (INTRXx) is generated when reception is completed.

When double buffer is enabled, the reception buffer full flag (SCxMOD2<RBFLL>) is set to "1" when the reception operation is completed and the received data is stored from the reception shift register to the reception buffer. SCxMOD2<RBFLL> is set to "0" when data is read from the reception buffer.

When double buffer is disabled, SCxMOD2<RBFLL> has no meaning.

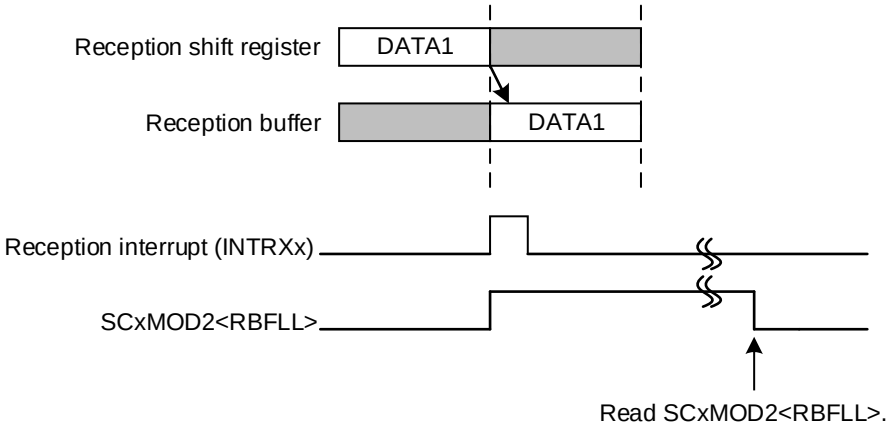


Figure 10.4 Reception Buffer Operation

10.10.3.2. Operation of FIFO

When FIFO is enabled, SCxMOD2<RBFLL> is "0" as soon as the reception data is transferred from the reception buffer to FIFO after the reception operation is completed.

An interrupt is generated according to SCxRFC<RIL[1:0]> setting.

The setting and operation for half duplex reception of 4-byte data are shown below.

SCxMOD1<FDPX[1:0]>= "01"	: Set transfer mode to half-duplex reception
SCxFCNF<RFST><TFIE><RFIE>	: Enable automatic disabling after reaching FILL level.
<RXTXCNT><CNFG>= "10111"	The number of byte of the used FIFO is same as FILL level which generates an interrupt.
SCxRFC<RIL[1:0]>= "00"	: FILL level which generates an interrupt is set to 4 bytes.
SCxRFC<RFCS><RFIS>= "11"	: Clears FIFO and sets interrupt generation conditions.
SCxFCNF<CNFG>= "1"	: FIFO is enabled.

After FIFO enable is set in the above setting, data reception starts when SCxMOD0<RXE> is set to "1". When 4-byte data is stored in FIFO, INTRXx occurs. When data is stored in the reception shift register, reception buffer, and reception FIFO, SCxMOD0<RXE> is automatically set to "0" to terminate reception.

When automatic disabling after reaching FILL level is disabled in the above settings, the reception operation is continued when data in FIFO is read.

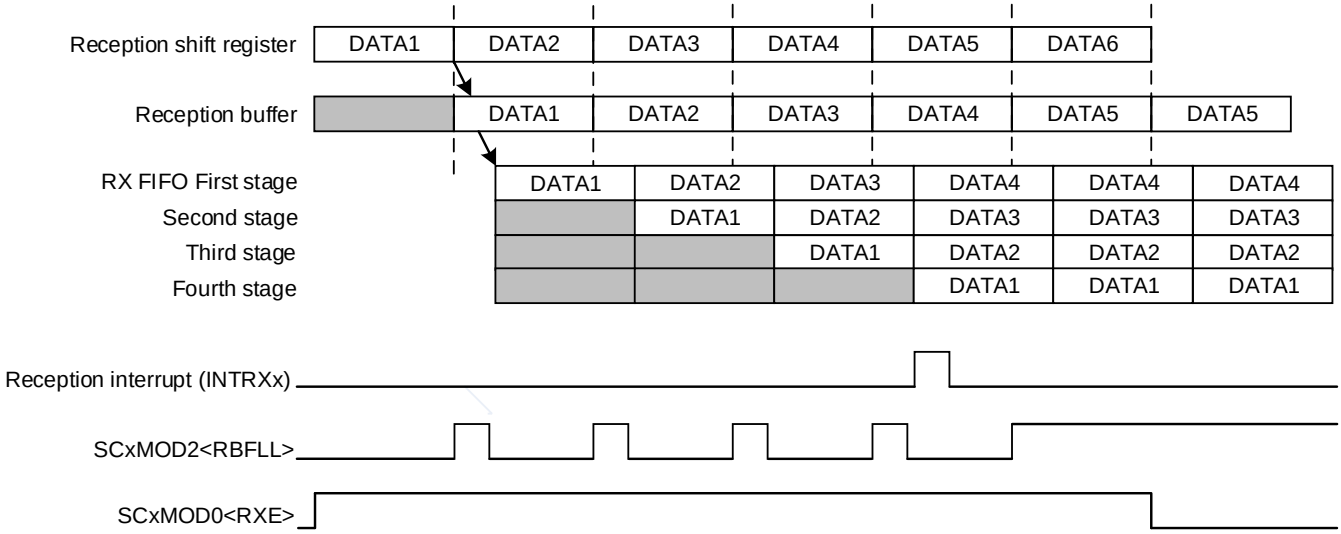


Figure 10.5 Reception FIFO Operation

10.10.3.3. Reception in SIO Mode and SCLKx Pin Output

The timing of stopping/restarting SCLKx pin output varies depending on the reception buffer and FIFO enabling/disabling.

(1) When double buffer is disabled

When data is stored in the reception shift register, SCLKx pin output is stopped. SCLKx pin output restarts when data is read from the reception buffer.

(2) When double buffer is enabled

When data is stored in both the reception shift register and reception buffer, SCLKx pin output is stopped. When data in reception buffer is read, the received data is transferred from the reception shift register to the reception buffer, and SCLKx pin output restarts.

(3) When FIFO is enabled

When data is stored in the reception shift register, reception buffer, and FIFO, SCLKx pin output is stopped. When data in FIFO is read, the received data is transferred from the reception buffer to FIFO and from the reception shift register to the reception buffer, restarts SCLKx pin output.

Note that when SCxFCNF<RXTXCNT> is set to "1", SCLKx pin output is stopped and SCxMOD0<RXE> is set to "0". And the reception operation is stopped.

10.10.3.4. Reading Reception Data

Reads data from SCxBUF regardless of whether FIFO is enabled or disabled.

When FIFO is disabled, SCxMOD2<RBFLL> is set to "0" by reading.

The next reception data can still be stored in the reception shift register before reading the reception buffer. The MSB is stored in SCxCR<RB8> in mode 2 with parity bit addition and mode 3.

FIFO cannot be enabled in mode 3. When parity bit is added in mode 1, the parity bit cannot be stored in FIFO. But a parity error is determined for each received data, and the result is stored in SCxCR<PERR>.

10.10.3.5. Wake-up Function

In mode 3, when SCxMOD0<WU> is set to "1" (the wake-up function control enabled), the wake-up operation of the slave controller can be used. INTRXx can be generated only when SCxCR<RB8> is "1".

10.10.3.6. Overrun Error

When FIFO is not enabled, an overrun error occurs when reception of the next data is completed before reading the reception buffer.

When an overrun error occurs, the content of the reception buffer is maintained. The content of SCxCR<RB8> is also maintained.

When FIFO is enabled, an overrun occurs when the reception of next data is completed before reading FIFO even though FIFO is FULL.

When an overrun error occurs, the contents of FIFO are maintained.

10.11. Transmission

10.11.1. Transmission Counter

The transmission counter is a 4-bit binary counter.

In SIO mode, it is increased by SIOCLK.

In UART mode, it is increased by 16 SIOCLK clocks. TXDCLK is generated every 16 SIOCLK clocks.

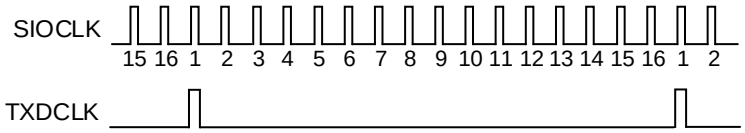


Figure 10.6 Generation of Transmission Clock in UART Mode

10.11.2. Transmission Control

10.11.2.1. For SIO Mode

When SCLKx pin output is used as the transfer clock, the data of the transmission shift register is output to TXDx pin one by one at the falling edge of SCLKx pin output.

When SCLKx pin input is used as the transfer clock, the data of the transmission shift register is output to TXDx pin one by one at the rising or falling edge of SCLKx pin input according to the SCxCR<SCLKS> setting.

10.11.2.2. For UART Mode

When transmission data is written to the transmission buffer, transmission starts at the rising edge of the next TXDCLK clock and generates the transmission shift clock.

10.11.3. Transmission Operation

10.11.3.1. Transmission Buffer Operation

When double buffer is disabled, transmission data is written to the transmission shift register, and a transmission interrupt (INTTXx) occurs when transmission is completed.

When double buffer is enabled or double buffer/FIFO are enabled, the data written to the transmission buffer is transferred to the transmission shift register. At the same time, INTTXx occurs and SCxMOD2<TBEMP> is set to "1". When the next data is written to the transmission buffer, SCxMOD2<TBEMP> is set to "0".

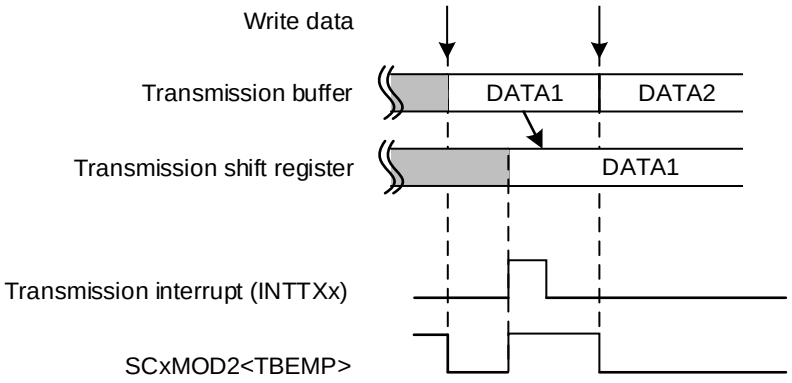


Figure 10.7 Transmission Buffer Operation

10.11.3.2. Operation of FIFO

When FIFO is enabled, the transmission buffer and FIFO can contain up to 5 bytes of transmission data. When transmission is enabled, the data stored in the transmission buffer is transferred to the transmission shift register and transmission starts. At this time, when transmission data is stored in FIFO, transmission data is transferred to the transmission buffer and SCxMOD2<TBEMP> flag is cleared to "0".

An interrupt is generated according to SCxTFC <TIL[1:0]> setting.

The setting and operation for half-duplex transmission of 4-byte data are shown below.

SCxMOD1<FDPX[1:0]>= "10"	: Set transfer mode to half duplex transmission
SCxFCNF<RFST><TFIE><RFIE>	: Enable automatic disabling when FIFO is empty.
<RXTXCNT><CNFG>= "10111"	The number of bytes of the used FIFO is the same as FILL level which generates an interrupt.
SCxTFC<TIL[1:0]>= "00"	: FILL level which generates interrupt is set to 0 bytes.
SCxTFC<TFCS><TFIS>= "11"	: Clears FIFO and sets interrupt generation conditions.
SCxFCNF<CNFG>= "1"	: FIFO is enabled.

After FIFO is enabled in the above setting, write 5 bytes of transmission data to the transmission buffer and FIFO, and set SCxMOD1<TXE> to "1" to start data transmission. INTTXx occurs when the last transmission data has been transferred to the transmission buffer. SCxMOD1<TXE> is automatically set to "0" when the transmission of the last transmission data is completed.

When automatic disabling after reaching FILL level is disabled in the above setting, transmission operation continues when transmission data is written.

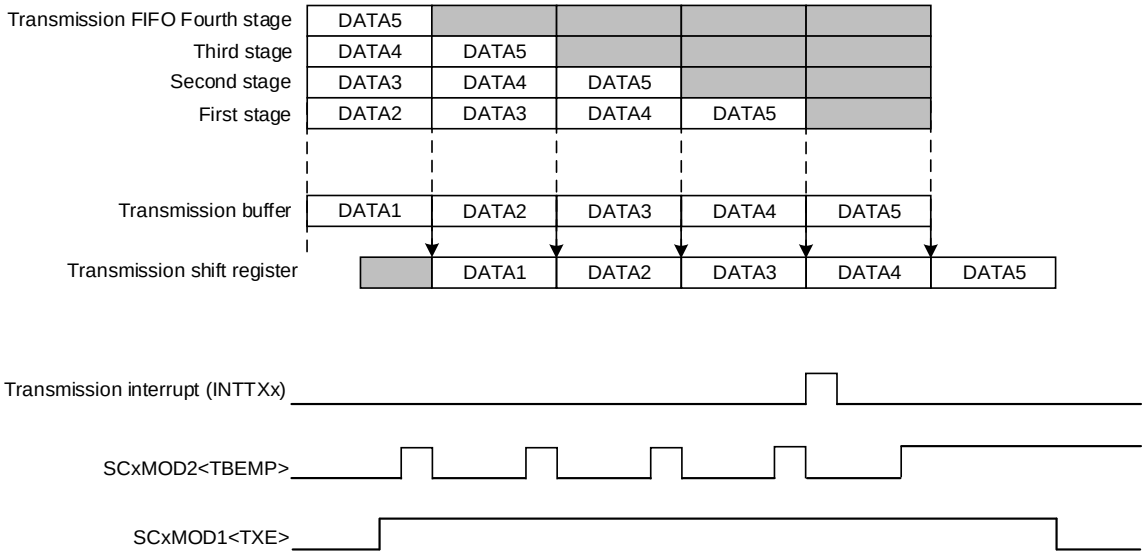


Figure 10.8 Transmission FIFO Operation

10.11.3.3. Transmission in SIO Mode and SCLKx Pin Output

The timing of stopping and restarting SCLKx pin output depends on the transmission buffer and FIFO enabling/disabling.

(1) When double buffer is disabled

When transmission of data in the transmission shift register is completed, SCLKx pin output is stopped. When the transmission data is written to the transmission buffer, SCLKx pin output restarts.

(2) When double buffer is enabled

When transmission of data in the transmission shift register and transmission buffer are completed, SCLKx pin output is stopped. When transmission data is written to the transmission buffer, data is transferred from the transmission buffer to the transmission shift register, and SCLKx pin output restarts.

(3) When FIFO is enabled

When transmission of data in the transmission shift register, transmission buffer, and FIFO are completed, SCLKx pin output is stopped. When the next data is written, the data is transferred from the transmission buffer to the transmission shift register and SCLKx pin output restarts.

Note that when SCxFCNF<RXTXCNT> is set to "1", SCLKx pin output is stopped and SCxMOD1<TXE> is set to "0". And the transmission operation is stopped.

10.11.3.4. Underrun Error

An underrun error occurs when the next data is not written to the transmission buffer before the next transmission starts after the transmission of the data in the transmission shift register is completed.

10.12. Handshake Function

The handshake function is the function which transmits 1 data by 1 data. This function avoids an overrun error.

The handshake function is used in UART mode.

The handshake function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTSx}}$ pin input becomes "High" level, transmission is stopped until the $\overline{\text{CTSx}}$ pin input becomes to "Low" level after completion of current transmission.

The next transmission data is written to the transmission buffer in INTTXx interrupt service routine to make SIO/UART to transmission wait state.

Note1: When the $\overline{\text{CTSx}}$ pin input becomes "High" level during transmission, transmission operation stops after completion of data transmission. (Figure 10.10 "a")

Note2: Transmission starts from the first TXDCLK clock after $\overline{\text{CTSx}}$ pin input becomes "Low" level. (Figure 10.10 "b")

Although there is no $\overline{\text{RTS}}$ pin, the handshake function can be easily constructed by assigning one of any port to the $\overline{\text{RTS}}$ pin. Set this port to "High" level in the reception interrupt service routine at the end of reception to request the transmission side to suspend transmission.

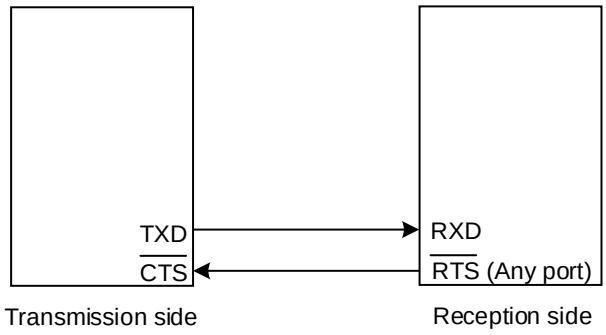


Figure 10.9 Handshake Function Connection

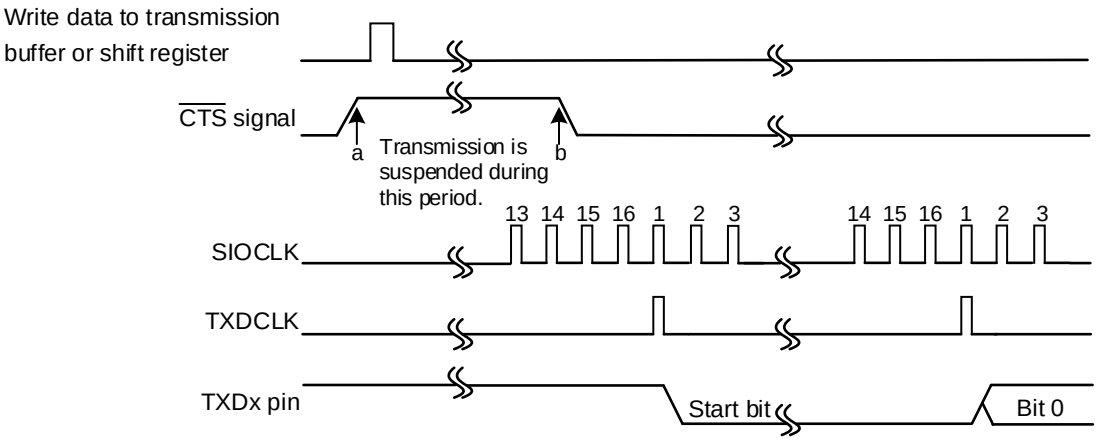


Figure 10.10 $\overline{\text{CTS}}$ Signal Timing

10.13. Interrupt/Error Generation Timing

10.13.1. Reception Interrupt (INTRXx)

Figure 10.11 shows the data flow for reception operation and the read path.Figure 10.11 Reception Operation Data Flow and Read Path

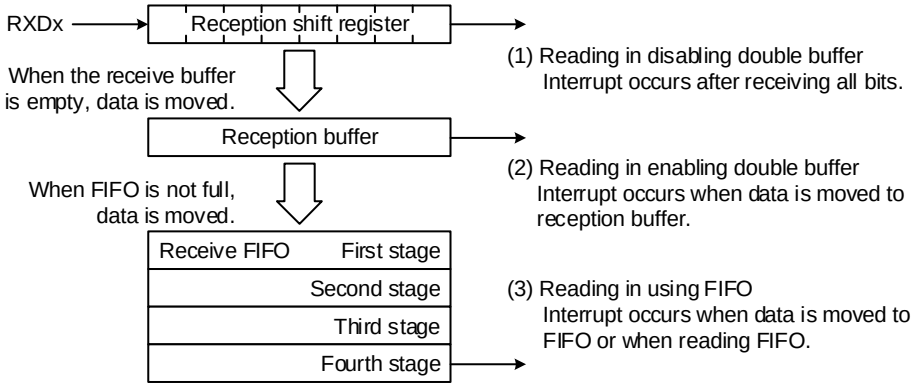


Figure 10.11 Reception Operation Data Flow and Read Path

10.13.1.1. Buffer Configuration and INTRXx Occurrence Timing

INTRXx occurs at the following timings depending on buffer configuration and the transfer mode:

Table 10.13 Buffer Configuration and INTRXx Generation Timing

Buffer Configuration	Transfer mode	
	UART mode	SIO mode
Double Buffer Disable	-	Immediately after the rising edge of the SCLKx output at the last-bit of the data when SCLKx pin output is used as transfer clock
Double Buffer Enable	Near the center of the first STOP bit	Immediately after the rising/falling edge of the SCLKx input at the last-bit of the data when SCLKx pin input is used as transfer clock (rising/falling edge according to SCxCR<SCLKS> setting.) Immediately after the rising edge of the last-bit SCLKx output when SCLKx pin output is used as transfer clock When data is transferred from the reception shift register to the reception buffer by reading the reception buffer

Note: No interrupt occurs when an overrun error occurs.

10.13.1.2. When FIFO is Enabled

When the FIFO is used, a receive interrupt occurs on depending on the timing described in Table 10.14 and the condition specified with SCxRFC<RFIS>.

Table 10.14 Conditions for INTRXx Generation when FIFO is Enabled

SCxRFC<RFIS>	Conditions of occurrence	Interrupt generation timing
0	When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	- When transfer a received data from receive buffer to receive FIFO
1	When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	- When read a receive data from receive FIFO - When transfer a received data from receive buffer to receive FIFO

10.13.2. Transmission Interrupt (INTTXx)

Figure 10.12 shows the data flow for transmission operation and the write path.

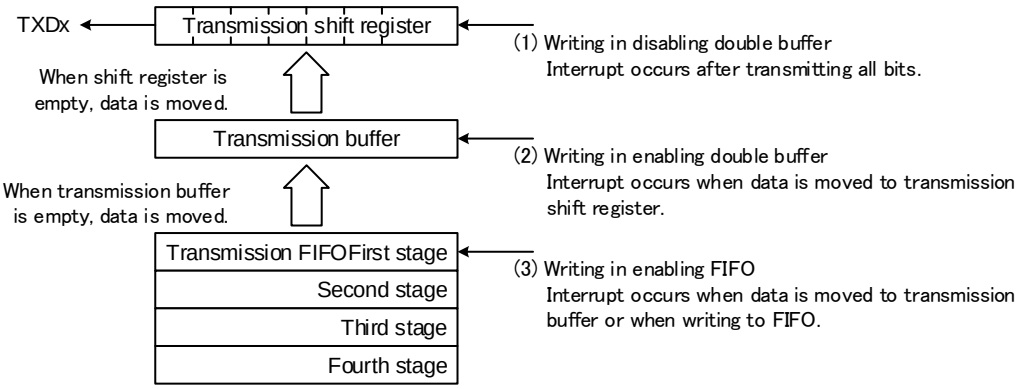


Figure 10.12 Transmission Operation Data Flow and Write Path

10.13.2.1. Buffer Configuration and INTTXx Generation Timing

INTTXx occurs at the following timings depending on the transfer mode and buffer configuration:

Table 10.15 Buffer Configuration and INTTXx Generation Timing

Buffer Configuration	UART mode	SIO mode
Double buffer disabled	Immediately before STOP bit transmission	Immediately after the rising or falling edge of the SCLKx pin input at the last-bit of the data when SCLKx pin input is used as transfer clock (rising or falling edge according to SCxCR<SCLKS> setting). Immediately after the falling edge of the last-bit SCLKx pin output when SCLKx pin output is used as transfer clock
Double buffer enabled	When data is transferred from the transmission buffer to the transmission shift register	

Note: With double buffer enabled, INTTXx also occurs when data is transferred from the transmission buffer to the transmission shift register by writing to transmission buffer.

10.13.2.2. When FIFO is Enabled

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 10.16 and the condition specified with SCxTFC<TFIS>.

Table 10.16 Conditions for INTTXx Generation when FIFO is Enabled

SCxTFC<TFIS>	Conditions of occurrence	Interrupt generation timing
0	When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>	- When transmitted data is transferred from transmit FIFO to transmit buffer
1	When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>	- When transmit data is write into transmit FIFO - When transmitted data is transferred from transmit FIFO to transmit buffer

10.13.3. Interrupt Generation Timing When Errors Occur

10.13.3.1. SIO Mode

Table 10.17 Timing of Error Interrupt Generation in SIO Mode

Error	Transfer clock	
	SCLKx pin output	SCLKx pin input
Overrun error	None	For reception, immediately after the rising/falling edge of the SCLKx pin input at the last-bit of the data (rising/falling edge according to SCxCR<SCLKS> setting).
Underrun error	None	For transmission, immediately after the rising/falling edge of SCLKx pin input of the first bit of the next data (rising/falling edge according to SCxCR<SCLKS> setting)

10.13.3.2. UART Mode

Table 10.18 Error Interrupt Generation Timing in UART Mode

Error	Mode		
	Mode 3	Mode 1, Mode 2	Mode 1, Mode 2
	Parity bit addition disabled	Parity bit addition disabled	Parity bit addition enabled
Framing error Overrun error	Near the center of STOP bit	Near the center of STOP bit	Near the center of STOP bit
Parity error	None	None	Near the center of the parity bit

10.14. Software Reset

Software reset is generated by writing "10" → "01" to SCxMOD2<SWRST[1:0]>. For the initialization range, refer to "10.3.9. SCxMOD2 (Mode Control Register 2)".

10.15. Explanation of Operation for Each Mode

10.15.1. Mode 0 (SIO Mode)

In this mode, the transfer clock shows below.

- SCLKx pin output is used as transfer clock.
- SCLKx pin input is used as transfer clock.

The following explains the reception, transmission, and transmission/reception operations when FIFO is disabled. For details of FIFO operation, refer to "10.7.3. FIFO".

10.15.1.1. Reception

(1) When SCLKx pin output is used as the transfer clock

When SCxMOD0<RXE> is set to "1", SCLKx pin output is started and the data of RXDx pin is captured.

- When double buffer is disabled (SCxMOD2<WBUF>= "0")

  Each time reception data is read, the transfer clock is output from SCLKx pin and the next data is stored in the reception shift register. INTRXx occurs when data reception is completed.

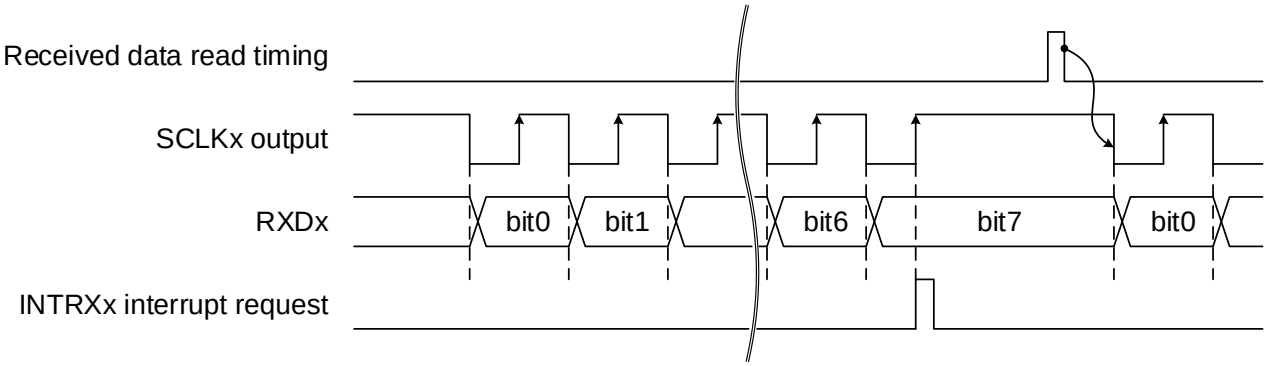
- When double buffer is enabled (SCxMOD2<WBUF>= "1")

  Because the data stored in the reception shift register is transferred to the reception buffer, the next data can be received continuously.

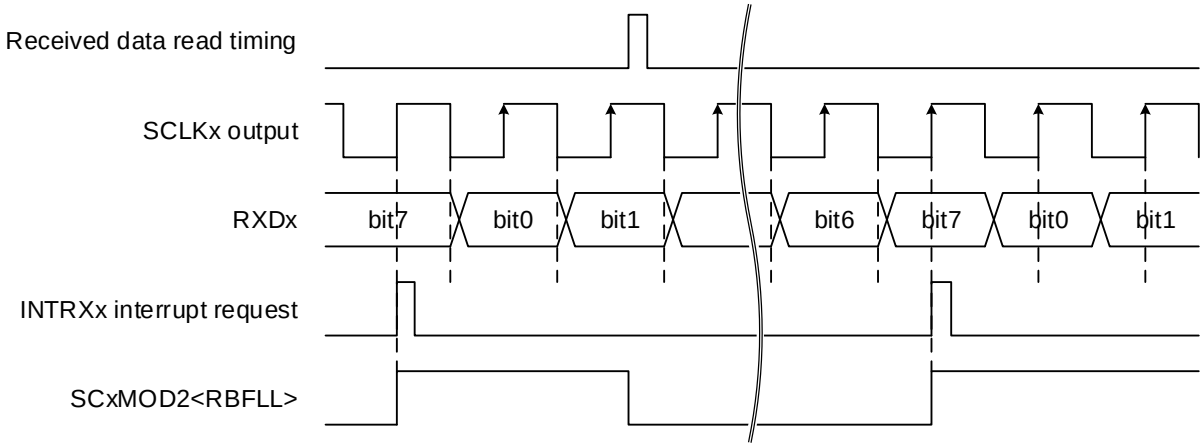
  When data is transferred from the reception shift register to the reception buffer, SCxMOD2<RBFLL> is set to "1" and INTRXx occurs.

  When there is data in the reception buffer and the data in the reception buffer is not read before completion of reception of the next data, INTRXx does not occur and SCLKx pin output is stopped.

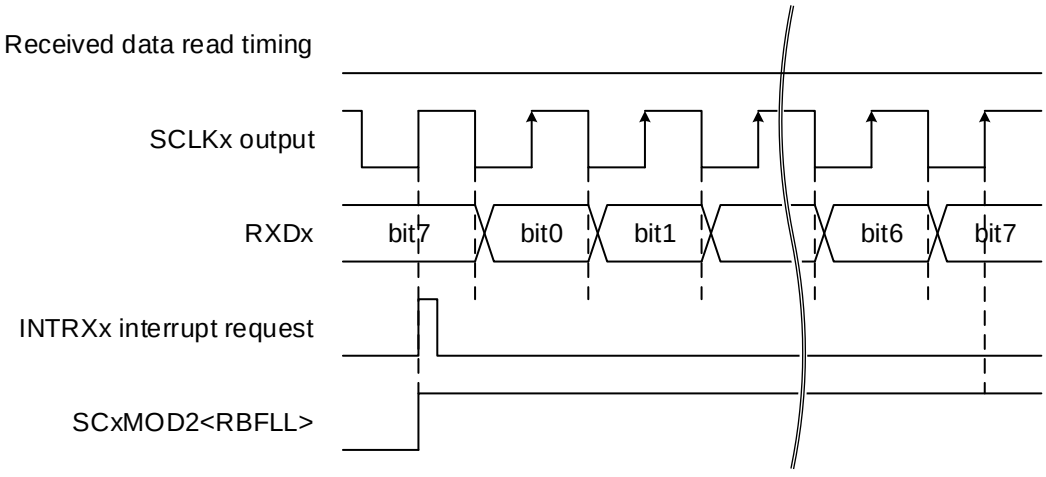
  In this situation, reading data from the reception buffer transfers the data in the reception shift register to the reception buffer and INTRXx occurs. Transfer clock is output from SCLKx pin and reception is restarted.



SCxMOD2<WBUF> = "0" (double buffer is disabled.)



SCxMOD2<WBUF> = "1" (double buffer is enabled and data is read from buffer.)



SCxMOD2<WBUF> = "1" (double buffer is enabled and data is read from buffer.)

Figure 10.13 SIO Mode Reception Operation (When SCLKx Pin Output is Used for Transfer Clock)

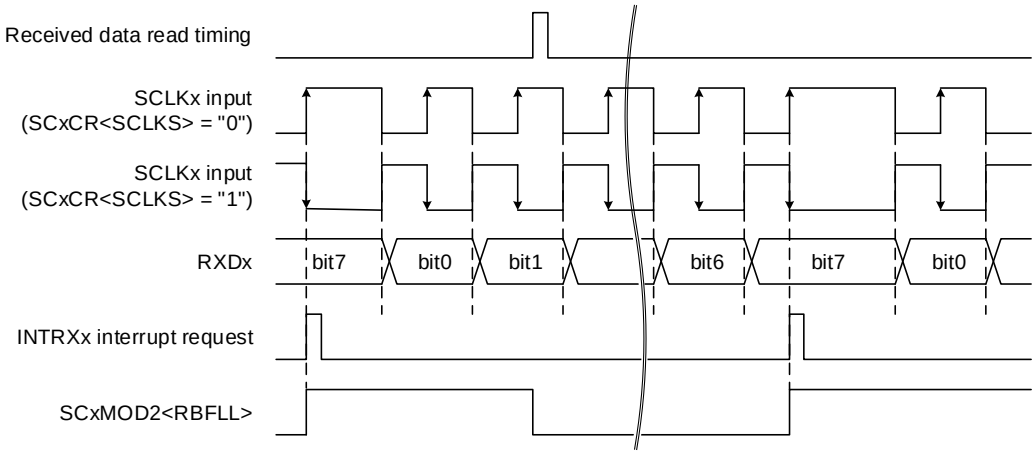
(2) When SCLKx pin input is used as the transfer clock

When SCxMOD0<RXE> is set to "1" and the transfer clock is input to SCLKx pin, the data of RXDx pin is captured.

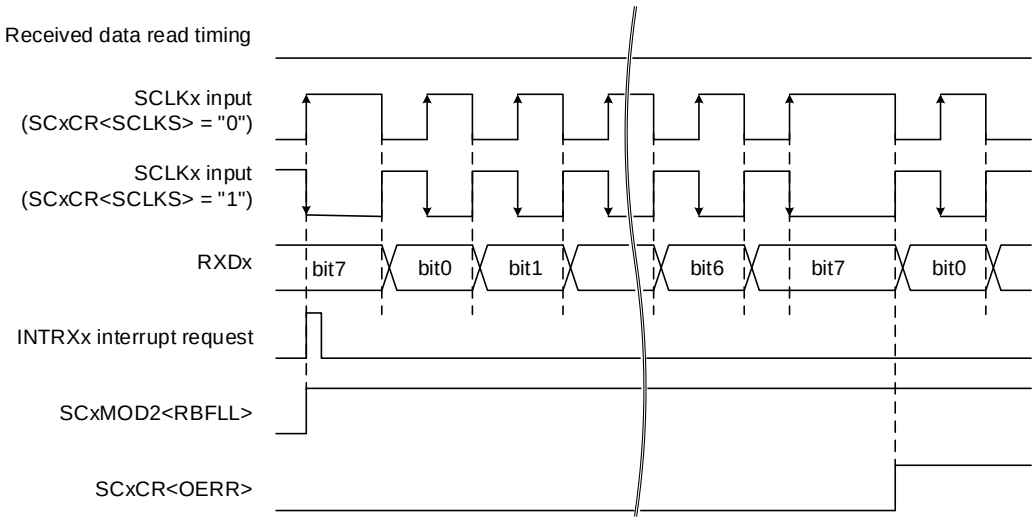
Reception double buffer is enabled whenever SCLKx pin input is used as the transfer clock. Therefore, the received data is transferred from the reception shift register to the reception buffer, and the next data can be received continuously.

INTRXx occurs each time received data is transferred to the reception buffer.

When there is data in the reception buffer and the data in the reception buffer is not read before the completion of reception of the next data, INTRXx does not occur and SCxCR<OERR> is set to "1".



When data is read from buffer



When data is not read from buffer

Figure 10.14 SIO Mode Reception Operation (When SCLKx Pin Input is Used as Transfer Clock)

10.15.1.2. Transmission

(1) When SCLKx pin output is used as the transfer clock

When SCxMOD1<TXE> is set to "1" after transmission data is written, the transfer clock is output from SCLKx pin and data is output from TXDx pin.

- When double buffer is disabled (SCxMOD2<WBUF> = "0")

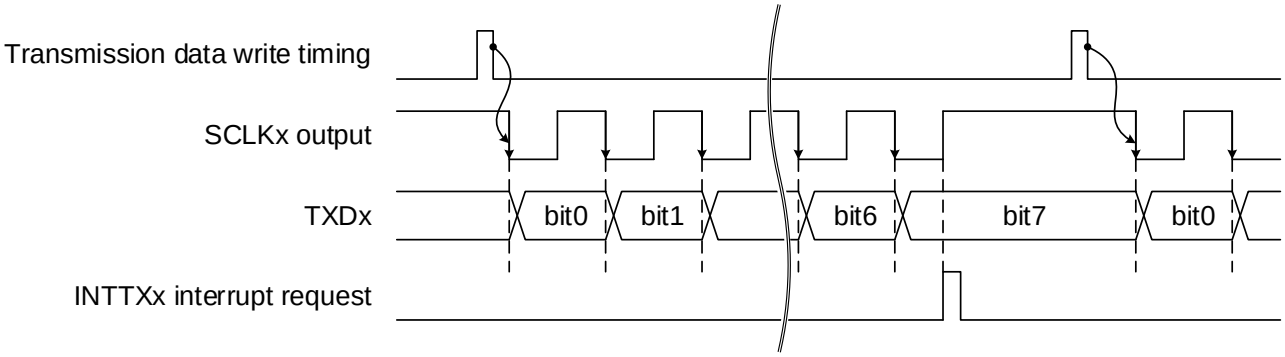
  Each time data is written to the transmission buffer, data is output from TXDx pin and transfer clock is output from SCLKx pin. INTTXx occurs when data transmission is completed.

- When double buffer is enabled (SCxMOD2<WBUF> = "1")

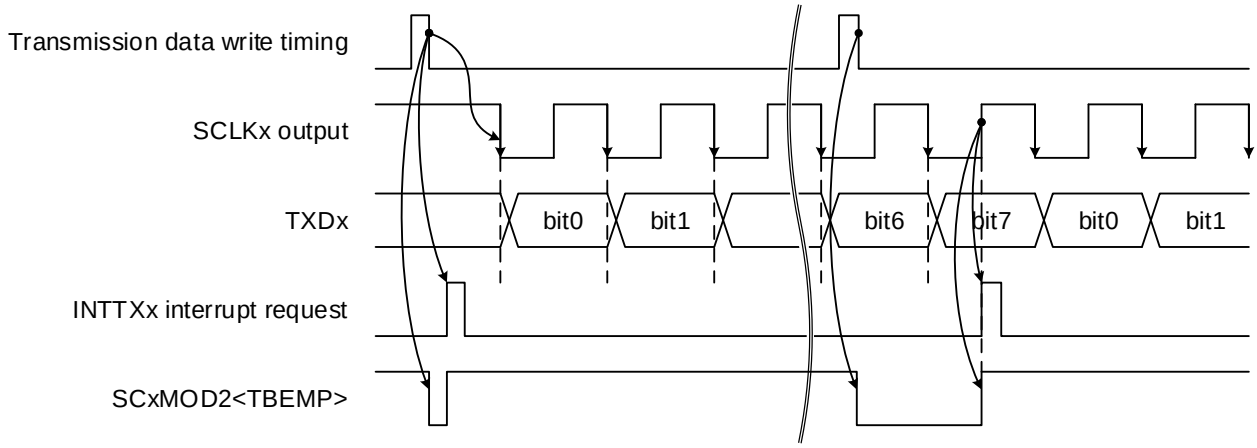
  Data is transferred from the transmission buffer to the transmission shift register when data is written to the transmission buffer while transmission is stopped or when data transmission in the transmission shift register is complete. At this time, SCxMOD2<TBEMP> is set to "1" and INTTXx occurs.

  When no data is written to the transmission buffer at the end of data transmission in the transmission shift register, INTTXx does not occur and SCLKx pin output is stopped.

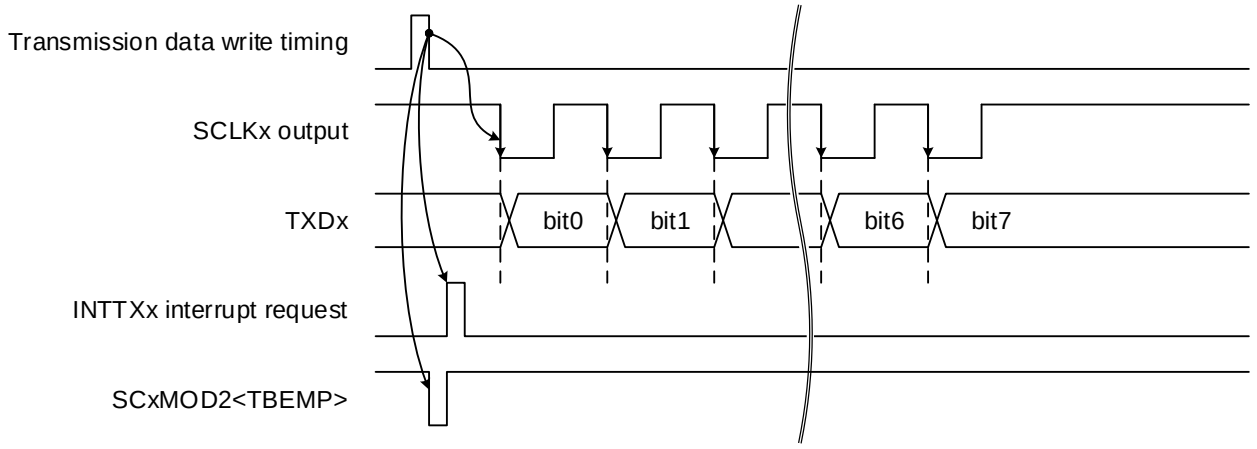
  In this situation, when data is written to the transmission buffer, the data in the transmission buffer is transferred to the transmission shift register and INTTXx occurs. The transfer clock is output from SCLKx pin and transmission restarts.



SCxMOD2<WBUF> = "0" (double buffer is disabled.)



SCxMOD2<WBUF> = "1" (double buffer is enabled and data is in buffer.)

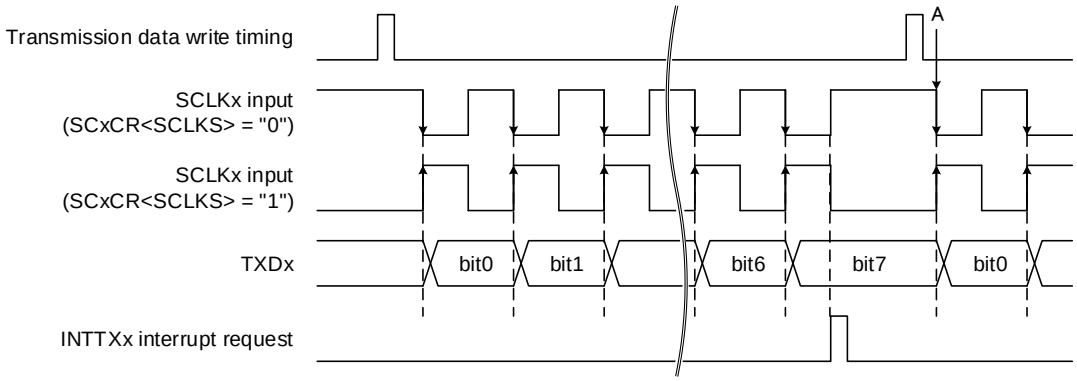


SCxMOD2<WBUF> = "1" (double buffer is enabled and data is not in buffer.)

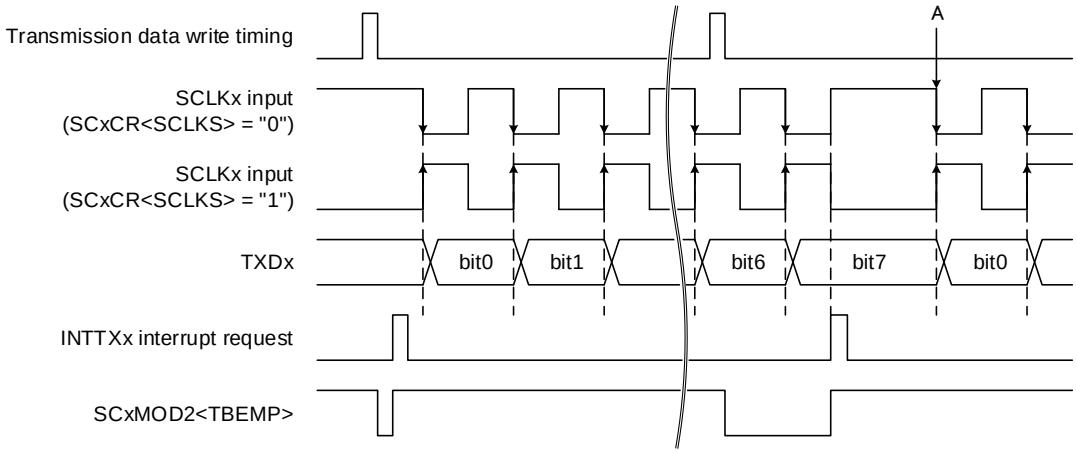
Figure 10.15 SIO Mode Transmission Operation
(When SCLKx Pin Output is Used for Transfer Clock)

(2) When SCLKx pin input is used as the transfer clock

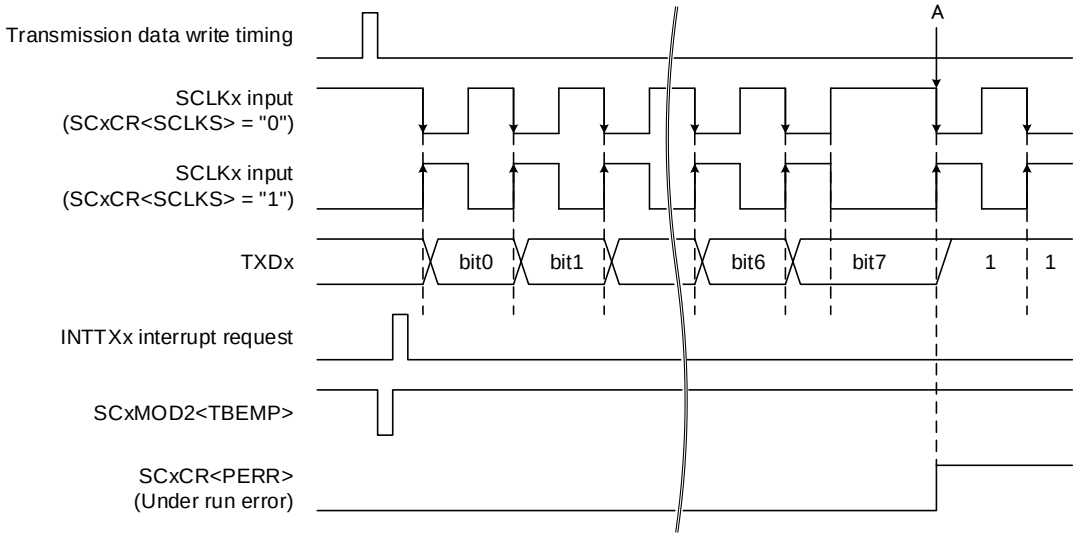
- When double buffer is disabled (SCxMOD2<WBUF> = "0")
  When a transfer clock is input to SCLKx pin when data is written in the transmission buffer, data is output from TXDx pin. INTTXx occurs when the data is transmitted. Write the next data by point A shown in Figure 10.16.
- When double buffer is enabled (SCxMOD2<WBUF> = "1")
  Data is transferred from the transmission buffer to the transmission shift register when data is written to the transmission buffer while transmission is stopped or when data transmission in the transmission shift register is complete. At this time, SCxMOD2<TBEMP> is set to "1" and INTTXx occurs. Write the next data by point A shown in Figure 10.16.
  If transfer clock is input to SCLKx pin when data transmission of the transmission shift register is completed and the next data is not written to the transmission buffer, the transmission operation is continued, but an underrun error occurs. At this time, SCxCR<PERR> is set to "1", and dummy data "0xFF" for 8 bits is output from TXDx pin.



SCxMOD2<WBUF> = "0" (double buffer is disabled.)



SCxMOD2<WBUF> = "1" (double buffer is enabled and data is in buffer.)



SCxMOD2<WBUF> = "1" (double buffer is enabled and data is not in buffer.)

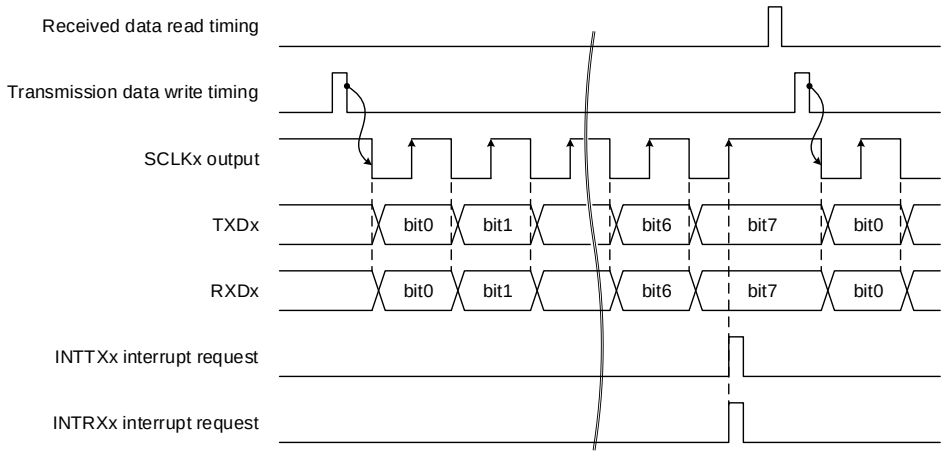
Figure 10.16 SIO Mode Transmission Operation
(When SCLKx Pin Input is Used for Transfer Clock)

10.15.1.3. Transmission/Reception (Full Duplex)

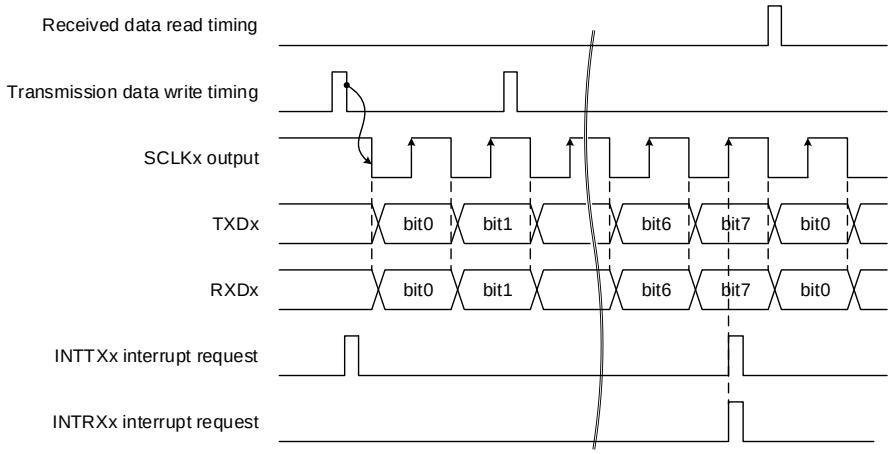
(1) When SCLKx pin input is used as the transfer clock

- When double buffer is disabled (SCxMOD2<WBUF> = "0")
  When data is written to the transmission buffer, SCLKx pin output starts.
  The reception data is stored in the reception shift register via RXDx pin and INTRXx occurs. In addition, INTTXx occurs when data written to the transmission buffer is output from TXDx pin and data transmission is completed. At this time, SCLKx pin output is stopped.
  When the reception buffer is read and the next data is written to the transmission buffer, SCLKx pin output is restarted and the next data is received and transmitted.
  The order in which the reception buffer is read and the transmission buffer is written is arbitrary. When both conditions are satisfied, SCLKx pin output is restarted.

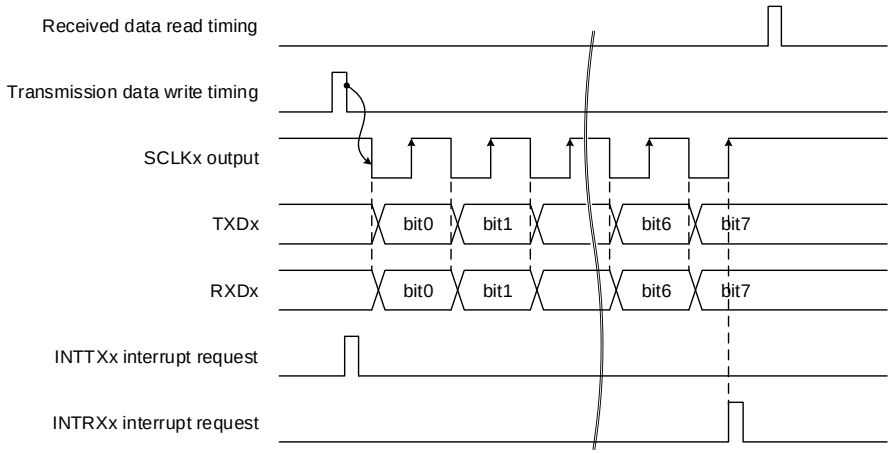
- When double buffer is enabled (SCxMOD2<WBUF> = "1")
  When data is written to the transmission buffer, SCLKx pin output starts.
  The reception data is stored in the reception shift register via RXDx pin and INTRXx occurs. In addition, INTTXx occurs when data written to the transmission buffer is output from TXDx pin, and data transmission is completed, and the next data is written to the transmission buffer.
  In such cases, SCLKx pin continues to be output, the next data is received and transmitted.
  INTTXx does not occurs when the data is not written to transmission buffer. SCLKx pin output is also stopped.
  When the reception buffer is read and the next data is written to the transmission buffer, SCLKx pin output is restarted.



SCxMOD2<WBUF> = "0" (double buffer is disabled.)



SCxMOD2<WBUF> = "0" (double buffer is enabled.)

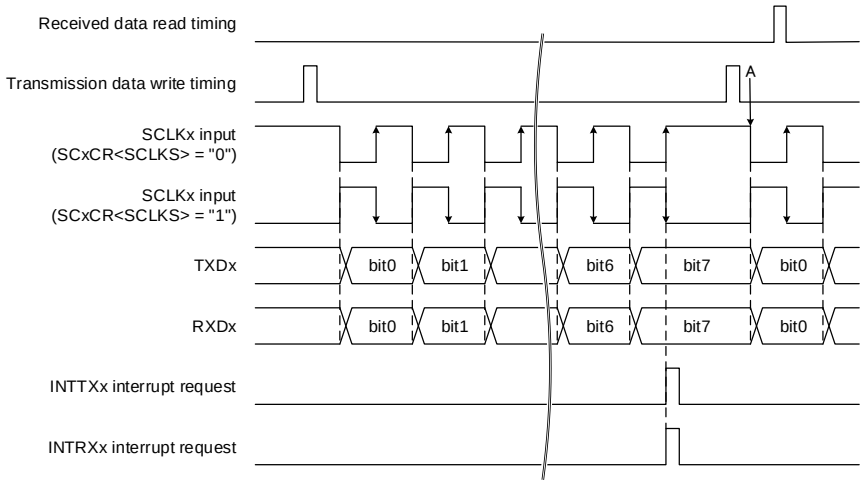


SCxMOD2<WBUF> = "0" (double buffer is enabled.)

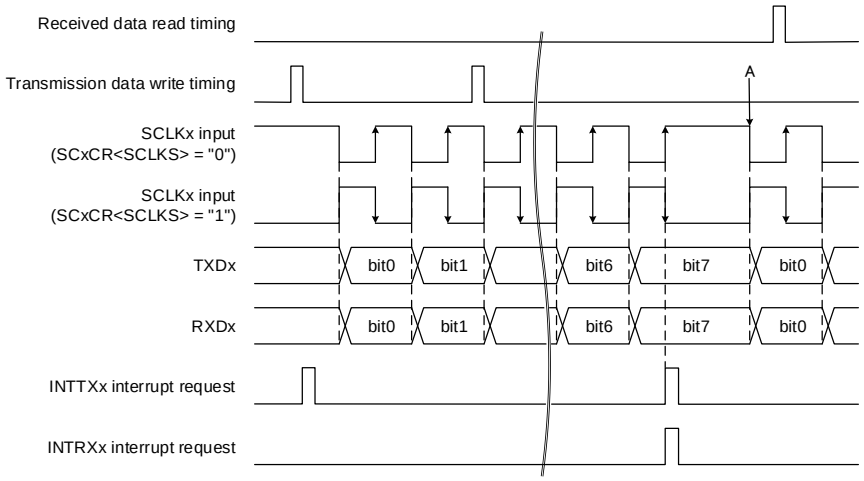
Figure 10.17 SIO Mode Transmission/Reception Operation (When SCLKx Pin Output is Used for Transfer Clock)

(2) When SCLKx pin is used as the transfer clock

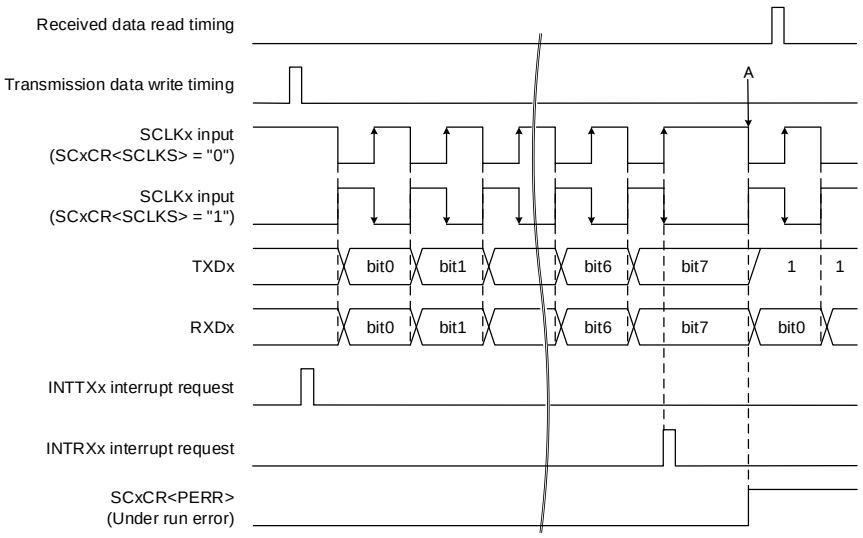
- When double buffer is disabled (SCxMOD2<WBUF>= "0")
  Double buffer is enabled on reception by regardless of the setting of SCxMOD2<WBUF>.
  When the transfer clock is input to SCLKx pin when data is written in the transmission buffer, the transmission data is output from TXDx pin. Reception data is also stored in the reception shift register through RXDx pin.
  An INTTXx occurs when outputting data is completed. When reception is completed, data is stored from the reception shift register to the reception buffer, and an INTRXx is generated.
  Write the next data by point A shown in Figure 10.18. Read the reception data before the next data reception is completed.
- When double buffer is enabled (SCxMOD2<WBUF>= "1")
  If the transfer clock is input to SCLKx pin when data is written in the transmission buffer, data is transferred from the transmission buffer to the transmission shift register, and an INTTXx is generated. Transmission data is output from TXDx pin in synchronization with the transfer clock input to SCLKx pin. Reception data is also stored in the reception shift register through RXDx pin.
  An INTTXx occurs when the transmission shift register has finished transmission data. At the end of reception, data is transferred from the reception shift register to the reception buffer and an INTRXx occurs. Write the next transmission data by point A shown in Figure 10.18. Read the reception data before the reception of the next data is completed. When the transfer clock of the next data is subsequently input to SCLKx pin, the data of the transmission buffer is transferred to the transmission shift register, and an INTTXx is generated. Reception data is also stored in the reception shift register through RXDx pin.
  If the data in the reception buffer has not been read before the reception of the last bit of this data, an overrun error occurs. If the next transmission data has not been written to the transmission buffer before point A shown in Figure 10.18, an underrun error occurs.



SCxMOD2<WBUF> = "0" (double buffer is disabled.)



SCxMOD2<WBUF> = "1" (double buffer is enabled with no error.)



SCxMOD2<WBUF> = "1" (double buffer is enabled with error.)

Figure 10.18 SIO Mode Transmission/Reception Operation
(When SCLKx Pin Input is Used for Transfer Clock)

10.15.2. Mode 1 (7-bit UART Mode)

When SCxMOD0<SM[1:0]> is set to "01", UART operates in mode 1.

Parity bit can be added in this mode. SCxCR <PE> enable/disable of parity bit addition is controlled. Uses SCxCR<EVEN> to set the parity bit to Odd or Even.

For transmission only, select STOP bit length in SCxMOD2<SBLEN>. For reception, STOP bit length is 1 bit only.

The following shows a setting example when transmitting data in the following format.

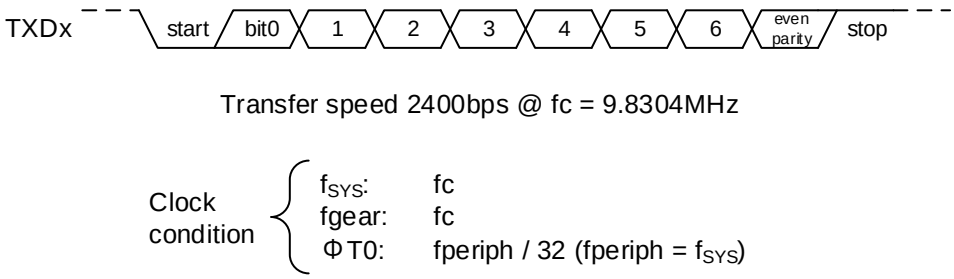


Figure 10.19 Mode 1 (7-bit UART Mode)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	×	0	0	0	0	1	0	1	Selects 7-bit UART. mode
SCxCR	←	×	1	1	×	×	×	0	0	Enables parity bit addition, set to even
SCxBRCR	←	0	0	1	0	0	1	0	0	Sets the baud rate to 2400 bps (ΦT16, N = 4)
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmission data

Note: ×: Don't care, *: Any parameter can be set.

10.15.3. Mode 2 (8-bit UART Mode)

When SCxMOD0<SM[1:0]> is set to "10", UART operates in mode 2.

Parity bit can be added in this mode. SCxCR<PE> is controlled enable/disable of parity bit addition. Use SCxCR<EVEN> to set the parity bit to Odd or Even.

For transmission only, select STOP bit length in SCxMOD2<SBLEN>. For reception, STOP bit length is 1 bit only.

The following shows a setting example when receiving data in the following format.

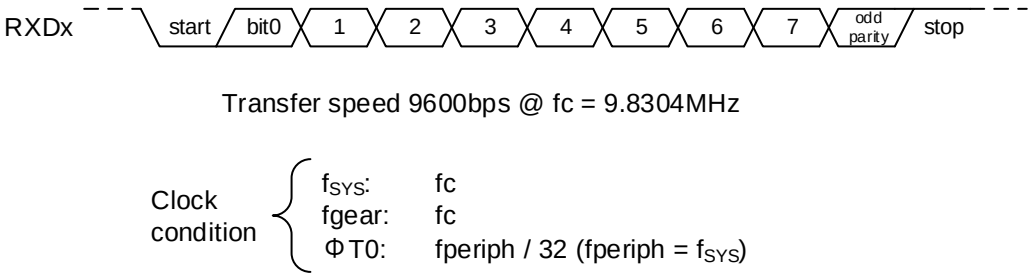


Figure 10.20 Mode 2 (8-bit UART Mode)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	×	0	0	0	1	0	0	1	Selects 8-bit UART mode.
SCxCR	←	×	1	1	×	×	×	0	0	Enables parity bit addition, set to Odd.
SCxBRCR	←	0	0	0	1	0	1	0	0	Sets the baud rate to 9600 bps (ΦT4, N = 4)
SCxMOD0	←	×	-	1	-	-	-	-	-	Enables reception

Note: ×: Don't care, -: Cannot be changed

10.15.4. Mode3 (9-bit UART mode)

When SCxMOD0<SM[1:0]>) is set to "11", UART operates in mode 3.

In this mode, the parity bit cannot be added, and SCxCR<PE> must be set to "0".

The MSB of the data is written to SCxMOD0<TB8> for transmission and is read from SCxCR<RB8> for reception. Writing data and reading data is firstly done from the MSB of the data .

For transmission only, select STOP bit length in SCxMOD2<SBLEN>. For reception, STOP bit length is 1 bit only.

In mode 3, the wake-up function of the slave controller can be used.

10.15.4.1. Wake-up Function

(1) Serial link with wake-up function

In mode 3, the wake-up function of the slave controller can be used. To enable the wake-up function, set SCxMOD0<WU> to "1".

When the wake-up function is enabled, INTRXx occurs only when the received data in SCxCR<RB8> is "1".

Note: Be sure to set PxOD register for TXDx pin of the slave controller to open drain output mode.

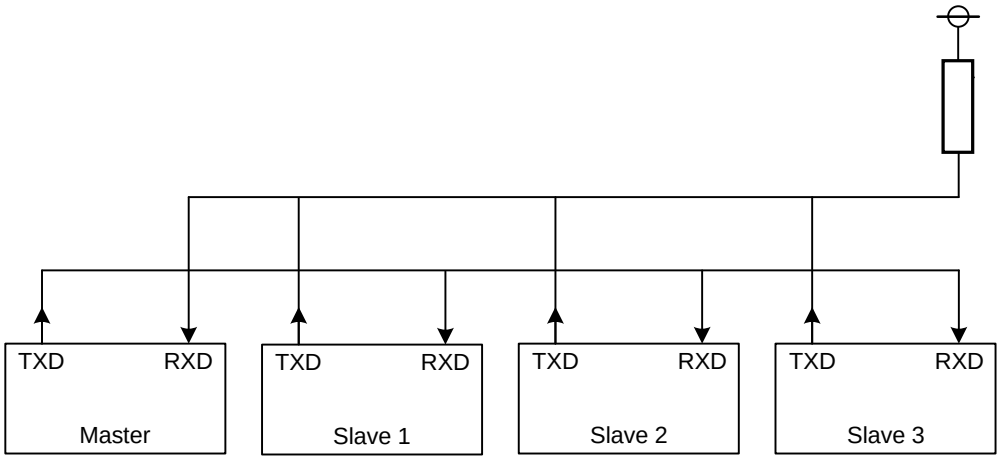


Figure 10.21 Serial Link with Wake-up Function

(2) Protocol

(a) Set the master and slave controllers to mode 3.

(b) Set SCxMOD0<WU> of each slave controller the respective to "1".
Set SCxMOD0<RXE> to "1" to enable reception.
The master controller transmits one frame containing the select code (8 bits) of the slave controller.
At this time, set SCxMOD0<TB8> of the master controller to "1".

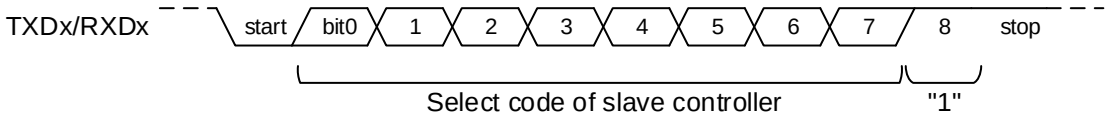


Figure 10.22 1st Frame Format for Wake-up Function

(c) Each slave controller receives the above frame and sets SCxMOD0<WU> to "0" when the received select code matches its own select code.

(d) The master controller transmits data to the specified slave controller (a controller whose SCxMOD0 <WU> is set to "0").
At this time, set SCxMOD0<TB8> of the master controller to "0".

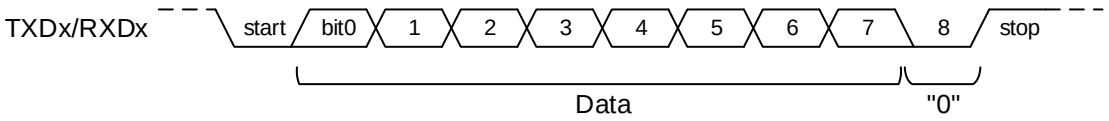


Figure 10.23 2nd Frame Format for Wake-up Function

(e) For slave controllers that did not match their own select code, INTRXx does not occurs because SCxMOD0<WU> remains "1" and the MSB of the received data is "0".
For a slave controller which matches its own select code, INTRXx occurs because SCxMOD0<WU> is set to "0". The slave controller can also transmission data to the master controller in INTRXx interrupt service routine and notify the master controller that reception is complete with the transmitted data.

11. 12-Bit Analog-to-Digital Converter (ADC)

The TMPM370FYDFG/TMPM370FYFG contain two 12-bit successive-approximation analog-to-digital converter (ADC). The ADC supports a vector control for motors in cooperation with the Vector Engine for motor control and PMD circuit.

The ADC unit A has 15 analog inputs. 6 inputs are able to use for measurement of motor 0. 3 inputs out of 6 inputs are connected to the output of op-amps/comparators in the MCU. Therefore, 12 inputs can use for external input.

The ADC unit B has 17 analog inputs. 6 inputs are able to use for measurement of motor 0. And 2 inputs are able to use for measurement of motor 1. 4 inputs out of 8 inputs are connected to the output of op-amps/comparators in the MCU. Therefore, 13 inputs can use for external input.

22 analog input pins (AINA0 to AINA8, AINA9/AINB0, AINA10/AINB1, AINA11/AINB2, AINB3 to AINB12) can also be used as input/output ports.

11.1. Functions and features

(1) Start AD conversion for any analog input synchronously with receiving trigger signal from PMD or TMRB.

(2) Start AD conversion for any analog input in the Software Trigger Program and the Constant Conversion Program.

(3) The ADC has twelve registers for AD conversion result

(4) The ADC generates interrupt signal at the end of the program which was started by trigger.

(5) The ADC generates interrupt signal at the end of the program which was started by the Software Trigger and/or the Constant Conversion.

(6) The ADC has the AD conversion monitoring function. When this function is enabled, an interrupt is generated when matching specified compare condition.

11.2. Block Diagram

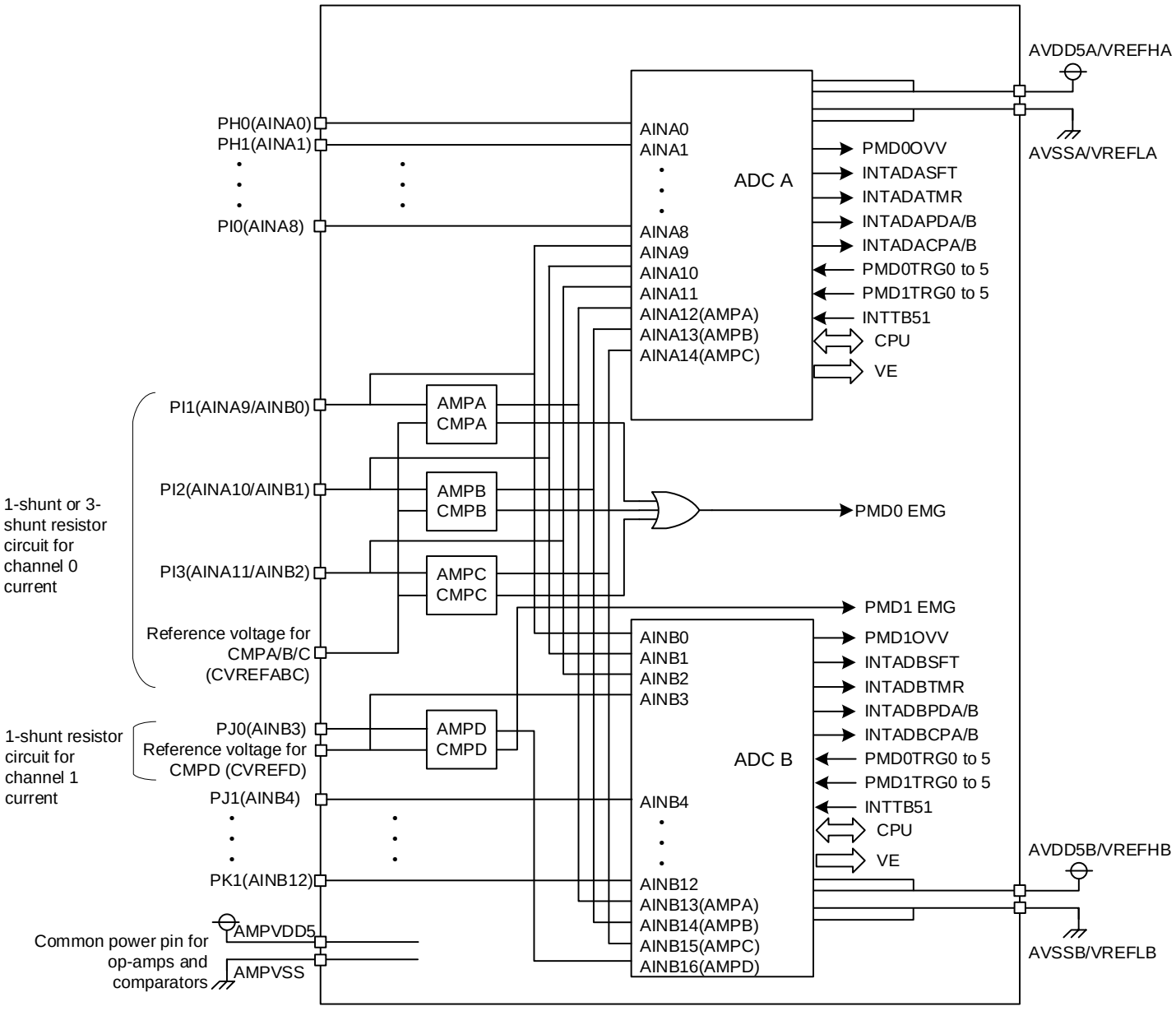


Figure 11.1 AD converter Block Diagram

11.3. List of Registers

Register Name		Address (Base+)
Clock Setting Register	ADxCLK	0x0000
Mode Setting Register 0	ADxMOD0	0x0004
Mode Setting Register 1	ADxMOD1	0x0008
Mode Setting Register 2	ADxMOD2	0x000C
Monitoring Setting Register 0	ADxCMPCR0	0x0010
Monitoring Setting Register 1	ADxCMPCR1	0x0014
Conversion Result Compare Register 0	ADxCMP0	0x0018
Conversion Result Compare Register 1	ADxCMP1	0x001C
Conversion Result Register 0	ADxREG0	0x0020
Conversion Result Register 1	ADxREG1	0x0024
Conversion Result Register 2	ADxREG2	0x0028
Conversion Result Register 3	ADxREG3	0x002C
Conversion Result Register 4	ADxREG4	0x0030
Conversion Result Register 5	ADxREG5	0x0034
Conversion Result Register 6	ADxREG6	0x0038
Conversion Result Register 7	ADxREG7	0x003C
Conversion Result Register 8	ADxREG8	0x0040
Conversion Result Register 9	ADxREG9	0x0044
Conversion Result Register 10	ADxREG10	0x0048
Conversion Result Register 11	ADxREG11	0x004C
PMD Trigger Program Number Select Register 0	ADxPSEL0	0x0050
PMD Trigger Program Number Select Register 1	ADxPSEL1	0x0054
PMD Trigger Program Number Select Register 2	ADxPSEL2	0x0058
PMD Trigger Program Number Select Register 3	ADxPSEL3	0x005C
PMD Trigger Program Number Select Register 4	ADxPSEL4	0x0060
PMD Trigger Program Number Select Register 5	ADxPSEL5	0x0064
PMD Trigger Program Number Select Register 6	ADxPSEL6	0x0068
PMD Trigger Program Number Select Register 7	ADxPSEL7	0x006C
PMD Trigger Program Number Select Register 8	ADxPSEL8	0x0070
PMD Trigger Program Number Select Register 9	ADxPSEL9	0x0074
PMD Trigger Program Number Select Register 10	ADxPSEL10	0x0078
PMD Trigger Program Number Select Register 11	ADxPSEL11	0x007C
PMD Trigger Interrupt Select Register 0	ADxPINTS0	0x0080
PMD Trigger Interrupt Select Register 1	ADxPINTS1	0x0084
PMD Trigger Interrupt Select Register 2	ADxPINTS2	0x0088

Register Name		Address (Base+)
PMD Trigger Interrupt Select Register 3	ADxPINTS3	0x008C
PMD Trigger Interrupt Select Register 4	ADxPINTS4	0x0090
PMD Trigger Interrupt Select Register 5	ADxPINTS5	0x0094
PMD Trigger Program Select Register 0	ADxPSET0	0x0098
PMD Trigger Program Select Register 1	ADxPSET1	0x009C
PMD Trigger Program Select Register 2	ADxPSET2	0x00A0
PMD Trigger Program Select Register 3	ADxPSET3	0x00A4
PMD Trigger Program Select Register 4	ADxPSET4	0x00A8
PMD Trigger Program Select Register 5	ADxPSET5	0x00AC
Timer Trigger Program Registers 0 to 3	ADxTSET03	0x00B0
Timer Trigger Program Registers 4 to 7	ADxTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADxTSET811	0x00B8
Software Trigger Program Registers 0 to 3	ADxSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADxSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADxSSET811	0x00C4
Constant Conversion Program Registers 0 to 3	ADxASET03	0x00B8
Constant Conversion Program Registers 4 to 7	ADxASET47	0x00BC
Constant Conversion Program Registers 8 to 11	ADxASET811	0x00D0
Mode Setting Register 3	ADxMOD3	0x00D4

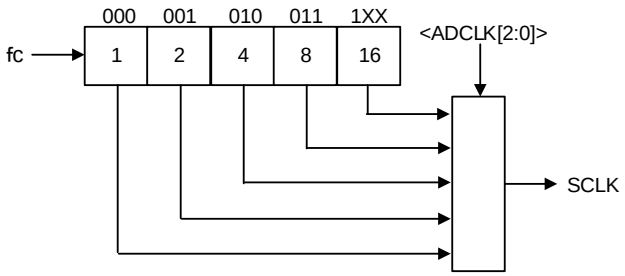
11.4. Register Descriptions

AD conversion is performed at the clock frequency selected in the Clock Setting Register.

11.4.1. ADxCLK (Clock Setting Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	TSH				ADCLK		
After reset	0	1	0	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31:7	-	R	Read as "0".
6:3	TSH[3:0]	R/W	Write as "1001".
2:0	ADCLK[2:0]	R/W	Select AD conversion clock (SCLK) 000: fc (Note1) 001: fc/2 010: fc/4 011: fc/8 1xx: fc/16



Note1: Frequency of SCLK can be used up to 40MHz. Do not set <ADCLK[2:0]> to "000" when fc is more than 40MHz.

Note2: AD conversion is performed at the clock selected in above register. The conversion clock must be selected to ensure the guaranteed accuracy.

Note3: The conversion clock setting must not be changed while AD conversion is in progress.

11.4.2. ADxMOD0 (Mode Setting Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	DACON	ADSS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	DACON	R/W	DAC control 0: OFF 1: ON Setting <DACON> to "1", when using the ADC.
0	ADSS	W	Start software triggered conversion 0: Don't care 1: Start conversion When ADxMOD1<ADEN> is set to "1" for enabling conversion, setting <ADSS> to "1" starts AD conversion. And receiving trigger signal from PMD or TMRB interrupt also starts AD conversion. For details of setting of PMD trigger timing and TMRB interrupt, refer to the section of PMD and TMRB.

11.4.3. ADxMOD1 (Mode Setting Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADEN	-	-	-	-	-	-	ADAS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	ADEN	R/W	AD conversion control 0: Disable 1: Enable Setting <ADEN> to "1" enables AD conversion. When setting <ADEN> to "1", setting <ADAS> to "1" starts AD conversion.
6:1	-	R	Read as "0".
0	ADAS	W	Constant conversion control 0: Disable 1: Enable

11.4.4. ADxMOD2 (Mode Setting Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	ADSFN	ADBFN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	ADSFN	R	Software conversion flag 0: Software conversion completed 1: Software conversion in progress
0	ADBFN	R	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress The <ADBFN> is an AD conversion busy flag. When AD conversion is started, <ADBFN> is set to "1". When finished AD conversion, <ADBFN> is cleared to "0".

11.4.5. ADxCMPCR0 (Monitoring Setting Register 0)

After determining the result, the interrupt signal (INTADxCPn) is generated.

(n = A, B, A: Monitor 0, B: Monitor 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:12	-	R	Read as "0".
11:8	CMPCNT0[3:0]	R/W	Number of comparisons for determining the result 0: Every comparison is valid. 1: Two comparisons or more is valid. . . 15: 16 comparisons or more is valid.
7	CMP0EN	R/W	Monitoring function control 0: Disable 1: Enable
6:5	-	R	Read as "0".
4	ADBIG0	R/W	Comparison condition 0: Larger than compare register 1: Smaller than compare register
3:0	REGS0[3:0]	R/W	AD conversion result register to be compared 0000: ADxREG0 0100: ADxREG4 1000: ADxREG8 0001: ADxREG1 0101: ADxREG5 1001: ADxREG9 0010: ADxREG2 0110: ADxREG6 1010: ADxREG10 0011: ADxREG3 0111: ADxREG7 1011: ADxREG11

11.4.6. ADxCMPCR1 (Monitoring Setting Register 1)

After determining the result, the interrupt signal (INTADxCPn) is generated.

(n = A, B, A: Monitor 0, B: Monitor 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:12	-	R	Read as "0".
11:8	CMPCNT1[3:0]	R/W	Comparison number for determining the result 0: Every comparison is valid. 1: Two comparisons or more is valid. . . 15: 16 comparisons or more is valid.
7	CMP1EN	R/W	Monitoring function control 0: Disable 1: Enable
6:5	-	R	Read as "0".
4	ADBIG1	R/W	Comparison condition 0: Larger than compare register 1: Smaller than compare register
3:0	REGS1[3:0]	R/W	AD conversion result register to be compared 0000: ADxREG0 0100: ADxREG4 1000: ADxREG8 0001: ADxREG1 0101: ADxREG5 1001: ADxREG9 0010: ADxREG2 0110: ADxREG6 1010: ADxREG10 0011: ADxREG3 0111: ADxREG7 1011: ADxREG11

11.4.7. ADxCMP0 (Conversion Result Compare Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	AD0CMP0							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	AD0CMP0				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	AD0CMP0[11:0]	R/W	The value to be compared with an AD conversion result Specify the value to be compared with an AD conversion result.
3:0	-	R	Read as "0".

11.4.8. ADxCMP1 (Conversion Result Compare Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	AD0CMP1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	AD0CMP1				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	AD0CMP1[11:0]	R/W	The value to be compared with an AD conversion result Specify the value to be compared with an AD conversion result.
3:0	-	R	Read as "0".

11.4.9. ADxREG0 (Conversion Result Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR0							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR0				-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR0[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR0	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG0 is read and is cleared to "0" when the low-order byte of ADxREG0 is read.
0	ADR0RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG0 register and is cleared to "0" when the low-order byte of ADxREG0 is read.

11.4.10. ADxREG1 (Conversion Result Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR1				-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR1[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR1	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG1 is read and is cleared to "0" when the low-order byte of ADxREG1 is read.
0	ADR1RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG1 register and is cleared to "0" when the low-order byte of ADxREG1 is read.

11.4.11. ADxREG2 (Conversion Result Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR2							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR2				-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR2[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR2	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG2 is read and is cleared to "0" when the low-order byte of ADxREG2 is read.
0	ADR2RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG2 register and is cleared to "0" when the low-order byte of ADxREG2 is read.

11.4.12. ADxREG3 (Conversion Result Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR3							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR3				-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR3[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR3	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG3 is read and is cleared to "0" when the low-order byte of ADxREG3 is read.
0	ADR3RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG3 register and is cleared to "0" when the low-order byte of ADxREG3 is read.

11.4.13. ADxREG4 (Conversion Result Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR4							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR4				-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR4[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR4	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG4 is read and is cleared to "0" when the low-order byte of ADxREG4 is read.
0	ADR4RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG4 register and is cleared to "0" when the low-order byte of ADxREG4 is read.

11.4.14. ADxREG5 (Conversion Result Register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR5							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR5				-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR5[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR5	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG5 is read and is cleared to "0" when the low-order byte of ADxREG5 is read.
0	ADR5RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG5 register and is cleared to "0" when the low-order byte of ADxREG5 is read.

11.4.15. ADxREG6 (Conversion Result Register 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR6							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR6				-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR6[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR6	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG6 is read and is cleared to "0" when the low-order byte of ADxREG6 is read.
0	ADR6RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG6 register and is cleared to "0" when the low-order byte of ADxREG6 is read.

11.4.16. ADxREG7 (Conversion Result Register 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR7							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR7				-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR7[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR7	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG7 is read and is cleared to "0" when the low-order byte of ADxREG7 is read.
0	ADR7RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG7 register and is cleared to "0" when the low-order byte of ADxREG7 is read.

11.4.17. ADxREG8 (Conversion Result Register 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR8							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR8				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR8[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR8	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG8 is read and is cleared to "0" when the low-order byte of ADxREG8 is read.
0	ADR8RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG8 register and is cleared to "0" when the low-order byte of ADxREG8 is read.

11.4.18. ADxREG9 (Conversion Result Register 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR9							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR9				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR9[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR9	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG9 is read and is cleared to "0" when the low-order byte of ADxREG9 is read.
0	ADR9RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG9 register and is cleared to "0" when the low-order byte of ADxREG9 is read.

11.4.19. ADxREG10 (Conversion Result Register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR10				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR10[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR10	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG10 is read and is cleared to "0" when the low-order byte of ADxREG10 is read.
0	ADR10RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG10 register and is cleared to "0" when the low-order byte of ADxREG10 is read.

11.4.20. ADxREG11 (Conversion Result Register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ADR11							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ADR11				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:4	ADR11[11:0]	R	The value of an AD conversion result
3:2	-	R	Read as "0".
1	OVR11	R	Over Run flag 0: No overrun occurred 1: Overrun occurred This flag is set to "1" when a new AD conversion result is stored before the value of ADxREG11 is read and is cleared to "0" when the low-order byte of ADxREG11 is read.
0	ADR11RF	R	AD conversion result store flag 0: No result stored 1: Result stored This flag is set to "1" when an AD conversion result is stored in the ADxREG11 register and is cleared to "0" when the low-order byte of ADxREG11 is read.

11.4.21. PMD Trigger Program Registers

AD conversion can be started by a trigger signal from the PMD.

The PMD trigger program registers are used to specify the program to be started by each of 12 triggers generated by the PMD, to control the interrupt to be generated upon completion of the program and to select the AIN input to be used.

The PMD trigger program registers include 3 types of registers.

- PMD Trigger Program Number Select Register (ADxPSEL0 to ADxPSEL11)
  The PMD Trigger Program Number Select Registers (ADxPSEL0 to ADxPSEL11) specify the program number (0 to 5) to be started by each of 12 AD conversion corresponding to 12 triggers (PMD0TRG0 to PMD0TRG5, PMD1TRG0 to PMD1TRG5) generated by the PMD.
  ADxPSEL0 to ADxPSEL5 registers correspond to PMD0TRG0 to PMD0TRG5. ADxPSEL6 to ADxPSEL11 registers correspond to PMD1TRG0 to PMD1TRG5.
- PMD Trigger Interrupt Select Register (ADxPINTS0 to ADxPINTS5)
  The PMD Trigger Interrupt Select Registers (ADxPINTS0 to ADxPINTS5) select the presence or absence of interrupt or the kind of interrupt (INTADxPDA or INTADxPDB) to be generated upon completion of each program number.
  ADxPINTS0 to ADxPINTS5 registers correspond to program 0 to 5.
- PMD Trigger Program Select Register (ADxPSET0 to ADxPSET5)
  The PMD Trigger Program Select Registers (ADxPSET0 to ADxPSET5) for each of program number (0 to 5) specify AIN input number which is used for AD conversion, and the U/V/W phase to inform to the vector engine. One PMD Trigger Program Select Register is assigned for one program number. One PMD Trigger Program Select Register has four sets of registers which specify AIN input number which is used for AD conversion, and the U/V/W phase to inform to the vector engine. Conversion result for each set is stored to Conversion Result Register 0 to 3 (ADxREG0 to ADxREG3),

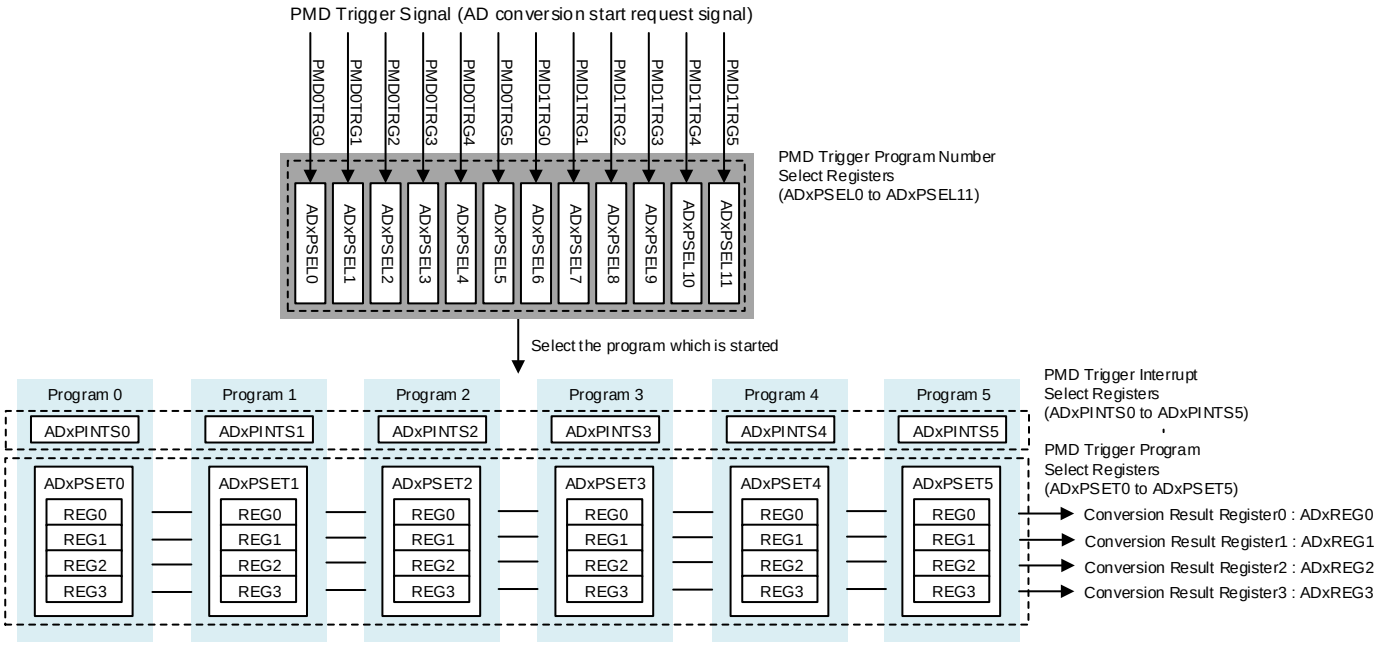


Figure 11.2 PMD Trigger Program Registers

11.4.21.1. ADxPSEL0 to ADxPSEL11 (PMD Trigger Program Number Select Register 0 to 11)

ADxPSEL0: PMD Trigger Program Number Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS0	-	-	-	-	PMDS0		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS0	R/W	PMD0TRG0 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS0[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL1: PMD Trigger Program Number Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS1	-	-	-	-	PMDS1		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS1	R/W	PMD0TRG1 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS1[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL2: PMD Trigger Program Number Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS2	-	-	-	-	PMDS2		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS2	R/W	PMD0TRG2 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS2[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL3: PMD Trigger Program Number Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS3	-	-	-	-	PMDS3		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS3	R/W	PMD0TRG3 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS3[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL4: PMD Trigger Program Number Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS4	-	-	-	-	PMDS4		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS4	R/W	PMD0TRG4 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS4[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL5: PMD Trigger Program Number Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS5	-	-	-	-	PMDS5		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS5	R/W	PMD0TRG5 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS5[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL6: PMD Trigger Program Number Select Register 6

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS6	-	-	-	-	PMDS6		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS6	R/W	PMD1TRG0 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS6[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL7: PMD Trigger Program Number Select Register 7

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS7	-	-	-	-	PMDS7		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS7	R/W	PMD1TRG1 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS7[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL8: PMD Trigger Program Number Select Register 8

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS8	-	-	-	-	PMDS8		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS8	R/W	PMD1TRG2 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS8[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL9: PMD Trigger Program Number Select Register 9

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS9	-	-	-	-	PMDS9		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS9	R/W	PMD1TRG3 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS9[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL10: PMD Trigger Program Number Select Register 10

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS10	-	-	-	-	PMDS10		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS10	R/W	PMD1TRG4 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS10[2:0]	R/W	Program number select (Refer to Table 11.1)

ADxPSEL11: PMD Trigger Program Number Select Register 11

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	PENS11	-	-	-	-	PMDS11		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7	PENS11	R/W	PMD1TRG5 trigger control 0: Disable 1: Enable
6:3	-	R	Read as "0".
2:0	PMDS11[2:0]	R/W	Program number select (Refer to Table 11.1)

Table 11.1 Program number select

<PMDS0[2:0]> to <PMDS11[2:0]>	Program number
000	Program 0
001	Program 1
010	Program 2
011	Program 3
100	Program 4
101	Program 5
110	Reserved
111	Reserved

11.4.21.2. ADxPINTS0 to 5 (PMD Trigger Interrupt Select Register 0 to 5)

ADxPINTS0: PMD Trigger Interrupt Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	INTSEL0	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	INTSEL0[1:0]	R/W	Interrupt select 00: No interrupt output 01: INTADxPDA 10: INTADxPDB 11: No interrupt output The starting interrupt is selected for program 0.

ADxPINTS1: PMD Trigger Interrupt Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	INTSEL1	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	INTSEL1[1:0]	R/W	Interrupt select 00: No interrupt output 01: INTADxPDA 10: INTADxPDB 11: No interrupt output The starting interrupt is selected for program 1.

ADxPINTS2: PMD Trigger Interrupt Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	INTSEL2	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	INTSEL2[1:0]	R/W	Interrupt select 00: No interrupt output 01: INTADxPDA 10: INTADxPDB 11: No interrupt output The starting interrupt is selected for program 2

ADxPINTS3: PMD Trigger Interrupt Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	INTSEL3	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	INTSEL3[1:0]	R/W	Interrupt select 00: No interrupt output 01: INTADxPDA 10: INTADxPDB 11: No interrupt output The starting interrupt is selected for program 3

ADxPINTS4: PMD Trigger Interrupt Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	INTSEL4	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	INTSEL4[1:0]	R/W	Interrupt select 00: No interrupt output 01: INTADxPDA 10: INTADxPDB 11: No interrupt output The starting interrupt is selected for program 4

ADxPINTS5: PMD Trigger Interrupt Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-	-	-	-	INTSEL5	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	INTSEL5[1:0]	R/W	Interrupt select 00: No interrupt output 01: INTADxPDA 10: INTADxPDB 11: No interrupt output The starting interrupt is selected for program 5

11.4.21.3. ADxPSET0 to 5 (PMD Trigger Program Select Register 0 to 5)

Each ADxPSETn (n = 0 to 5: Program number) is composed of 4 sets. One set includes <AINSPnm[4:0]> for selecting AIN pin, <UVWISnm[1:0]> for informing to the vector engine, and <ENSPnm> for enabling ADxREGm (m = 0 to 3).

ADxREGm / ADxPSETn	m = 0	m = 1	m = 2	m = 3
n = 0	<ENSP00> <UVWIS00[1:0]> <AINSP00[4:0]>	<ENSP01> <UVWIS01[1:0]> <AINSP01[4:0]>	<ENSP02> <UVWIS02[1:0]> <AINSP02[4:0]>	<ENSP03> <UVWIS03[1:0]> <AINSP03[4:0]>
n = 1	<ENSP10> <UVWIS10[1:0]> <AINSP10[4:0]>	<ENSP11> <UVWIS11[1:0]> <AINSP11[4:0]>	<ENSP12> <UVWIS12[1:0]> <AINSP12[4:0]>	<ENSP13> <UVWIS13[1:0]> <AINSP13[4:0]>
n = 2	<ENSP20> <UVWIS20[1:0]> <AINSP20[4:0]>	<ENSP21> <UVWIS21[1:0]> <AINSP21[4:0]>	<ENSP22> <UVWIS22[1:0]> <AINSP22[4:0]>	<ENSP23> <UVWIS23[1:0]> <AINSP23[4:0]>
n = 3	<ENSP30> <UVWIS30[1:0]> <AINSP30[4:0]>	<ENSP31> <UVWIS31[1:0]> <AINSP31[4:0]>	<ENSP32> <UVWIS32[1:0]> <AINSP32[4:0]>	<ENSP33> <UVWIS33[1:0]> <AINSP33[4:0]>
n = 4	<ENSP40> <UVWIS40[1:0]> <AINSP40[4:0]>	<ENSP41> <UVWIS41[1:0]> <AINSP41[4:0]>	<ENSP42> <UVWIS42[1:0]> <AINSP42[4:0]>	<ENSP43> <UVWIS43[1:0]> <AINSP43[4:0]>
n = 5	<ENSP50> <UVWIS50[1:0]> <AINSP50[4:0]>	<ENSP51> <UVWIS51[1:0]> <AINSP51[4:0]>	<ENSP52> <UVWIS52[1:0]> <AINSP52[4:0]>	<ENSP53> <UVWIS53[1:0]> <AINSP53[4:0]>

Table 11.2 Select the AIN pin

<AINSP00[4:0]> to <AINSP53[4:0]>	ADC Unit A	ADC Unit B
0_0000	AINA0	AINB0
0_0001	AINA1	AINB1
0_0010	AINA2	AINB2
0_0011	AINA3	AINB3
0_0100	AINA4	AINB4
0_0101	AINA5	AINB5
0_0110	AINA6	AINB6
0_0111	AINA7	AINB7
0_1000	AINA8	AINB8
0_1001	AINA9	AINB9
0_1010	AINA10	AINB10
0_1011	AINA11	AINB11
0_1100	AINA12	AINB12
0_1101	AINA13	AINB13
0_1110	AINA14	AINB14
0_1111	Reserved	AINB15
1_0000	Reserved	AINB16
1_0001 to 1_1111	Reserved	Reserved

ADxPSET0: PMD Trigger Program Register 0

	31	30	29	28	27	26	25	24
bit symbol	ENSP03	UVWIS03		AINSP03				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSP02	UVWIS02		AINSP02				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSP01	UVWIS01		AINSP01				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSP00	UVWIS00		AINSP00				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP03	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	UVWIS03[1:0]	R/W	Phase select (for Vector Engine) See table below.
28:24	AINSP03[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
23	ENSP02	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	UVWIS02[1:0]	R/W	Phase select (for Vector Engine) See table below.
20:16	AINSP02[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
15	ENSP01	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	UVWIS01[1:0]	R/W	Phase select (for Vector Engine) See table below.
12:8	AINSP01[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
7	ENSP00	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	UVWIS00[1:0]	R/W	Phase select (for Vector Engine) See table below.
4:0	AINSP00[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".

Phase select

<UVWIS00[1:0]> to <UVWIS03[1:0]>	Phase select
00	Not specified
01	U
10	V
11	W

ADxPSET1: PMD Trigger Program Register 1

	31	30	29	28	27	26	25	24
bit symbol	ENSP13	UVWIS13		AINSP13				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSP12	UVWIS12		AINSP12				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSP11	UVWIS11		AINSP11				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSP10	UVWIS10		AINSP10				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP13	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	UVWIS13[1:0]	R/W	Phase select (for Vector Engine) See table below.
28:24	AINSP13[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
23	ENSP12	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	UVWIS12[1:0]	R/W	Phase select (for Vector Engine) See table below.
20:16	AINSP12[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
15	ENSP11	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	UVWIS11[1:0]	R/W	Phase select (for Vector Engine) See table below.
12:8	AINSP11[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
7	ENSP10	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	UVWIS10[1:0]	R/W	Phase select (for Vector Engine) See table below.
4:0	AINSP10[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".

Phase select

<UVWIS10[1:0]> to <UVWIS13[1:0]>	Phase select
00	Not specified
01	U
10	V
11	W

ADxPSET2: PMD Trigger Program Register 2

	31	30	29	28	27	26	25	24
bit symbol	ENSP23	UVWIS23		AINSP23				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSP22	UVWIS22		AINSP22				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSP21	UVWIS21		AINSP21				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSP20	UVWIS20		AINSP20				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP23	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	UVWIS23[1:0]	R/W	Phase select (for Vector Engine) See table below.
28:24	AINSP23[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
23	ENSP22	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	UVWIS22[1:0]	R/W	Phase select (for Vector Engine) See table below.
20:16	AINSP22[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
15	ENSP21	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	UVWIS21[1:0]	R/W	Phase select (for Vector Engine) See table below.
12:8	AINSP21[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
7	ENSP20	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	UVWIS20[1:0]	R/W	Phase select (for Vector Engine) See table below.
4:0	AINSP20[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".

Phase select

<UVWIS20[1:0]> to <UVWIS23[1:0]>	Phase select
00	Not specified
01	U
10	V
11	W

ADxPSET3: PMD Trigger Program Register 3

	31	30	29	28	27	26	25	24
bit symbol	ENSP33	UVWIS33		AINSP33				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSP32	UVWIS32		AINSP32				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSP31	UVWIS31		AINSP31				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSP30	UVWIS30		AINSP30				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP33	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	UVWIS33[1:0]	R/W	Phase select (for Vector Engine) See table below.
28:24	AINSP33[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
23	ENSP32	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	UVWIS32[1:0]	R/W	Phase select (for Vector Engine) See table below.
20:16	AINSP32[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
15	ENSP31	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	UVWIS31[1:0]	R/W	Phase select (for Vector Engine) See table below.
12:8	AINSP31[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
7	ENSP30	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	UVWIS30[1:0]	R/W	Phase select (for Vector Engine) See table below.
4:0	AINSP30[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".

Phase select

<UVWIS30[1:0]> to <UVWIS33[1:0]>	Phase select
00	Not specified
01	U
10	V
11	W

ADxPSET4: PMD Trigger Program Register 4

	31	30	29	28	27	26	25	24
bit symbol	ENSP43	UVWIS43		AINSP43				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSP42	UVWIS42		AINSP42				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSP41	UVWIS41		AINSP41				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSP40	UVWIS40		AINSP40				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP43	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	UVWIS43[1:0]	R/W	Phase select (for Vector Engine) See table below.
28:24	AINSP43[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
23	ENSP42	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	UVWIS42[1:0]	R/W	Phase select (for Vector Engine) See table below.
20:16	AINSP42[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
15	ENSP41	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	UVWIS41[1:0]	R/W	Phase select (for Vector Engine) See table below.
12:8	AINSP41[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
7	ENSP40	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	UVWIS40[1:0]	R/W	Phase select (for Vector Engine) See table below.
4:0	AINSP40[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".

Phase select

<UVWIS40[1:0]> to <UVWIS43[1:0]>	Phase select
00	Not specified
01	U
10	V
11	W

ADxPSET5: PMD Trigger Program Register 5

	31	30	29	28	27	26	25	24
bit symbol	ENSP53	UVWIS53		AINSP53				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSP52	UVWIS52		AINSP52				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSP51	UVWIS51		AINSP51				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSP50	UVWIS50		AINSP50				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP53	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	UVWIS53[1:0]	R/W	Phase select (for Vector Engine) See table below.
28:24	AINSP53[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
23	ENSP52	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	UVWIS52[1:0]	R/W	Phase select (for Vector Engine) See table below.
20:16	AINSP52[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
15	ENSP51	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	UVWIS51[1:0]	R/W	Phase select (for Vector Engine) See table below.
12:8	AINSP51[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".
7	ENSP50	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	UVWIS50[1:0]	R/W	Phase select (for Vector Engine) See table below.
4:0	AINSP50[4:0]	R/W	AIN select Refer to "Table 11.2 Select the AIN pin".

Phase select

<UVWIS50[1:0]> to <UVWIS53[1:0]>	Phase select
00	Not specified
01	U
10	V
11	W

11.4.22. ADxTSET03/ADxTSET47/ADxTSET811 (Timer Trigger Program Registers)

AD conversion can be started by trigger signal INTTB51 generated by TMRB5.

Timer Trigger Program Registers are configured by twelve setting register set. The Timer Trigger Program Register number (m = 0 to 11) corresponds to the AD Conversion Result Register number.

Setting <ENSTm> to "1" enables one setting register set.

<AINSTm[4:0]> are used to select the AIN input which is used for AD conversion.

When finished AD conversion by a timer trigger, INTADxTMR is generated.

Table 11.3 Select the AIN pin

<AINST0[4:0]> to <AINST11[4:0]>	ADC Unit A	ADC Unit B
0_0000	AINA0	AINB0
0_0001	AINA1	AINB1
0_0010	AINA2	AINB2
0_0011	AINA3	AINB3
0_0100	AINA4	AINB4
0_0101	AINA5	AINB5
0_0110	AINA6	AINB6
0_0111	AINA7	AINB7
0_1000	AINA8	AINB8
0_1001	AINA9	AINB9
0_1010	AINA10	AINB10
0_1011	AINA11	AINB11
0_1100	AINA12	AINB12
0_1101	AINA13	AINB13
0_1110	AINA14	AINB14
0_1111	Reserved	AINB15
1_0000	Reserved	AINB16
1_0001 to 1_1111	Reserved	Reserved

ADxTSET03: Timer Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST3	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINST3[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
23	ENST2	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINST2[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
15	ENST1	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINST1[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
7	ENST0	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINST0[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".

ADxTSET47: Timer Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENST6	-	-	AINST6				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENST5	-	-	AINST5				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENST4	-	-	AINST4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST7	R/W	ADxREG7 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINST7[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
23	ENST6	R/W	ADxREG6 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINST6[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
15	ENST5	R/W	ADxREG5 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINST5[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
7	ENST4	R/W	ADxREG4 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINST4[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".

ADxTSET811: Timer Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENST10	-	-	AINST10				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENST9	-	-	AINST9				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENST8	-	-	AINST8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST11	R/W	ADxREG11 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINST11[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
23	ENST10	R/W	ADxREG10 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINST10[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
15	ENST9	R/W	ADxREG9 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINST9[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".
7	ENST8	R/W	ADxREG8 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINST8[4:0]	R/W	AIN select Refer to "Table 11.3 Select the AIN pin".

11.4.23. ADxSSET03/ADxSSET47/ADxSSET811 (Software Trigger Program Registers)

AD conversion can be started by software.

Software Trigger Program Registers are configured by twelve setting register set. The numbers of the Software Trigger Program Registers (m = 0 to 11) correspond to those of the AD Conversion Result Registers.

Setting <ENSSm> to "1" enables one setting register set.

<AINSSm[4:0]> are used to select the AIN input which is used for AD conversion.

When finished AD conversion by a software trigger, INTADxSFT is generated.

Table 11.4 Select the AIN pin

<AINSS0[4:0]> to <AINSS11[4:0]>	ADC Unit A	ADC Unit B
0_0000	AINA0	AINB0
0_0001	AINA1	AINB1
0_0010	AINA2	AINB2
0_0011	AINA3	AINB3
0_0100	AINA4	AINB4
0_0101	AINA5	AINB5
0_0110	AINA6	AINB6
0_0111	AINA7	AINB7
0_1000	AINA8	AINB8
0_1001	AINA9	AINB9
0_1010	AINA10	AINB10
0_1011	AINA11	AINB11
0_1100	AINA12	AINB12
0_1101	AINA13	AINB13
0_1110	AINA14	AINB14
0_1111	Reserved	AINB15
1_0000	Reserved	AINB16
1_0001 to 1_1111	Reserved	Reserved

ADxSSET03: Software Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSS2	-	-	AINSS2				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSS1	-	-	AINSS1				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSS0	-	-	AINSS0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS3	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINSS3[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
23	ENSS2	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINSS2[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
15	ENSS1	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINSS1[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
7	ENSS0	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINSS0[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".

ADxSSET47: Software Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSS6	-	-	AINSS6				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSS5	-	-	AINSS5				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSS4	-	-	AINSS4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS7	R/W	ADxREG7 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINSS7[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
23	ENSS6	R/W	ADxREG6 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINSS6[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
15	ENSS5	R/W	ADxREG5 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINSS5[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
7	ENSS4	R/W	ADxREG4 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINSS4[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".

ADxSSET811: Software Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSS10	-	-	AINSS10				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSS9	-	-	AINSS9				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSS8	-	-	AINSS8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS11	R/W	ADxREG11 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINSS11[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
23	ENSS10	R/W	ADxREG10 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINSS10[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
15	ENSS9	R/W	ADxREG9 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINSS9[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".
7	ENSS8	R/W	ADxREG8 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINSS8[4:0]	R/W	AIN select Refer to "Table 11.4 Select the AIN pin".

11.4.24. ADxASET03/ADxASET47/ADxASET811 (Constant Conversion Program Registers)

AD conversion can be used as constant conversion.

Constant Conversion Program Registers are configured by twelve setting register set. The numbers of the Constant Conversion Program Registers (m = 0 to 11) correspond to those of the AD Conversion Result Registers.

Setting <ENSAm> to "1" enables one setting register set.

<AINSAm[4:0]> are used to select the AIN input which is used for AD conversion.

Table 11.5 Select the AIN pin

<AINSA0[4:0]> to <AINSA11[4:0]>	ADC Unit A	ADC Unit B
0_0000	AINA0	AINB0
0_0001	AINA1	AINB1
0_0010	AINA2	AINB2
0_0011	AINA3	AINB3
0_0100	AINA4	AINB4
0_0101	AINA5	AINB5
0_0110	AINA6	AINB6
0_0111	AINA7	AINB7
0_1000	AINA8	AINB8
0_1001	AINA9	AINB9
0_1010	AINA10	AINB10
0_1011	AINA11	AINB11
0_1100	AINA12	AINB12
0_1101	AINA13	AINB13
0_1110	AINA14	AINB14
0_1111	Reserved	AINB15
1_0000	Reserved	AINB16
1_0001 to 1_1111	Reserved	Reserved

ADxASET03: Constant Conversion Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSA2	-	-	AINSA2				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSA1	-	-	AINSA1				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSA0	-	-	AINSA0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA3	R/W	ADxREG3 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINSA3[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
23	ENSA2	R/W	ADxREG2 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINSA2[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
15	ENSA1	R/W	ADxREG1 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINSA1[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
7	ENSA0	R/W	ADxREG0 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINSA0[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".

ADxASET47: Constant Conversion Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSA6	-	-	AINSA6				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSA5	-	-	AINSA5				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSA4	-	-	AINSA4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA7	R/W	ADxREG7 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINSA7[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
23	ENSA6	R/W	ADxREG6 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINSA6[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
15	ENSA5	R/W	ADxREG5 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINSA5[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
7	ENSA4	R/W	ADxREG4 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINSA4[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".

ADxASET811: Constant Conversion Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSA11	-	-	AINSA11				
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	ENSA10	-	-	AINSA10				
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	ENSA9	-	-	AINSA9				
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ENSA8	-	-	AINSA8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA11	R/W	ADxREG11 enable 0: Disable 1: Enable
30:29	-	R	Read as "0".
28:24	AINSA11[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
23	ENSA10	R/W	ADxREG10 enable 0: Disable 1: Enable
22:21	-	R	Read as "0".
20:16	AINSA10[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
15	ENSA9	R/W	ADxREG9 enable 0: Disable 1: Enable
14:13	-	R	Read as "0".
12:8	AINSA9[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".
7	ENSA8	R/W	ADxREG8 enable 0: Disable 1: Enable
6:5	-	R	Read as "0".
4:0	AINSA8[4:0]	R/W	AIN select Refer to "Table 11.5 Select the AIN pin".

11.4.25. ADxMOD3 (Mode Setting Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	1	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	PMODE			-	-	-
After reset	0	1	0	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:11	-	R/W	Write as "00000".
10:6	-	R/W	Write as "10001".
5:3	PMODE[2:0]	R/W	Write as "100".
2:0	-	R/W	Write as "000".

Note: The value which is specified in above table must be set to ADxMOD3.

11.5. Operation Descriptions

11.5.1. Analog Reference Voltage

For the analog reference voltage, "High" level and "Low" level voltage are applied to the VREFHx and VREFLx pins of ADC unit B, respectively. There are no registers for controlling current between VREFHx and VREFLx. The constant current is flowed continuously.

Note1: During AD conversion, do not change the output data of port I/J/K, to avoid the influence on the conversion result.

Note2: AD conversion results might be unstable by the following conditions.

- Input operation is executed during AD conversion.
- Output operation is executed during AD conversion.
- Output current of port varies during AD conversion.

Take a countermeasure such as averaging the multiple conversion results, to get precise value.

11.5.2. Starting AD Conversion

AD conversion is started by the following 3 trigger signals.

- Software trigger
- PMD trigger
- Timer trigger

These start triggers are given priorities as shown below.

PMD trigger 0 > ··· > PMD trigger 5 > Timer trigger > Software trigger > Constant conversion

When a higher-priority trigger occurs while an AD conversion is in progress, the ongoing AD conversion is handled to stop, and a program corresponding to a higher-priority trigger starts. When the higher-priority PMD trigger occurs while a PMD triggered AD conversion is in progress, the PMD trigger is handled after the ongoing AD conversion is completed.

It has some delay from generation of trigger to start of AD conversion. The following timing chart and table show the delay.

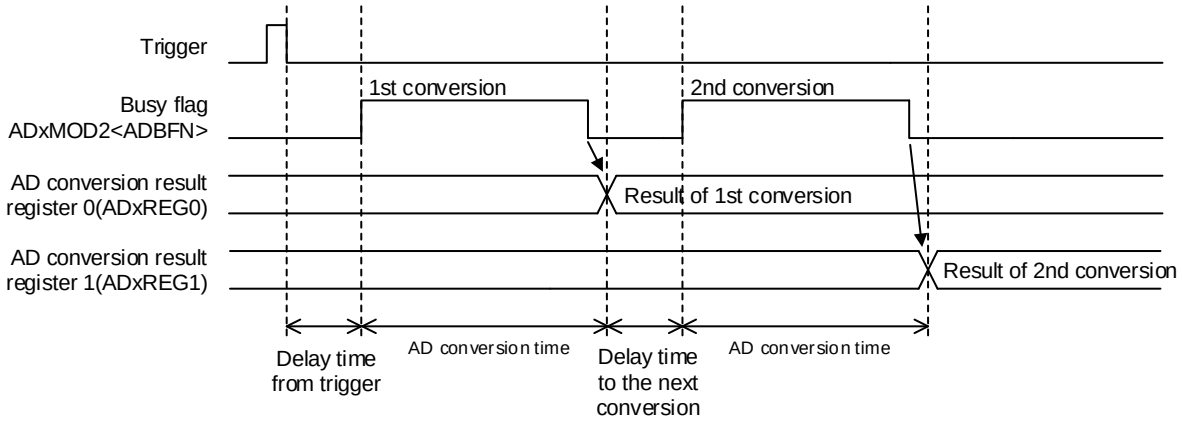


Figure 11.3 Timing Chart of AD Conversion

Table 11.6 AD conversion time and delay time (SCLK = 40MHz)

		fsys = 80MHz		fsys = 40MHz	
	Conversion start trigger	MIN	MAX	MIN	MAX
Delay time from trigger [μs] (Note1)	PMD	0.125	0.163	0.225	0.3
	TMRB	0.125	0.263	0.225	0.5
	Software, Constant conversion	0.138	0.275	0.25	0.525
AD comparison time [μs]	-	1.85		1.85	
Delay time to the next conversion [μs] (Note2)	PMD	0.1	0.125	0.175	0.225
	TMRB, Software, Constant conversion	0.1	0.238	0.175	0.425

Note1: Delay time from trigger to start of AD conversion

Note2: Delay time to the 2nd or after conversion in plural conversions with one trigger

11.5.3. AD Conversion Monitoring Function

The ADC has the AD conversion monitoring function. When this function is enabled, an interrupt is generated when the result matches the specified comparison condition.

To enable the monitoring function, set ADxCMPCRn<CMPnEN> to "1". In the monitoring function, the value of AD conversion result register specified by <REGSn> and the value of ADxCMPn are compared. When the compared result is matched to the condition which is specified by <ADBIGn>, the compare counter is incremented. The comparison is executed at the timing of storing the conversion result.
When the compare counter is matched to the condition specified by <CMPCNTn[3:0]>, INTADxCPn is generated.

Note1: The AD conversion result store flag (<ADR0RF> to <ADR11RF>) is not cleared by reading of the AD conversion monitoring function.

Note2: The AD conversion monitoring function differs from reading the conversion result by software. Therefore, when the next conversion is completed without reading the previous result by software, the overrun flag (<OVR0> to <OVR11>) is set.

11.6. Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of PMD and timer trigger.

11.6.1. Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADxSSET03, ADxSSET47 and ADxSSET811 (Refer to Figure 11.4).

When ADxMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register (Refer to Figure 11.5).

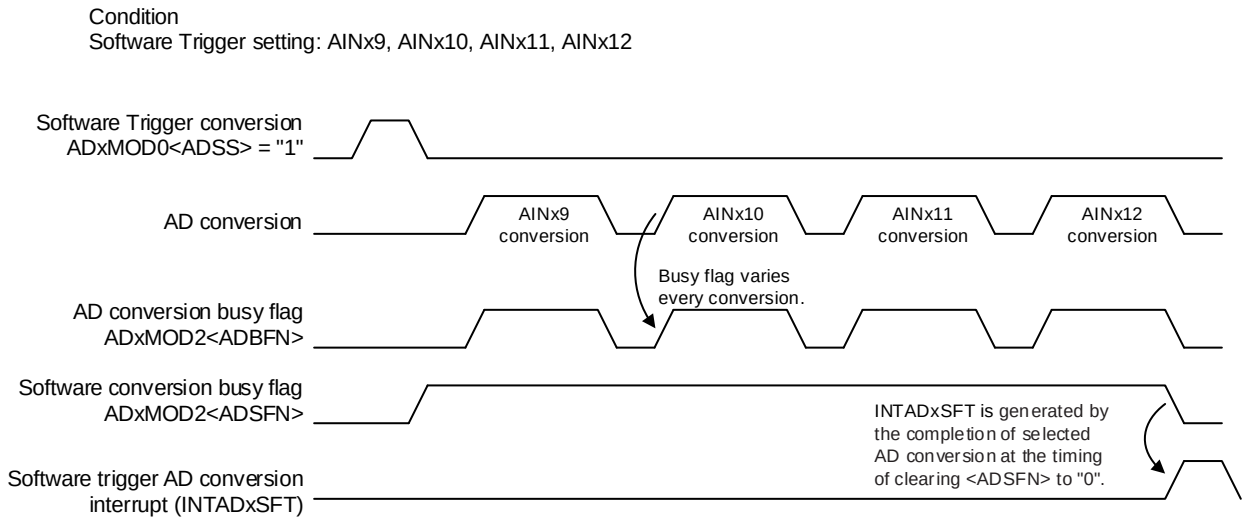


Figure 11.4 Software Trigger AD Conversion Timing Chart

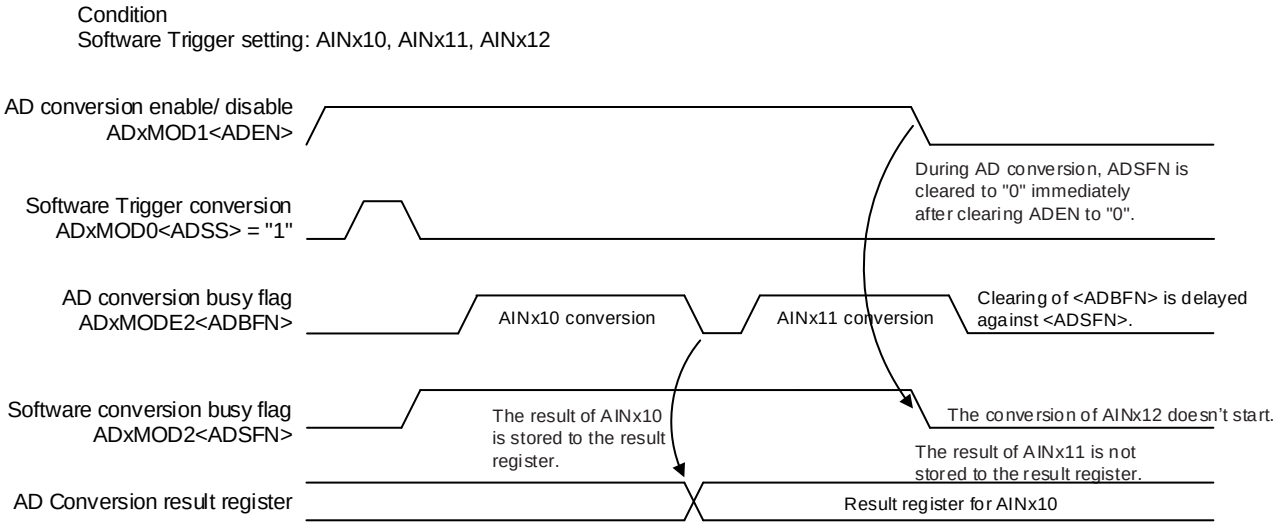


Figure 11.5 Writing "0" to <ADEN> during Software Trigger AD Conversion

11.6.2. Constant Conversion

In the constant conversion, when the next conversion completes without reading the previous result from the conversion result register, the overrun flag is set to "1". In this case, the previous conversion result in the conversion result register is overwritten by the next result. The overrun flag is cleared by reading of the conversion result. (Refer to Figure 11.6)

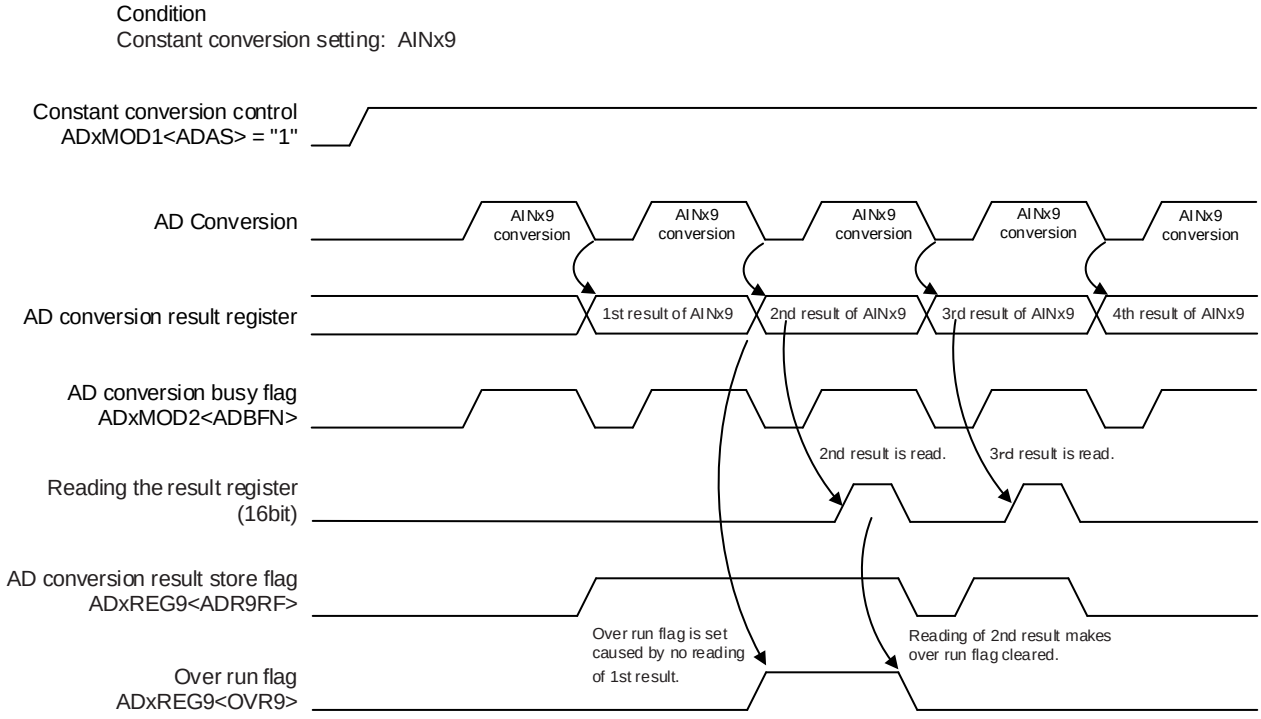


Figure 11.6 Constant Conversion Timing Chart

11.6.3. AD Conversion by Trigger

When the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately and start AD conversion corresponding to PMD trigger. (Refer to Figure 11.7) After the completion of conversion by PMD trigger, the software trigger conversion starts from the beginning programmed setting. When the timer trigger is occurred, also same response. (Refer to Figure 11.8)

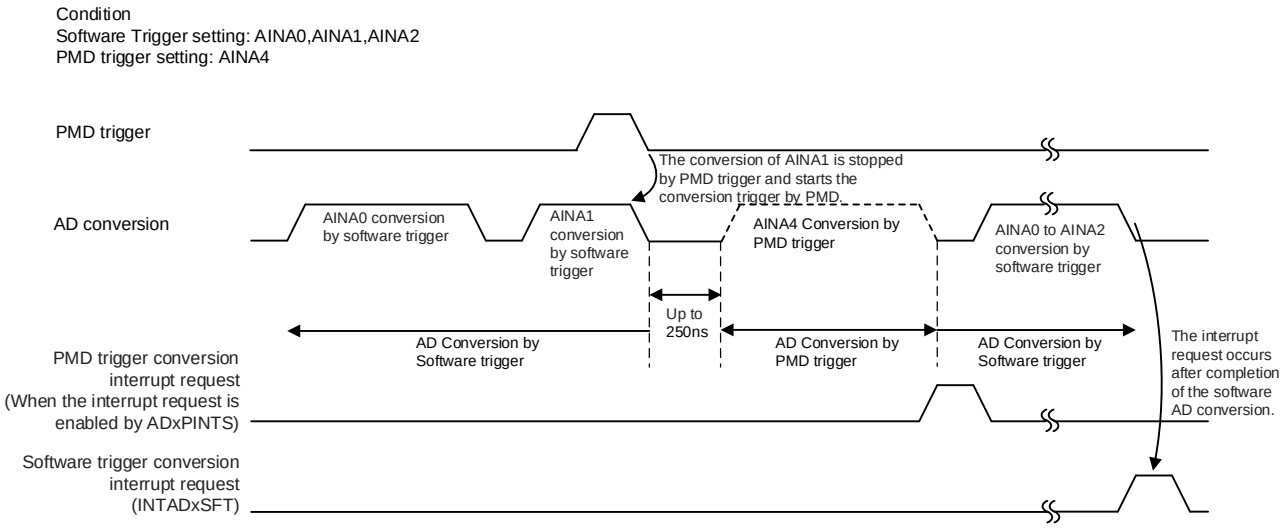


Figure 11.7 AD Conversion by PMD Trigger During AD conversion

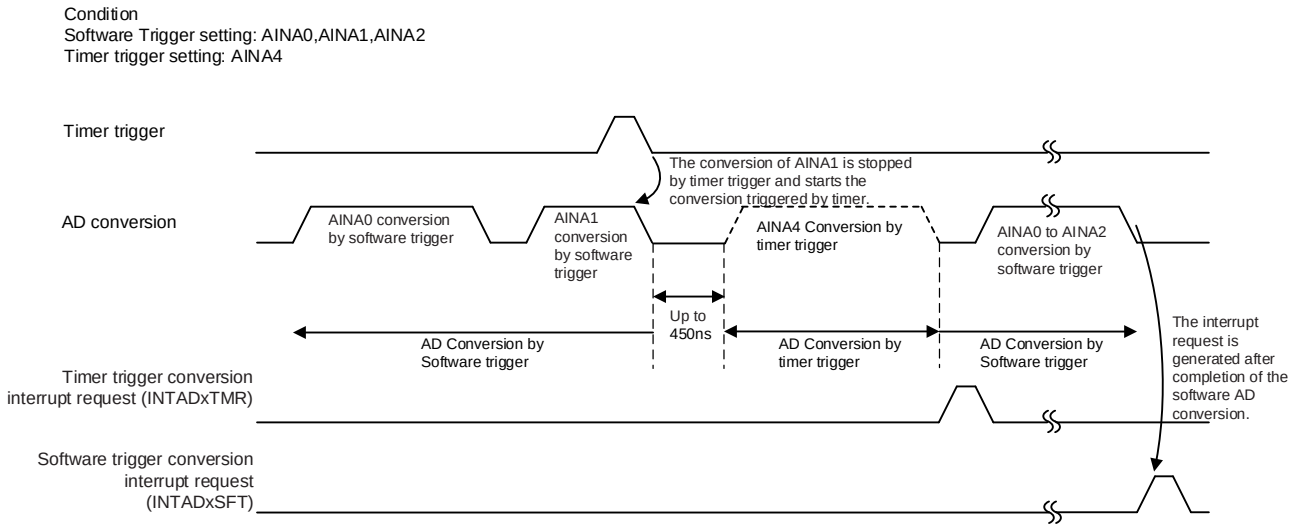


Figure 11.8 AD Conversion by Timer Trigger During AD conversion

Note: When timer trigger is not used, do not use INTTB51. Set TB5IM<TBIM1> to "1".

11.7. Usage Example

11.7.1. Example When Using PMD ch0 (3 shunts) and one AD converter (unit A)

An example circuit when using PMD ch0 (3 shunts) and one AD converter (unit A) is shown below;

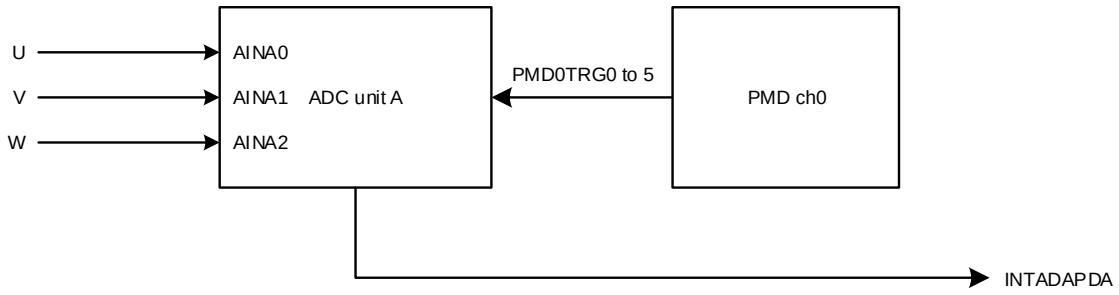


Figure 11.9 Example Circuit When Using PMD Ch0 (3 Shunts) and One AD Converter (Unit A)

An example of AD converter unit A settings is shown in "Table 11.7 Example of AD Converter Unit A Settings". A program number (0 to 5) for 6 trigger signals from PMD ch0 (PMD0TRG0 to PMD0TRG5) is selected by ADAPSEL0<PMDS0[2:0]> to ADAPSEL5<PMDS5[2:0]>.

"U", "V" and "W" indicate the phase of a motor. AIN inputs which obtain these phases are selected.

When a trigger signal input occurs, AD conversion of AIN specified by ADAPSETn[7:0] is firstly started. Secondly, AD conversion of AIN specified by ADAPSETn[15:8] is started. Each conversion result is stored to one's conversion result register, INTADAPDA interrupt request occurs.

Table 11.7 Example of AD Converter Unit A Settings

Program number	0	1	2	3	4	5
ADAPSETn[7:0]	U	U	W	V	W	U
ADAPSETn[15:8]	V	W	U	U	V	W
Interrupt request	INTADAPDA	INTADAPDA	INTADAPDA	INTADAPDA	INTADAPDA	INTADAPDA

11.7.2. Example When Using PMD ch0 (3 shunts) and two AD converters (unit A and B)

An example circuit when using PMD ch0 (3 shunts) and two AD converters (unit A and B) is shown below;

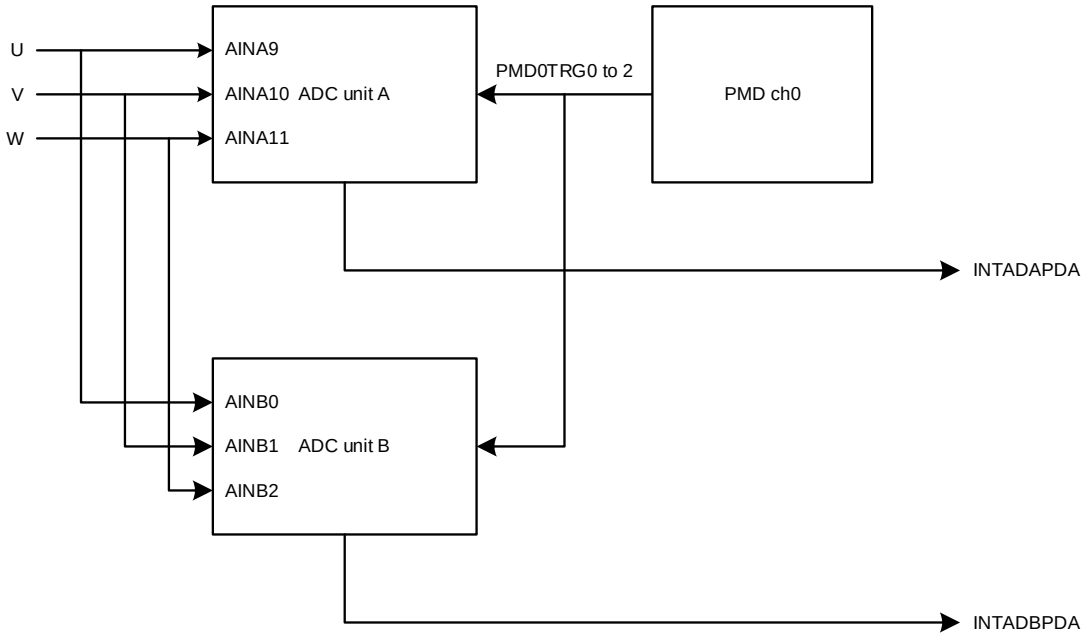


Figure 11.10 Example Circuit When Using PMD ch0 (3 shunts) and two AD converters (unit A and B)

An example of AD converter unit A and B settings is shown in "Table 11.8 Example of AD Converter Unit A and B Settings".

A program number (0 to 2) for 3 trigger signals from PMD ch0 (PMD0TRG0 to PMD0TRG2) is selected by ADAPSEL0<PMDS0[2:0]> to ADAPSEL2<PMDS2[2:0]> in AD converter unit A.

A program number (0 to 2) for 3 trigger signals from PMD ch0 (PMD0TRG0 to PMD0TRG2) is selected by ADBPSEL0<PMDS0[2:0]> to ADBPSEL2<PMDS2[2:0]> in AD converter unit B.

"U", "V" and "W" indicate the phase of a motor. AIN inputs which obtain these phases are selected.

When a trigger signal input occurs, AD conversions of AIN specified by ADAPSETn[7:0] and ADBPSETn[7:0] are started simultaneously. Each conversion result is stored to one's conversion result register, INTADAPDA and INTADBPDA interrupt requests occur.

Table 11.8 Example of AD Converter Unit A and B Settings

Program number	0	1	2
ADAPSETn[7:0]	U	V	W
Interrupt request	INTADAPDA	INTADAPDA	INTADAPDA
ADBPSETn[7:0]	U	V	W
Interrupt request	INTADBPDA	INTADBPDA	INTADBPDA

11.7.3. Example When Using PMD ch0 (3 shunts), ch1 (1 shunt) and two AD converters (unit A and B)

An example circuit when using PMD ch0 (3 shunts), ch1 (1 shunt) and two AD converters (unit A and B) is shown below;

The AD convertor unit B is set as not generating an interrupt request in this case.

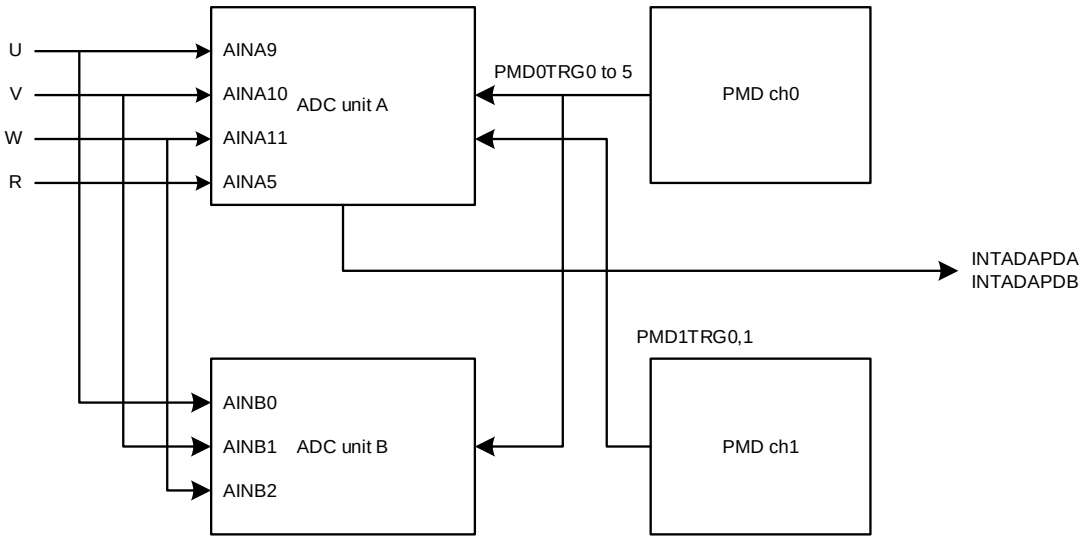


Figure 11.11 Example Circuit When Using PMD ch0 (3 shunts), ch1 (1 shunt) and two AD converters (unit A and B)

An example of AD converter unit A and B settings is shown in "Table 11.9 Example of AD Converter Unit A Settings" and "Table 11.10 Example of AD Converter Unit B Settings".

In AD converter unit A, a program number (0 to 2) for 6 trigger signals (PMD0TRG0 to PMD0TRG5) out of 8 trigger signals from PMD ch0 and ch1 is selected by ADAPSEL0<PMDS0[2:0]> to ADAPSEL5<PMDS5[2:0]> and a program number (3 and 4) for 2 trigger signals (PMD1TRG0 and PMD1TRG1) is selected by ADAPSEL6<PMDS6[2:0]> and ADAPSEL7<PMDS7[2:0]>.

In AD converter unit B, a program number (0 to 2) for 6 trigger signals (PMD0TRG0 to PMD0TRG5) from PMD ch0 is selected by ADBPSEL0<PMDS0[2:0]> to ADBPSEL5<PMDS5[2:0]>

"U", "V" and "W" indicate the phase of a motor. AIN inputs which obtain these phases are selected. "R" indicates a resistor. AIN input which is connected with the resister is selected.

When a trigger signal input occurs, AD conversions of AIN specified by ADAPSETn[7:0], ADAPSETn[15:8], ADAPSETn[23:16] and ADBPSET[7:0] are started. Each conversion result is stored to one's conversion result register, INTADAPDA interrupt request for the trigger signal from PMD ch0 and INTADAPDB interrupt request for one from PMD ch1 occur.

Table 11.9 Example of AD Converter Unit A Settings

Trigger signal	PMD0TRG0 PMD0TRG3	PMD0TRG1 PMD0TRG4	PMD0TRG2 PMD0TRG5	PMD1TRG0	PMD1TRG1
Program number	**0**	**1**	**2**	**3**	**4**
ADAPSETn[7:0]	U	V	W	-	-
ADAPSETn[15:8]	-	-	-	R	-
ADAPSETn[23:16]	-	-	-	-	R
Interrupt request	INTADAPDA	INTADAPDA	INTADAPDA	-	INTADAPDB

Table 11.10 Example of AD Converter Unit B Settings

Trigger signal	PMD0TRG0 PMD0TRG3	PMD0TRG1 PMD0TRG4	PMD0TRG2 PMD0TRG5
Program number	**0**	**1**	**2**
ADBPSETn[7:0]	V	W	U
Interrupt request	-	-	-

11.7.4. Example When Using PMD ch0 (1 shunt) and one AD converter (unit A)

An example circuit when using PMD ch0 (1 shunt) and one AD converter (unit A) is shown below;

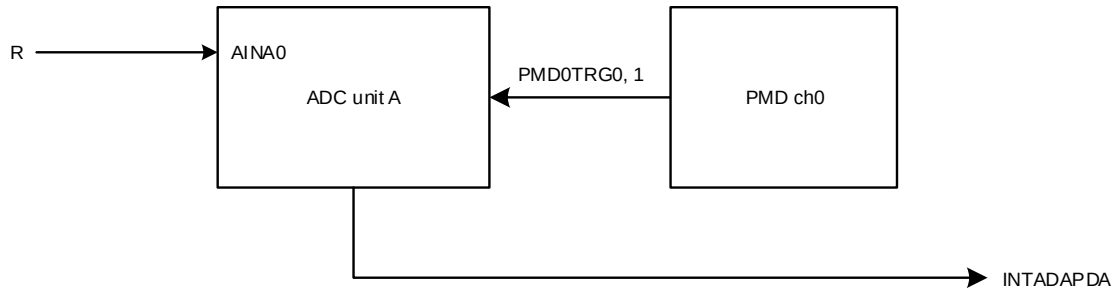


Figure 11.12 Example Circuit When Using PMD ch0 (1 shunt) and one AD converter (unit A)

An example of AD converter unit A settings is shown in "Table 11.11 Example of AD Converter Unit A Settings".

A program number (0 and 1) for 2 trigger signals from PMD ch0 (PMD0TRG0 and PMD0TRG1) is selected by ADAPSEL0<PMDS0[2:0]> and ADAPSEL1<PMDS1[2:0]>.

"R" indicates a resistor. AIN input which is connected with the resister is selected.

When a trigger signal input occurs, AD conversion of AIN specified by ADAPSETn[7:0] is firstly started. Secondly, AD conversion of AIN specified by ADAPSETn[15:8] is started. Each conversion result is stored to one's conversion result register, INTADAPDA interrupt request occurs.

Table 11.11 Example of AD Converter Unit A Settings

Trigger signal	PMD0TRG0	PMD0TRG1
Program number	**0**	**1**
ADAPSETn[7:0]	R	-
ADAPSETn[15:8]	-	R
Interrupt request	-	INTADAPDA

12. Motor Control Circuit (PMD)

12.1. Outline

The PMD consists the conduction output control and the DC overvoltage detection input to achieve one-shunt sensorless motor control. It also can control a motor in conjunction with ADC, AMP/CMP and VE.

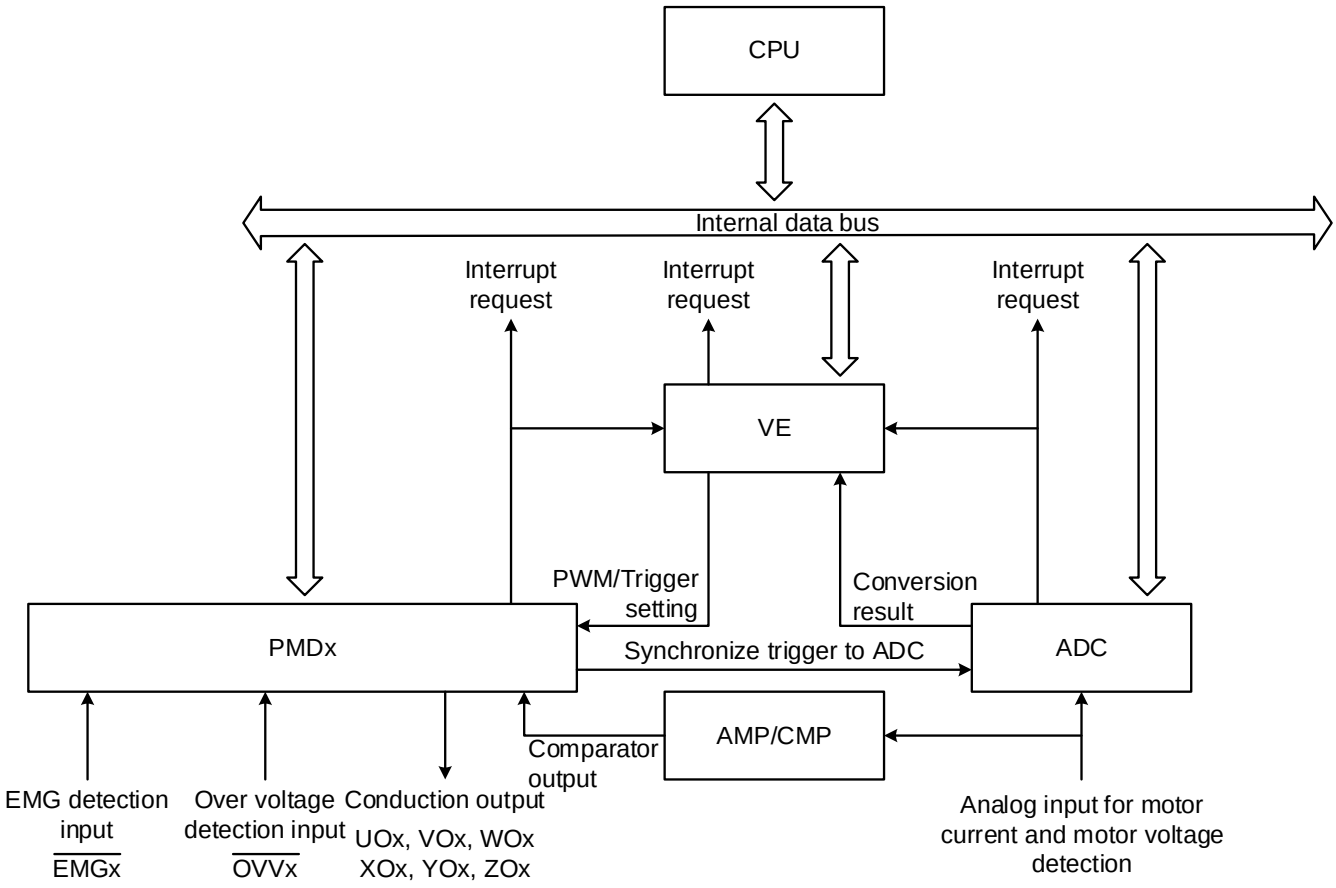


Figure 12.1 In Conjunction with PMD, ADC, AMP/CMP and VE

12.2. Block Diagram

The PMD is broadly divided into two blocks: a waveform generator circuit and a synchronous trigger generator circuit.

The following circuits constitute the waveform generation circuit.

- The pulse width modulation circuit generates 3-phase independent PWM waveforms with equal PWM frequencies.
- The conduction control circuit determines the output patterns of the upper and lower phases of U, V, and W phases.
- The protection circuit performs emergency output stop by $\overline{\text{EMGx}}$ pin input, PMDxEMG (Comparator output) and $\overline{\text{OVVx}}$ pin input.
- The dead time control circuit prevents short circuits during switching of the upper and lower phases.

The synchronous trigger generator also generates a synchronous trigger signal to ADC.

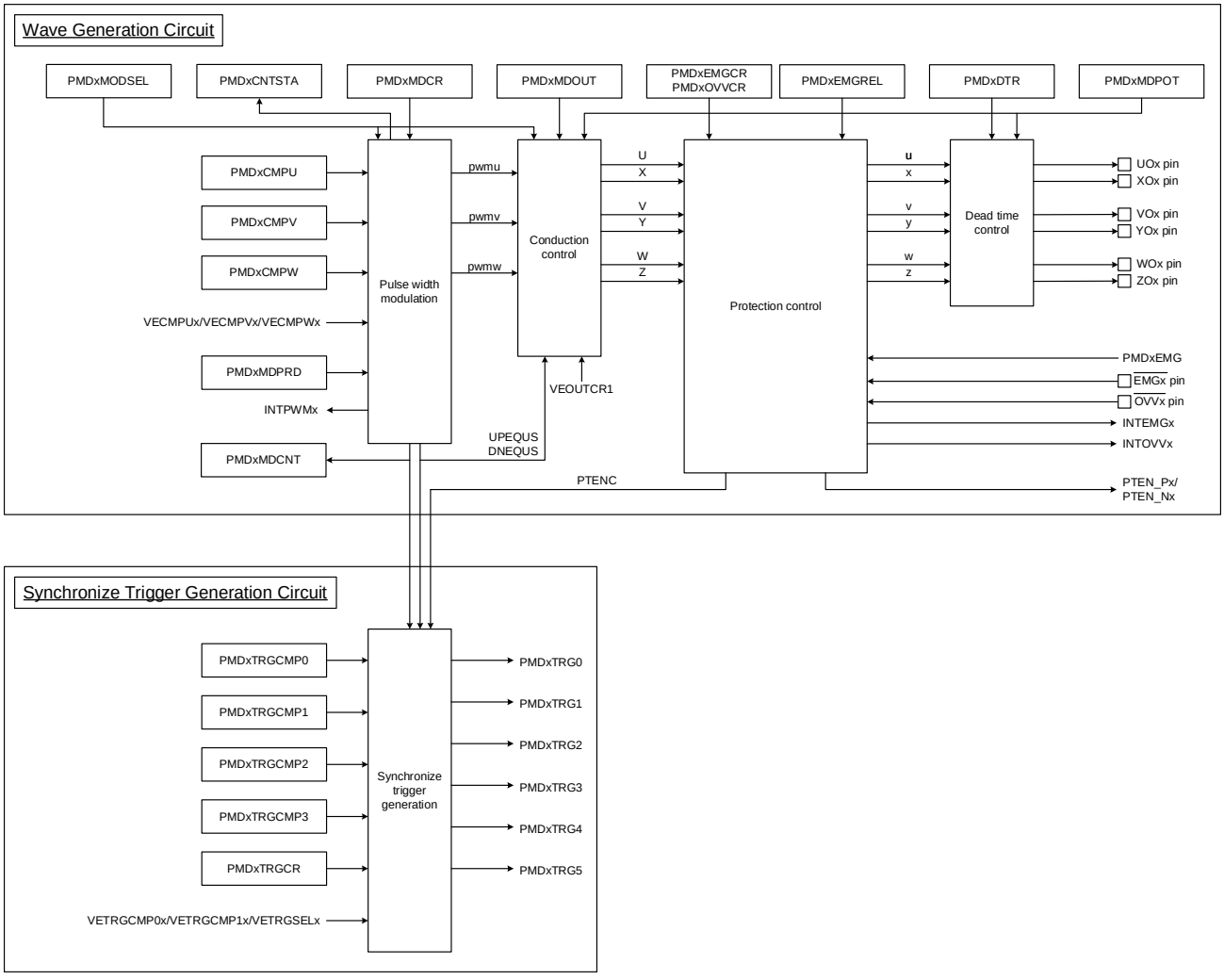


Figure 12.2 PMD Block Diagram

12.3. Registers

12.3.1. List of Registers

The control registers and addresses are listed below.

Register name		Base+ (Address)
PMD Enable Register	PMDxMDEN	0x0000
Port Output Mode Register	PMDxPORTMD	0x0004
PMD Control Register	PMDxMDCR	0x0008
PWM Counter Status Register	PMDxCNTSTA	0x000C
PWM Counter Register	PMDxMDCNT	0x0010
PWM Cycle Register	PMDxMDPRD	0x0014
PWM Compare U Register	PMDxCMPU	0x0018
PWM Compare V Register	PMDxCMPV	0x001C
PWM Compare W Register	PMDxCMPW	0x0020
Mode Select Register	PMDxMODESEL	0x0024
PMD Output Control Register	PMDxMDOUT	0x0028
PMD Output Setting Register	PMDxMDPOT	0x002C
EMG Release Register	PMDxEMGREL	0x0030
EMG Control Register	PMDxEMGCR	0x0034
EMG Status Register	PMDxEMGSTA	0x0038
OVV Control Register	PMDxOVVCR	0x003C
OVV Status Register	PMDxOVVSTA	0x0040
Dead Time Register	PMDxDTR	0x0044
Trigger Compare 0 Register	PMDxTRGCMP0	0x0048
Trigger Compare 1 Register	PMDxTRGCMP1	0x004C
Trigger Compare 2 Register	PMDxTRGCMP2	0x0050
Trigger Compare 3 Register	PMDxTRGCMP3	0x0054
Trigger Control Register	PMDxTRGCR	0x0058
Trigger Output Mode Setting Register	PMDxTRGMD	0x005C
Trigger Output Select Register	PMDxTRGSEL	0x0060
Reserved	-	0x007C

Note: Access to the "Reserved" address is prohibited.

12.3.2. PMD Control

12.3.2.1. PMDxMDEN (PMD Enable Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	-	PWMEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
0	PWMEN	R/W	Waveform synthesis function enable/disable control 0: Disabled 1: Enabled When the ports are used as a function output (PWM output), when PMDxMDEN<PWMEN> is set to "0", the ports used as a function output are set to "Hi-Z". Set PMDxMDEN<PWMEN> to "1" after making settings such as output port polarity selection, etc..

12.3.2.2. PMDxPORTMD (Port Output Mode Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	PORTMD	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1:0	PORTMD[1:0]	R/W	U-phase, V-phase, W-phase, X-phase, Y-phase, Z-phase output pin control 00: Upper phase "Hi-Z"/Lower phase "Hi-Z" 01: Upper phase "Hi-Z"/Lower phase ON (Note3) 10: Upper phase ON (Note3)/Lower phase "Hi-Z" 11: Upper phase ON (Note3)/Lower phase ON (Note3) Set the output of the upper phase (UOx, VOx, and WOx pins) and the lower phase (XOx, YOx, and ZOx pins) when debugger is halted.

Note1: When PMDxMDEN<PWMEN> is "0", the UOx, VOx, WOx, XOx, YOx, and ZOx output pins are set to "Hi-Z" regardless of the port output control register setting.

Note2: The UOx, VOx, WOx, XOx, YOx, and ZOx output pin control is also performed according to the setting of PMDxEMGCR<EMGMD[1:0]> when the $\overline{\text{EMGx}}$ pin input is "Low" level

Note3: ON: PWM output continues.

12.3.2.3. PMDxMODESEL (Mode Select Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	-	MDSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
0	MDSEL	R/W	Mode select register 0: Bus mode 1: VE Mode PMDxMODESEL<MDSEL> selects the input method to the subsequent stage of the double buffer from the bus mode which uses the register value set from the bus or the VE mode which uses the value from VE. PMDxCMPU, PMDxCMPV, PMDxCMPW, PMDxTRGCMP0, PMDxTRGCMP1, and PMDxMDOUT have a double buffer configuration. When PMDxMODESEL<MDSEL> is "0", the data written to the register is immediately loaded to the subsequent stage of the double buffer. When PMDxMODESEL<MDSEL> is "1", the data is loaded to the subsequent stage of the double buffer at the update timing inside the PMD.

12.3.3. Pulse Width Modulation circuit

The pulse width modulation circuit has a PMD counter as a 16-bit up/down counter. It generates PWM carrier with a resolution of 12.5ns@fsys = 80MHz. The PWM carrier waveform mode can be selected from edge-aligned PWM (sawtooth wave modulation) as PWM mode 0 and center-aligned PWM (triangle wave modulation) as mode 1.

In addition, by setting the PWM cycle extension mode (PMDxMDCR<PWMCK> = "1"), the PWM counter generates PWM carriers with a resolution of 50ns.

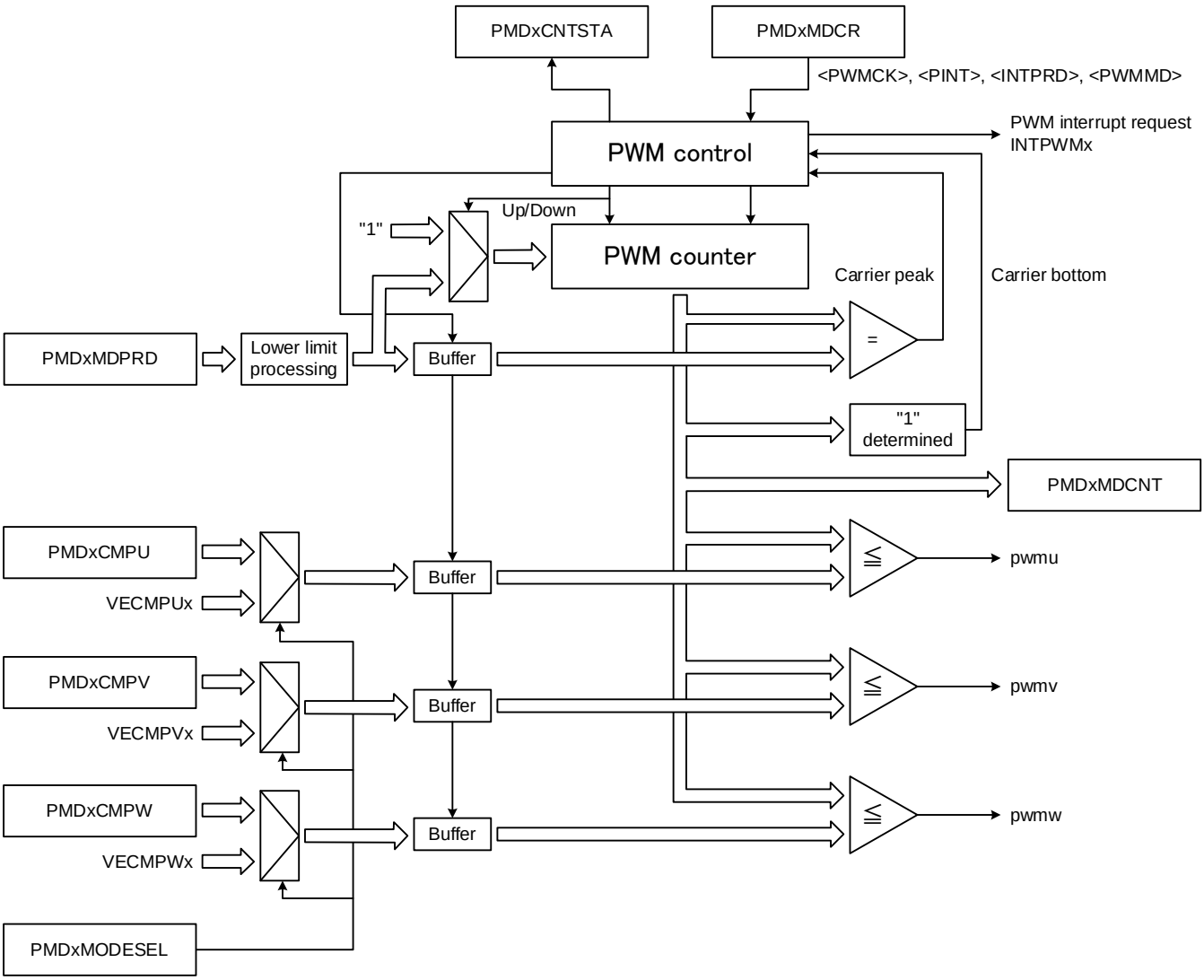


Figure 12.3 Pulse Width Modulation Circuit

12.3.3.1. PWM Cycle Setting

The PWM cycle is determined by PMDxMDPRD<MDPRD[15:0]>. PMDxMDPRD has a double-buffer configuration and the comparator inputs are updated in PWM cycle. Updating (loading every half cycle) for each PWM half cycle can also be selected.

$$\text{Sawtooth wave PWM: PMDxMDPRD register setting} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]}}$$

$$\text{Triangle wave PWM: PMDxMDPRD register setting} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency} \times 2\text{[Hz]}}$$

12.3.3.2. Compare Function

The comparator compares the 3-phase PWM compare register (PMDxCMPU/V/W) value with the carrier generated by PMDxMDCNT<MDCNT[15:0]> to produce a PWM waveform with the desired duty.

The PMD compare register for each phase has a comparison register, and a double-buffer configuration.

The PMD compare register value is synchronized with the PWM cycle and loaded into the compare register when PMDxMDCNT<MDCNT[15:0]> matches PMDxMDPRD<MDPRD[15:0]>.

Update in PWM half cycle (loading every half cycle) can also be selected.

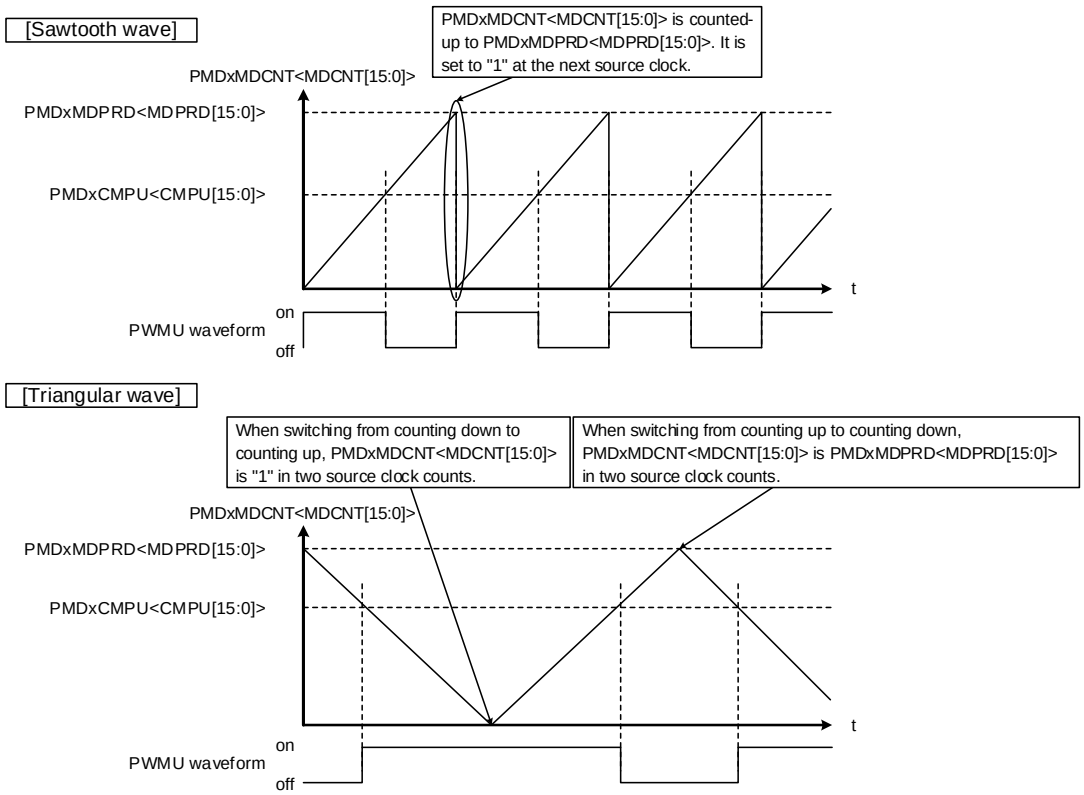


Figure 12.4 PWM Waveforms of Sawtooth and Triangle Waveform

12.3.3.3. Waveform Mode

The following 3-phase PWM generation modes can be selected.

(1) 3-phase independent duty mode: Generates 3-phase independent PWM waveforms by setting independent values to the 3-phase PMD compare register. This is used for arbitrary drive waveform generation, such as a sine wave.

(2) 3-phase common duty mode: A value is set only to the U-phase PMD compare register, and 3-phase identical PWM waveforms are generated with the set value of the U-phase. This is used for square-wave driving of DC motor.

12.3.3.4. Interrupt Processing

The pulse width modulation circuit generates a PWM interrupt request (INTPWMx) in synchronization with the PWM waveform. The frequency of INTPWMx can be selected from the following.

(1) Once per half cycle of PWM cycle

(2) Once per one cycle of PWM cycle

(3) Once per two cycles of PWM cycle

(4) Once per four cycles of PWM cycle

12.3.3.5. PMDxMDCR (PMD Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:7	-	R	Read as "0".
6	PWMCK	R/W	PWM cycle extension mode selection 0: Normal cycle 1: Quadruple cycle When the normal cycle is selected, the PWM counter operates at a resolution of 12.5ns@fsys = 80MHz. The sawtooth wave resolution is 12.5ns and the triangle wave resolution is 25ns. When the quadruple cycle is selected, the PWM counter operates at a resolution of 50ns@fsys = 80MHz. The resolution of the sawtooth wave is 50ns and that of the triangle wave is 100ns.
5	SYNTMD	R/W	UOx, VOx, WOx, XOx, YOx, ZOx output pin mode setting Set the UOx, VOx, WOx, XOx, YOx, and ZOx output pins. For details, refer to Table 12.2.
4	DTYMD	R/W	Duty mode selection 0: 3-phase common duty mode 1: 3-phase independent duty mode Selects whether to use PMDxCMPU value for the duty setting for all three phases or to use PMDxCMPU/V/W setting value independently for each of the three phases.
3	PINT	R/W	INTPWMx generation timing select 0: When PWM counter PMDxMDCNT<MDCNT[15:0]> = "0x0001" (min), INTPWMx occurs. 1: When PWM counter PMDxMDCNT<MDCNT[15:0]> = PMDxMDPRD<MDPRD[15:0]>, INTPWMx occurs. INTPWMx generation timing is selected from when PWM counter is "0x0001" or when PWM counter is PMDxMDPRD<MDPRD[15:0]>. When PMDxMDCR<PINT>= "1", INTPWMx generation timing is as shown below according to the setting of PMDxMDCR<PWMMD>. When PMDxMDCR<PWMMD> = "0": When PMDxMDCNT<MDCNT[15:0]> equals to PMDxMDPRD <MDPRD[15:0]>, INTPWMx occurs. When PMDxMDCR<PWMMD>= "1": When PMDxMDCNT<MDCNT[15:0]> equals to "0x0001" or PMDxMDPRD<MDPRD[15:0]>, INTPWMx occurs.
2:1	INTPRD[1:0]	R/W	INTPWMx generation cycle select 00: Generates INTPWMx once per half cycle of PWM cycle (only when PMDxMDCR<PWMMD>= "1") (Note) 01: Generates INTPWMx once per one cycle of PWM cycle 10: Generates INTPWMx once per 2 cycles of PWM cycle 11: Generates INTPWMx once per 4 cycles of PWM cycle Selects the frequency at which INTPWMx occurs from every 0.5, 1, 2, and 4 PWM cycles.

0	PWMMD	R/W	PWM carrier waveform selection 0: PWM mode 0 (edge-aligned PWM, sawtooth wave modulation) 1: PWM mode 1 (center-aligned PWM, triangle wave modulation) Selects PWM mode. PWM mode 0 is the edge-aligned PWM, and PWM mode 1 is the center-aligned PWM.

Note: When PMDxMDCR<INTPRD[1:0]> is "00" and PMDxMDCNT<MDCNT[15:0]> matches "0x0001" or PMDxMDPRD<MDPRD[15:0]>, data are stored to the subsequent stage of the double buffers of PMDxCMPU/V/W and PMDxMDPRD.

12.3.3.6. PMDxCNTSTA (PWM Counter Status Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	-	UPDWN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
0	UPDWN	R	PWM counter flag 0: Up counting in progress 1: Down counting in progress Indicates whether the PWM counter is up counting or down counting. When PWM mode 0 is selected, "0" is always read.

12.3.3.7. PMDxMDCNT (PWM Counter Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	MDCNT							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	MDCNT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	MDCNT[15:0]	R	PWM counter reading The value of the 16-bit PWM counter for counting the PWM cycle can be read. When PMDxMDCR<PWMCK> is "0", the PWM counter operates at a resolution of 12.5ns@fsys = 80MHz. The sawtooth wave resolution is 12.5ns and the triangle wave resolution is 25ns. When PMDxMDCR<PWMCK> is "1", the PWM counter operates at a resolution of 50ns@fsys = 80MHz. The sawtooth wave resolution is 50ns and the triangle wave resolution is 100ns. When the waveform synthesis function is disabled (PMDxMDEN<PWMEN>= "0"), the PWM counter value is as follows according to the setting of PMDxMDCR<PWMMD>. When PMDxMDCR<PWMMD> = "0": "0x0001" When PMDxMDCR<PWMMD> = "1": value of PMDxMDPRD<MDPRD[15:0]>

12.3.3.8. PMDxMDPRD (PWM Cycle Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	MDPRD							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	MDPRD							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	MDPRD[15:0]	R/W	PWM cycle setting Set PMDxMDPRD<MDPRD[15:0]> to "0x0010" or higher. Set PMDxMDPRD<MDPRD[15:0]> to the PWM cycle. It has a double buffer configuration, so it can be changed even while the PWM counter is running. The value written every PWM cycle is loaded in the subsequent stage of the double buffer. It is loaded when the PWM counter matches PMDxMDPRD<MDPRD[15:0]>. When PMDxMDCR<INTPRD[1:0]> is set to "00", it is loaded when the PWM counter matches "0x0001" or PMDxMDPRD<MDPRD[15:0]>. At this time, set the LSB of PMDxMDPRD<MDPRD[15:0]> to "0". When PMDxMDPRD<MDPRD[15:0]> is set to a value less than "0x0010", PMDxMDPRD <MDPRD[15:0]> is set to setting value, but it is treated as setting "0x0010".

Note: Do not write to this register by a byte transfer instruction (upper 8 bits ([15:8]) and lower 8 bits ([7:0]) are written separately).

12.3.3.9. PMDxCMPU (PWM Compare U Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CMPU1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CMPU1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	CMPU1[15:0]	R	First stage of double buffer (written value) can be read.
		W	U-phase PWM pulse width setting Set the U-phase PWM pulse width with PMDxCMPU<CMPU1[15:0]>. The resolution of the U-phase PWM pulse width is 12.5ns@fsys = 80MHz. The sawtooth wave resolution is 12.5ns and the triangle wave resolution is 25ns. PMDxCMPU<CMPU1[15:0]> has a double buffer configuration, so it can be changed even while the PWM counter is running. The value written every PWM cycle is loaded in the subsequent stage of the double buffer. PWM pulse width is determined by comparing the value of the PWM counter and the subsequent stage of double buffer. PWM pulse width is loaded when the PWM counter matches PMDxMDPRD<MDPRD[15:0]>. When PMDxMDCR<INTPRD[1:0]> is set to "00", PWM pulse width is loaded when the PWM counter matches "0x0001" or PMDxMDPRD<MDPRD[15:0]>.

Note1: To load the data written in this register to the subsequent stage of double buffer immediately, set PMDxMODESEL <MDSEL> to "0".

Note2: Do not write to this register by byte transfer command (upper 8 bits ([15:8]) and lower 8 bits ([7:0]) are written separately).

12.3.3.10. PMDxCMPV (PWM Compare V Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CMPV1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CMPV1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	CMPV1[15:0]	R	First stage of double buffer (written value) can be read.
		W	V-phase PWM pulse width setting Set the V-phase PWM pulse width with PMDxCMPV<CMPV1[15:0]>. The resolution of the V-phase PWM pulse width is 12.5ns@fsys = 80MHz. The sawtooth wave resolution is 12.5ns and the triangle wave resolution is 25ns. PMDxCMPV<CMPV1[15:0]> has a double buffer configuration, so it can be changed even while the PWM counter is running. The value written every PWM cycle is loaded in the subsequent stage of the double buffer. PWM pulse width is determined by comparing the value of the PWM counter and the subsequent stage of double buffer. PWM pulse width is loaded when the PWM counter matches PMDxMDPRD<MDPRD[15:0]>. When PMDxMDCR<INTPRD[1:0]> is set to "00", PWM pulse width is loaded when the PWM counter matches "0x0001" or PMDxMDPRD<MDPRD[15:0]>.

Note1: To load the data written in this register to the subsequent stage of double buffer immediately, set PMDxMODESEL <MDSEL> to "0".

Note2: Do not write to this register by byte transfer command (upper 8 bits ([15:8]) and lower 8 bits ([7:0]) are written separately).

12.3.3.11. PMDxCMPW (PWM Compare W Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CMPW1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CMPW1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	CMPW1[15:0]	R	First stage of double buffer (written value) can be read.
		W	W-phase PWM pulse width setting Set the W-phase PWM pulse width with PMDxCMPW<CMPW1[15:0]>. The resolution of the W-phase PWM pulse width is 12.5ns@fsys = 80MHz. The sawtooth wave resolution is 12.5ns and the triangle wave resolution is 25ns. PMDxCMPW<CMPW1[15:0]> has a double buffer configuration, so it can be changed even while the PWM counter is running. The value written every PWM cycle is loaded in the subsequent stage of the double buffer. PWM pulse width is determined by comparing the value of the PWM counter and the subsequent stage of double buffer. PWM pulse width is loaded when the PWM counter matches PMDxMDPRD<MDPRD[15:0]>. When PMDxMDCR<INTPRD[1:0]> is set to "00", PWM pulse width is loaded when the PWM counter matches "0x0001" or PMDxMDPRD<MDPRD[15:0]>.

Note1: To load the data written in this register to the subsequent stage of double buffer immediately, set PMDxMODESEL <MDSEL> to "0".

Note2: Do not write to this register by byte transfer command (upper 8 bits ([15:8]) and lower 8 bits ([7:0]) are written separately).

12.3.4. Conduction Control Circuit

The conduction control circuit controls the PWM output according to the content set in PMDxMDOUT and PMDxMDPOT.

The conduction control circuit is broadly divided into the selection of PMDxMDOUT reload timing and the PWM output setting.

Use PMDxMDPOT<PSYNCS> to select PMDxMDOUT reload timing. Synchronization timing can be selected from PWM counter synchronization and asynchronous. PWM output is updated at reload timing.

For the PWM output setting, set "Low" active/"High" active for each of the upper and lower phases of the PWM output by using PMDxMDPOT<POLH>, <POLL>.

Set the selection of "High" level/"Low" level output and PWM output in PMDxMDOUT<WPWM>, <VPWM>, <UPWM> for the U-, V-, and W-phases of the PWM output, respectively.

When "High" level/"Low" level output is selected, the conduction control circuit outputs "High" level or "Low" level.

When PWM output is selected, the conduction control circuit outputs the PWM waveform.

Refer to Table 12.1 and Table 12.2 for the relation between the setting of PWM output by PMDxMDOUT and the setting of polarity by PMDxMDPOT.

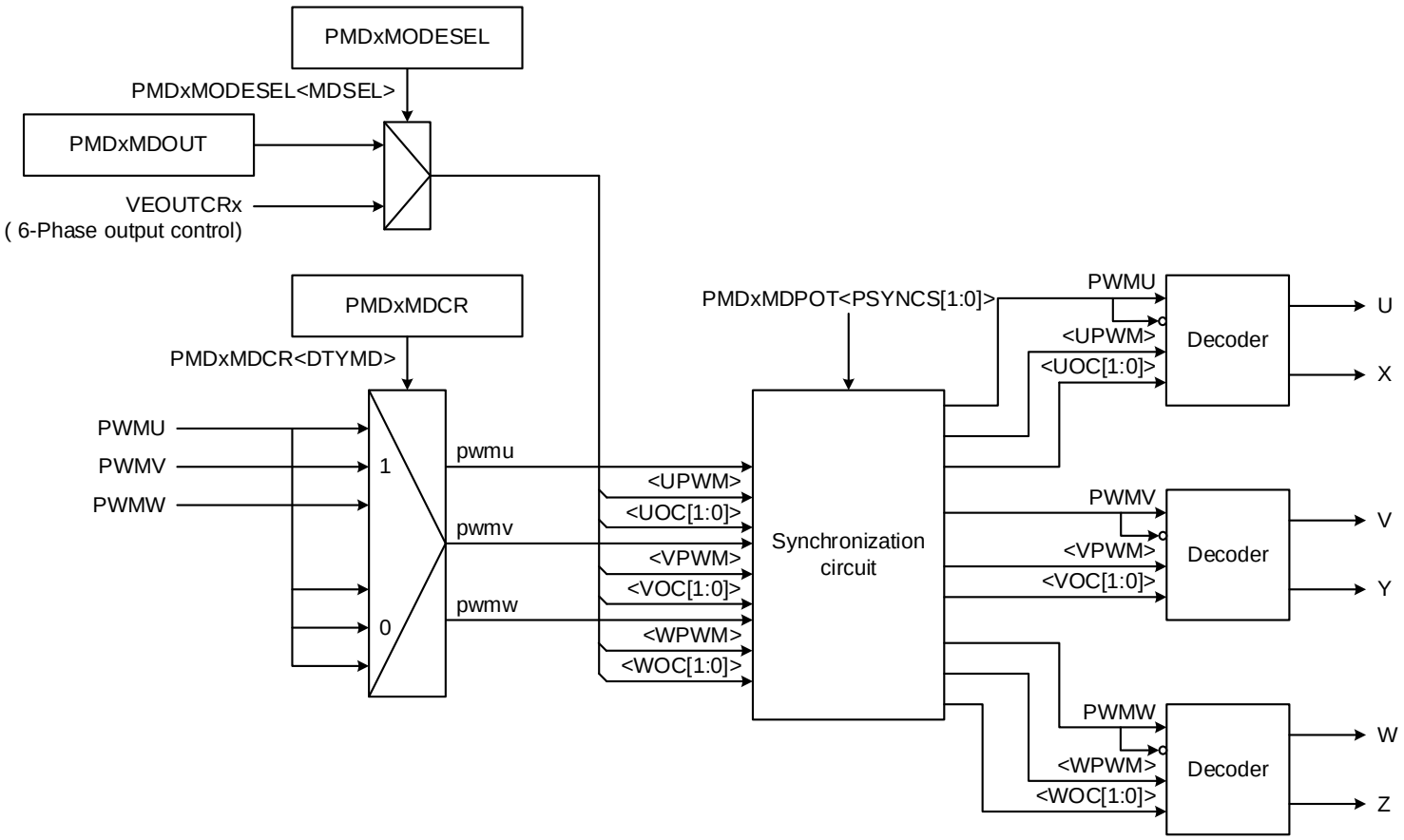


Figure 12.5 Conduction Control Circuit

Table 12.1 Conduction Control Circuit Output by PMDxMDOUT
(When PMDxMDPOT<POLH><POLL> = "00")

PMDxMDOUT <WOC[1:0]> <VOC[1:0]> <UOC[1:0]>	PMDxMDOUT <UPWM>, <VPWM>, <WPWM>			
	0: "High"/"Low" level output		1: PWM output	
	Upper phase output	Lower phase output	Upper phase output	Lower phase output
00	"High" level	"High" level	PWM output	$\overline{\text{PWM}}$ output
01	"High" level	"Low" level	"High" level	$\overline{\text{PWM}}$ output
10	"Low" level	"High" level	$\overline{\text{PWM}}$ output	"High" level
11	"Low" level	"Low" level	$\overline{\text{PWM}}$ output	PWM output

Table 12.2 Conduction Control Circuit Output by PMDxMDOUT
(When PMDxMDPOT<POLH><POLL> = "11")

PMDxMDOUT <WOC[1:0]> <VOC[1:0]> <UOC[1:0]>	PMDxMDOUT <UPWM>, <VPWM>, <WPWM>			
	0: "High"/"Low" level output		1: PWM output	
	Upper phase output	Lower phase output	Upper phase output	Lower phase output
00	"Low" level	"Low" level	$\overline{\text{PWM}}$ output	PWM output
01	"Low" level	"High" level	"Low" level	PWM output
10	"High" level	"Low" level	PWM output	"Low" level
11	"High" level	"High" level	PWM output	$\overline{\text{PWM}}$ output

The detection of one shunt current can be performed by setting as shown below.

Table 12.3 One Shunt Current Detection Settings

Register	Normal	U-phase PWM shift	V-phase PWM shift	W-phase PWM shift
PMDxCMPU	duty_U	<MDPRD[15:0]>-duty_U	duty_U	duty_U
PMDxCMPV	duty_V	duty_V	<MDPRD[15:0]>-duty_V	duty_V
PMDxCMPW	duty_W	duty_W	duty_W	<MDPRD[15:0]>-duty_W
PMDxMDOUT <UOC[1:0]>	11	00	11	11
PMDxMDOUT <VOC[1:0]>	11	11	00	11
PMDxMDOUT <WOC[1:0]>	11	11	11	00

12.3.4.1. PMDxMDPOT (PMD Output Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	POLH	POLL	PSYNCS	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:4	-	R	Read as "0".
3	POLH	R/W	PWM output upper phase polarity select 0: "Low" active 1: "High" active
2	POLL	R/W	PWM output lower polarity select 0: "Low" active 1: "High" active
1:0	PSYNCS[1:0]	R/W	PMDxMDOUT reload timing select 00: PWM counter asynchronous 01: Reload when PWM counter <MDCNT[15:0]> = "0x0001" 10: Reload when PWM counter <MDCNT[15:0]> = PMDxMDPRD<MDPRD[15:0]> 11: Reload when PWM counter <MDCNT[15:0]> = "0x0001" or PMDxMDPRD<MDPRD[15:0]> Selects the reload timing of PMDxMDOUT. PWM output is updated at reload timing. When set this register to "00", the PWM output is updated at PMDxMDOUT updating timing. It is also valid for VEOUTCR1 from VE.

Note: Set this register when PMDxMDEN<PWMEN> is set to "0".

12.3.4.2. PMDxMDOUT (PMD Output Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	WPWM	VPWM	UPWM
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	WOC		VOC		UOC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:11	-	R	Read as "0".
10	WPWM	R/W	U-, V-, and W-phase output control 0: "High"/"Low" level 1: PWM output For details, refer to Table 12.1 and Table 12.2.
9	VPWM	R/W	
8	UPWM	R/W	
7:6	-	R	Read as "0".
5:4	WOC[1:0]	R/W	U-, V-, and W-phase output control For details, refer to Table 12.1 and Table 12.2.
3:2	VOC[1:0]	R/W	
1:0	UOC[1:0]	R/W	

Note1: To load the data written in this register to the subsequent stage of double buffer immediately, set PMDxMODESEL <MDSEL> to "0".

Note2: Do not write to this register by byte transfer command (upper 8 bits ([15:8]) and lower 8 bits ([7:0]) are written separately).

12.3.5. Protection Control Circuit

The protection control circuit consists of an EMG protection control circuit, an OVV protection control circuit, and a port output disable circuit.

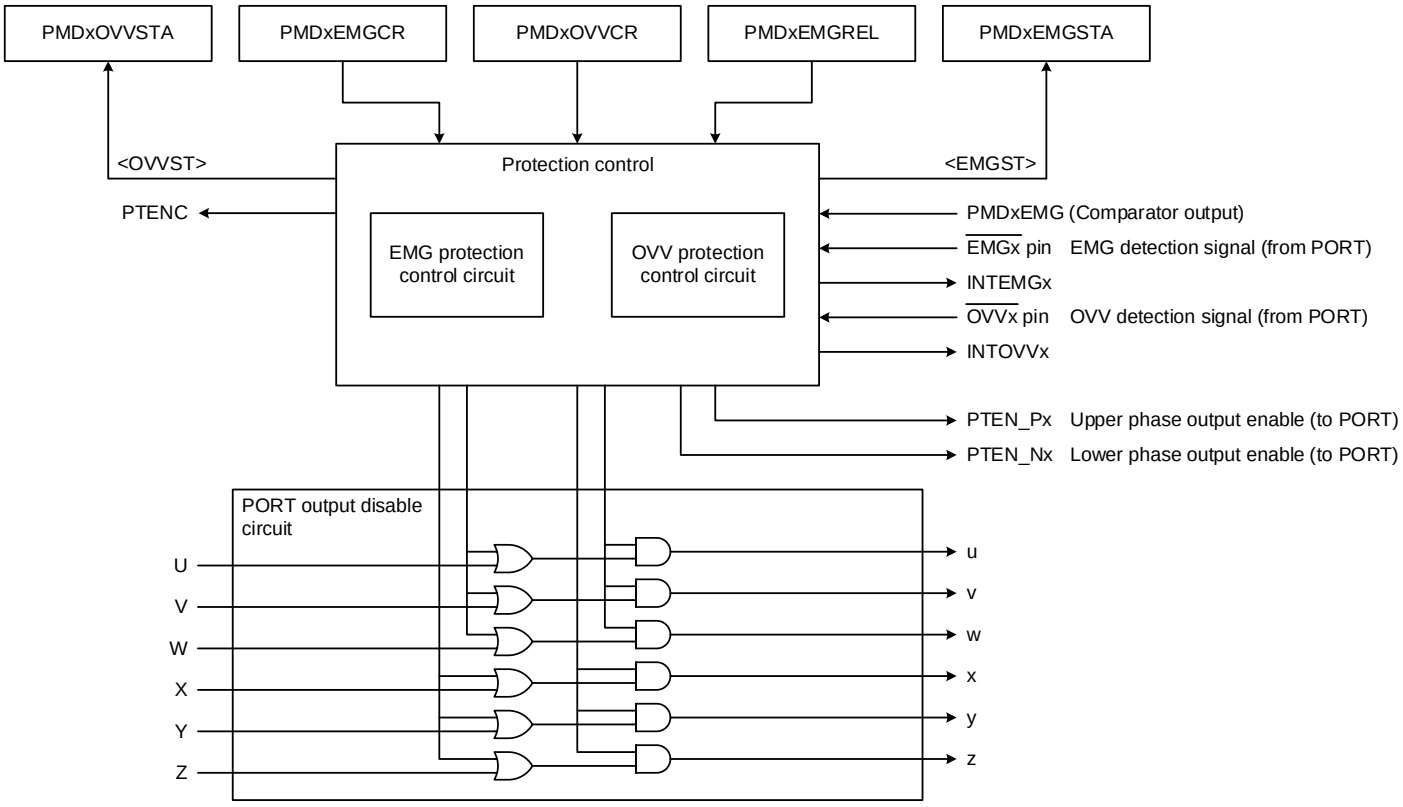


Figure 12.6 Protection Control Circuit

12.3.5.1. EMG Protection Control Circuit

The EMG protection control circuit is a protection circuit for emergency stop. It is used in combination with the port output disable circuit.

The operation of EMG protection control circuit is set by PMDxEMGCR.
When PMDxEMGCR<EMGEN> is set to "1", operation of EMG protection control circuit is enabled.
The PMD stops immediately when EMG input becomes "Low" level. EMG input is selected from $\overline{\text{EMGx}}$ pin input or PMDxEMG (Comparator output).

Note: After the reset is released, EMG protection control circuit is enabled.

- $\overline{\text{EMGx}}$ pin
  A noise filter is inserted into $\overline{\text{EMGx}}$ pin. The duration of noise filtering is specified by PMDxEMGCR <EMGCNT[3:0]>. The noise filter is ignored when PMDxEMGCR<EMGCNT[3:0]>is set to "0000".
- PMDxEMG (Comparator output)
  PMDxEMG is a output signal from comparator in AMP/CMP. Refer to reference manual "19. Op-amplifiers/Analog Comparators (AMP/CMP)" for details.

Note: When PMDxEMGCR<EMGCNT[3:0]> is rewritten while enabling EMG protection control circuit, EMG protection control circuit may be under EMG protection. Therefore, execute "Release under EMG protection" when PMDxEMGCR<EMGCNT[3:0]> is rewritten.

- Operation of the protection control circuit
  When EMG input becomes the "Low" level, EMG protection control circuit operates and is under EMG protection. At this time, a control signal is output to disable UOx, VOx, WOx, XOx, YOx, and ZOx output pins. This will disable each output pin immediately. When each output pin is disabled, the output is set by PMDxEMGCR<EMGMD[1:0]>. It also generates EMG interrupt (INTEMGx).
  When PMDxEMGSTA<EMGST> is read and the read value is "1", it indicates that EMG protection control circuit is under EMG protection.

- Release under EMG protection

  Set all of PMDxMDOUT<WPWM>, <VPWM>, <UPWM>, <WOC[1:0]>, <VOC[1:0]>, and <UOC[1:0]> to "0" to inactivate the UOx, VOx, WOx, XOx, YOx, and ZOx output pins.

  Then, set PMDxEMGCR<EMGRS> to "1" to release under EMG protection.

  When releasing under EMG protection, be sure to read PMDxEMGSTA<EMGI> to confirm by checking that it becomes "1". When PMDxEMGSTA<EMGI> is "1", EMG input is "High" level. When EMG input is "Low" level and the release sequence is executed, it is ignored.

Note: For I/O port which is also used as $\overline{\text{EMGx}}$ pin input, the port function is selected after the reset is released, but EMG protection control circuit is enabled. Therefore, EMG protection control circuit may be under EMG protection. Follow the procedure below to release under EMG protection.

(1) Select EMG function by using the function register (PxFRn) of the port.

(2) Read PMDxEMGSTA<EMGI> and check that it is "1".

(3) Set all of PMDxMDOUT<WPWM>, <VPWM>, <UPWM>, <WOC[1:0]>, <VOC[1:0]>, and <UOC[1:0]> to "0", and all of UOx, VOx, WOx, XOx, YOx, and ZOx output pins are inactive.

(4) Set PMDxEMGCR<EMGRS> to "1" to release under EMG protection.

- Disable EMG protection control circuit operation

  To disable EMG protection control circuit operation, set PMDxEMGREL to "0x5A" → "0xA5". Then set PMDxEMGCR<EMGEN> to "0". To prevent EMG protection control circuit from being accidentally disabled, three instructions must be executed continuously.

12.3.5.2. PMDxEMGREL (EMG Release Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	EMGREL							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:0	EMGREL[7:0]	W	EMG protection control circuit/OVV protection control circuit disable code Setting "0x5A" "→ "0xA5" disables EMG protection control circuit and OVV protection control circuit. After writing the disable code, set PMDxEMGCR<EMGEN> or PMDxEMGCR<OVVEN> to "0".

Note: Writing a disable code is required when EMG protection control circuit is disabled and when OVV protection control circuit is disabled.

12.3.5.3. PMDxEMGCR (EMG Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	EMGCNT			
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	INHEN	EMGMD		EMGISEL	EMGRS	EMGEN
After reset	0	0	1	1	1	0	0	1

Bit	Bit Symbol	Type	Function
31:12	-	R	Read as "0".
11:8	EMGCNT[3:0]	R/W	EMG input detection time "0000" to "1111" (When set to "0000", the noise filter is turned through.) Sets the noise cancel time for $\overline{\text{EMGx}}$ pin input. The noise cancel removal time is expressed by the following formula. PMDxEMGCR<EMGCNT[3:0]>× 16 / fsys (resolution 200ns@fsys = 80MHz)
7:6	-	R	Read as "0".
5	INHEN	R/W	PMD in halt enable/disable 0: Disabled 1: Enabled (initial state is enabled) Selects whether to stop the PMD in halt.
4:3	EMGMD[1:0]	R/W	EMG protection mode selection 00: All phases "Hi-Z" 01: All upper phases ON/All lower phases "Hi-Z" 10: All upper phases "Hi-Z"/All lower phases ON 11: All phases "Hi-Z" Set the output of the upper phase (UOx, VOx, and WOx pins) and the lower phase (XOx, YOx, and ZOx pins) when EMG protection control circuit operates. "ON" indicates PWM output.
2	EMGISEL	R/W	EMG input selection 0: EMGx pin input 1: PMDxEMG (Comparator output) EMG input which is input to the protection circuit is selected from $\overline{\text{EMGx}}$ pin input or PMDxEMG (Comparator output).
1	EMGRS	R	Read as "0".
		W	Releasing under EMG protection 0: Don't care 1: Release under EMG protection Set all of PMDxMDOUT<WPWM>, <VPWM>, <UPWM>, <WOC[1:0]>, <VOC[1:0]>, and <UOC[1:0]> to "0". After confirm that PMDxEMGSTA<EMGI> is "1", set PMDxEMGCR<EMGRS> to "1" to release under EMG protection.
0	EMGEN	R/W	EMG protection control circuit control 0: Disabled 1: Enabled To disable, set PMDxEMGREL to "0x5A" → "0xA5". After that, set PMDxEMGCR<EMGEN> to "0". Execute 3 instructions continuously.

12.3.5.4. PMDxEMGSTA (EMG Status Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	EMGI	EMGST
After reset	0	0	0	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	EMGI	R	EMG input status 0: "Low" level input 1: "High" level input By reading, the state of EMG input can be monitored.
0	EMGST	R	EMG protection status 0: Normal operation 1: Under EMG protection By reading, the state of EMG protection can be monitored.

12.3.6. OVV protection control

OVV protection control circuit is a protection circuit for emergency stop. Use in combination with the port output disable circuit.

The operation of OVV protection control circuit is set by PMDxOVVCR.
When PMDxOVVCR<OVVEN> is set to "1", the operation of OVV protection control circuit is enabled. The PMD stops immediately when OVV input becomes to the "Low" level.

Note: After the reset is released, OVV protection control circuit is disabled.

- OVV input

  OVV input can be selected from $\overline{\text{OVVx}}$ pin input and ADC monitoring function signal by PMDxOVVCR <OVVISEL>.

  The ADC monitoring function can disable/enable the monitoring signal by PMDxOVVCR<ADIN0EN> and <ADIN1EN>.

  A noise filter is inserted into $\overline{\text{OVVx}}$ pin. The duration of noise filtering is specified by PMDxOVVCR <OVVCNT[3:0]>. When setting PMDxOVVCR<OVVCNT[3:0]> to "0000", it operates as a "0001" setting. PMDxOVVCR<OVVCNT[3:0]> is enabled only when PMDxOVVCR<OVVISEL> is set to "1".

Note: When PMDxOVVCR<OVVCNT[3:0]> is rewritten while enabling OVV protection control circuit, OVV protection control circuit may be under OVV protection. Therefore, execute "Release under OVV protection" when PMDxOVVCR<OVVCNT[3:0]> is rewritten.

- Operation of OVV protection

  When OVV input changes to the active level, OVV protection control circuit operates and is under OVV protection. At this time, a control signal is output to disable UOx, VOx, WOx, XOx, YOx, and ZOx output pins. This will disable each output pin immediately. When each output pin is disabled, the output is set by PMDxOVVCR<OVVMD[1:0]>. It also generates OVV interrupt (INTOVVx).

  When PMDxOVVSTA<OVVST> is read, and the read value is "1", it indicates that OVV protection control circuit is under OVV protection.

- Release under OVV protection

  Set PMDxOVVCR<OVVRS> to "1" to release under OVV protection.

  When releasing under OVV protection, be sure to read PMDxOVVSTA<OVVI> to confirm by checking that it becomes "1". When PMDxOVVSTA<OVVI> is "1", OVV input is inactive level. When OVV input is at the active level and a release sequence is executed, it is ignored.

  When PMDxOVVCR<OVVRS> is set to "1", OVV protection is released at the timing where the PWM counter and PMDxMDPRD<MDPRD[15:0]> match after OVV input becomes inactive level.

  However, in 0.5 cycle PWM interrupt setting, OVV protection is released at the timing when the PWM counter matches "0x0001" or PMDxMDPRD<MDPRD[15:0]>.

- Disable OVV protection control circuit operation

  To disable OVV protection control circuit operation, set PMDxEMGREL to "0x5A" → "0xA5". Then set PMDxOVVCR<OVVEN> to "0". To prevent OVV protection control circuit from being accidentally disabled, three instructions must be executed continuously.

12.3.6.1. PMDxOVVCR (OVV Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	OVVCNT			
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:12	-	R	Read as "0".
11:8	OVVCNT[3:0]	R/W	OVV input detection time (Note1) "0000" to "1111" (When "0000" is set, it operates as a "0001" setting.) Sets the noise cancel time of OVV input. The noise cancel time is expressed by the following formula. PMDxOVVCR<OVVCNT[3:0]>× 16 / fsys (resolution 200ns@fsys = 80MHz)
7	-	R	Read as "0".
6	ADIN1EN	R/W	ADC monitoring signal 1 input control (Note2) 0: Disabled 1: Enabled This bit enables or disables the monitoring signal from the ADC monitoring function 1. The monitoring signal of the monitoring function 1 of the ADC can be used as OVV input only when it is set to enable and the ADC monitoring signal is selected as OVV input.
6	ADIN0EN	R/W	ADC monitor signal 0 input control (Note2) 0: Disabled 1: Enabled This bit enables or disables the monitoring signal from the ADC monitoring function 0. The monitoring signal of the monitoring function 0 of the ADC can be used as OVV input only when it is set to enable and the ADC monitoring signal is selected as OVV input.
4:3	OVVMD[1:0]	R/W	OVV protection mode selection (Note3) 00: No output control 01: All upper phases ON/All lower phases OFF 10: All upper phases OFF/All lower phases ON 11: All phases OFF Sets the output of the upper phases (UOx, VOx, and WOx pins) and the lower phases (XOx, YOx, and ZOx pins) when OVV protection control circuit operates. "ON" indicates active level output, and "OFF" indicates inactive level output.
2	OVVISEL	R/W	OVV input selection (Note1) 0: $\overline{\text{OVVx}}$ pin 1: ADC monitoring signal Selects OVV input to be input to OVV protection control circuit.
1	OVVRS	R/W	Release under OVV protection 0: Don't care 1: Release under OVV protection After checking that PMDxOVVSTA<OVVI> is "1", setting PMDxOVVCR<OVVRS> to "1" to releases under OVV protection.

0	OVVEN	R/W	Control of OVV protection control circuit 0: Disabled 1: Enabled To disable, set PMDxEMGREL to "0x5A" → "0xA5". After this, set PMDxOVVCR<OVVEN> to "0". Execute 3 instructions continuously.

Note1: Effective only when PMDxOVVCR<OVVISEL> is set to "1".

Note2: For details of the ADC monitoring function, Refer to "11.5.3. AD Conversion Monitoring Function" in the operation description of "12-Bit Analog-to-Digital Converter (ADC)".

Note3: When OVV/EMG protection control circuit operates at the same time, the setting of PMDxEMGCR <EMGMD[1:0]> takes precedence.

12.3.6.2. PMDxOVVSTA (OVV Status Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	OVVI	OVVST
After reset	0	0	0	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	OVVI	R	OVV input status 0: "Low" level input 1: "High" level input By reading, the state of OVV input can be monitored.
0	OVVST	R	OVV protection status 0: Normal operation 1: Under OVV protection By reading, the state of OVV protection can be monitored.

12.3.7. Protection Control When Break Occurs in Debugger

The UOx, VOx, WOx, XOx, YOx, and ZOx output pins are disabled even during PMD protection control when a break occurs in the debugger. Use PMDxPORTMD<PORTMD> to set the output when the output pins are disabled.

12.3.8. Dead time circuit

The dead time circuit consists of a dead time circuit and an output polarity switching circuit.

The dead time circuit delays the on time to avoid short circuiting of the upper phases (UOx, VOx, and WOx pins) and the lower phases (XOx, YOx, and ZOx pins) when the upper and lower phases are reversed.

The delay time can be set with a resolution of 8 / fsys (100ns@fsys = 80MHz) according to PMDxDTR <DTR[7:0]>.

The output polarity switching circuit sets "High" active or "Low" active for the upper phase (UOx, VOx, and WOx pins) and the lower phase (XOx, YOx, and ZOx pins) respectively by PMDxMDPOT<POLH>, <POLL>.

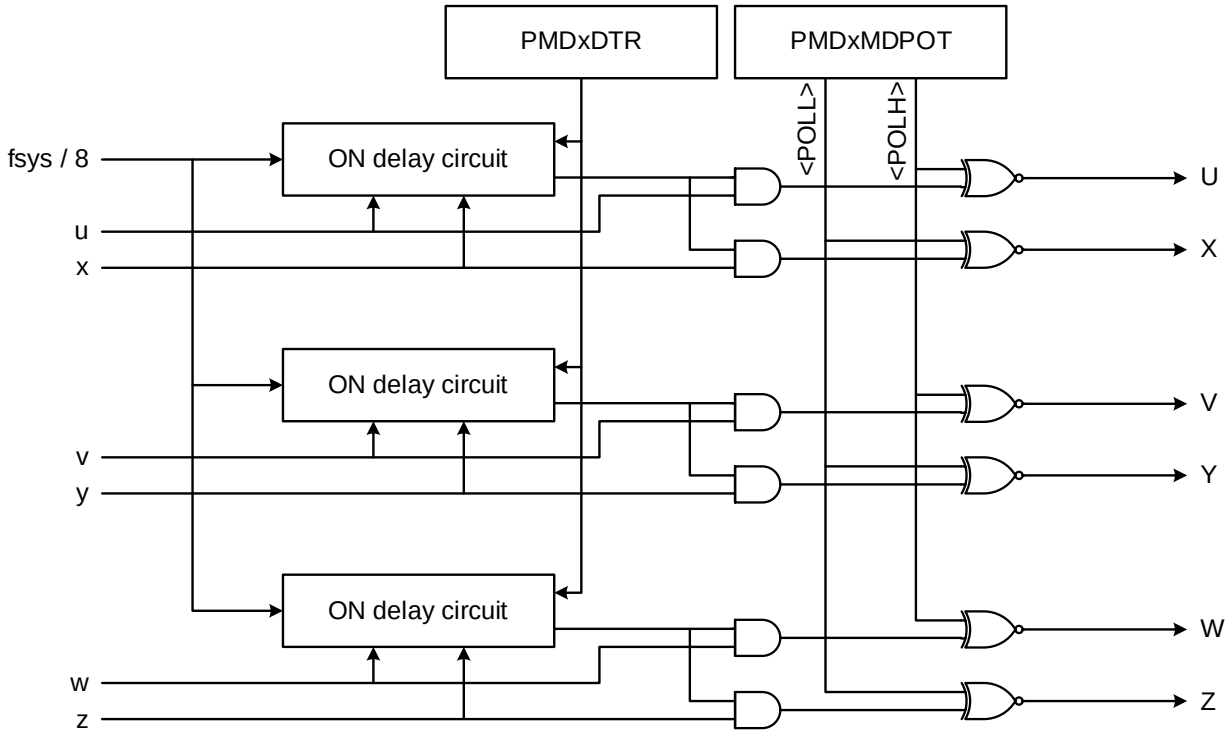


Figure 12.7 Dead Time Circuit

12.3.8.1. PMDxDTR (Dead Time Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	DTR							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:0	DTR[7:0]	R/W	Dead time setting. "0x00" to "0xFF" Dead time is expressed by the following formula. <DTR[7:0]>× 8 / fsys (Up to 25.5μs@fsys = 80MHz)

Note: Change this register when PMDxMDEN<PWMEN> is "0".

12.3.9. Synchronous Trigger Generation Circuit

The synchronous trigger generation circuit generates a trigger signal to sample the ADC in synchronization with the phase-specific PWM carrier.

A trigger signal (TRG0 to TRG3) is generated when PMDxMDCNT<MDCNT[15:0]> matches PMDxTRGCMPn <TRGCMPn[15:0]>.

Select the generation timing from the following.

(1) Match during up counting operation

(2) Match during down counting operation

(3) Match in both up and down counting operations

When edge-aligned mode is selected, the generation timing is up counting.

When PMDxMDEN<PWMEN> is "0", no trigger signal is output.

12.3.9.1. Timing of Updating PMDxTRGCMPn (n = 0 to 3)

PMDxTRGCMPn has a double buffer configuration. When PMDxTRGCR<TRGnBE> is set to "0", the update timing of the subsequent stage of the double buffer is determined by PMDxTRGCR<TRGnMD[2:0]>. When PMDxTRGCR<TRGnBE> is set to "1", the data written to PMDxTRGCMPn is immediately transferred to the subsequent stage of the double buffer.

Table 12.4 Timing of Updating Subsequent Stage of Trigger Compare Register

PMDxTRGCR<TRGnMD[2:0]>	PMDxTRGCMPn<TRGCMPn[15:0]> subsequent stage of double buffer update timing
000: Trigger output disabled	Always update subsequent stage of double buffer.
001: Trigger output at match during down counting	Subsequent stage of double buffer updated at carrier peaks (PMDxCNT<MDCNT[15:0]> matches PMDxMDPRD<MDPRD[15:0]>)
010: Trigger output at match during up counting	Subsequent stage of double buffer updated at carrier bottom (PMDxCNT<MDCNT[15:0]> matches "0x0001")
011: Trigger output at match during up and down counting	Subsequent stage of double buffer updated at carrier peaks and bottoms
100: Trigger output at PWM carrier peak	Always update subsequent stage of double buffer.
101: Trigger output at PWM carrier bottom	
110: Trigger output at PWM carrier peak and bottom	
111: Trigger output disabled	

12.3.9.2. Trigger Output Pattern

When the trigger output mode is set to trigger fixed output mode (PMDxTRGMD<TRGOUT> = "0"), the ADC synchronous trigger signal (PMDxTRGn) is output as shown below.

- When PMDxMDCNT<MDCNT[15:0]> matches PMDxTRGCMPn<TRGCMPn[15:0]>, the signal is output to the ADC synchronous trigger signal (PMDxTRGn) regardless of PMDxTRGSEL<TRGSEL[2:0]>.

When the trigger output mode is set to trigger selected output mode (PMDxTRGMD<TRGOUT>= "1"), the ADC synchronous trigger signal (PMDxTRGn) is output as shown below.

- When PMDxMDCNT<MDCNT[15:0]> matches PMDxTRGCMP0<TRGCMP0[15:0]>, the signal is output to the ADC synchronous trigger signal (PMDxTRGn) selected by PMDxTRGSEL<TRGSEL[2:0]>.

Table 12.5 Trigger Output

PMDxTRGMD <TRGOUT>	PMDxTRGCMPn compare register	PMDxTRGSEL <TRGSEL[2:0]>	ADC synchronous trigger output
0	PMDxTRGCMP0	-	PMDxTRG0
	PMDxTRGCMP1		PMDxTRG1
	PMDxTRGCMP2		PMDxTRG2
	PMDxTRGCMP3		PMDxTRG3
1	PMDxTRGCMP0	000	PMDxTRG0
		001	PMDxTRG1
		010	PMDxTRG2
		011	PMDxTRG3
		100	PMDxTRG4
		101	PMDxTRG5
	PMDxTRGCMP1	-	No trigger output
	PMDxTRGCMP2		No trigger output
	PMDxTRGCMP3		No trigger output

Note: -: Don't care

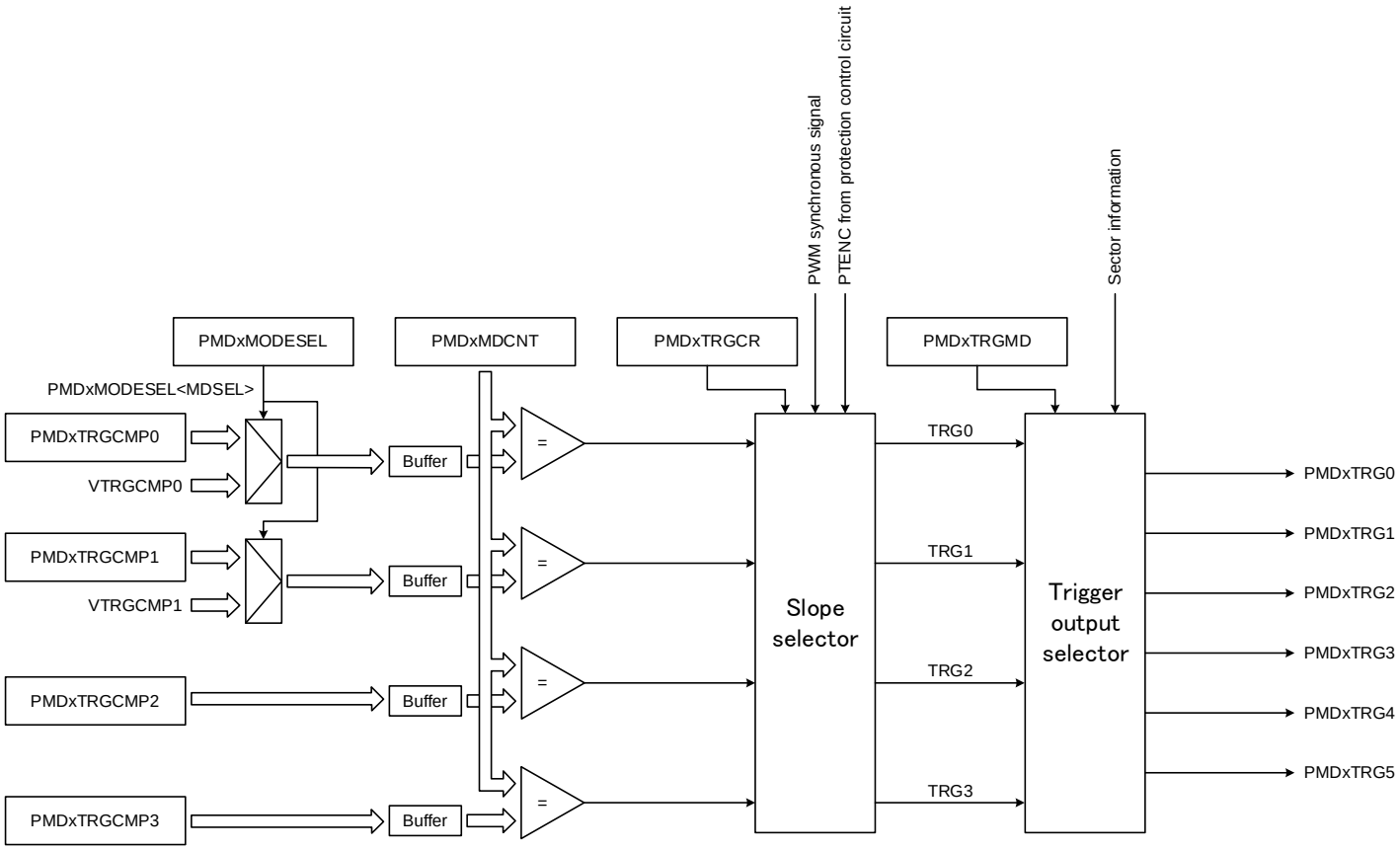


Figure 12.8 Synchronous Trigger Generation Circuit

12.3.9.3. PMDxTRGCMPn (Trigger Compare n Register) (n = 0 to 3)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TRGCMPn							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TRGCMPn							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TRGCMPn[15:0]	R	When reading, the value (data set from the bus) in the first stage of double buffer can be read.
		W	Trigger compare register "0x0001" to PMDxMDPRD<MDPRD[15:0]> - 1 Others: Prohibited A trigger signal TRGn is output when the PWM counter PMDxCNT<MDCNT[15:0]> matches PMDxTRGCMPn<TRGCMPn[15:0]>.

Note1: To load the data written to the subsequent stage of double buffer immediately, set PMDxMODESEL <MDSEL> to "0".

Note2: Do not write to this register by byte transfer command (upper 8 bits ([15:8]) and lower 8 bits ([7:0]) are written separately).

Note3: When PMDxMDCR<PWMMD> is set to "0" (PWM mode 0) and PMDxTRGCMPn<TRGCMPn> is set to "0x0001", the trigger signal TRGn is not output for the first cycle after PMDxMDEN<PWMEN> is set to "1".

12.3.9.4. PMDxTRGCR (Trigger Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15	TRG3BE	R/W	Subsequent stage of double buffer for PMDxTRGCMP3 update timing selection (Note 1) 0: Synchronous update (Note 2) 1: Asynchronous update (transferred immediately after writing)
14:12	TRG3MD[2:0]	R/W	PMDxTRGCMP3 mode setting 000: Trigger output disabled 001: Trigger output at match during down counting 010: Trigger output at match during up counting 011: Trigger output at match during up and down counting 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak and bottom 111: Trigger output disabled Selects the matching mode for the trigger output TRG3. When PWM mode 0 (sawtooth wave) is selected by PMDxMDCR<PMDMD>, TRG3 is output at match during up counting or PWM carrier peak even if down counting or PWM bottom are selected. When PMDxTRGCMP3<TRGCMP3[15:0]> is set to "0x0001", and mode 1 (triangle wave) is selected by PMDxMDCR<PWMMD>, setting PMDxTRGCR<TRG3MD[2:0]> to "011" makes the trigger output once a cycle.
11	TRG2BE	R/W	Subsequent stage of double buffer for PMDxTRGCMP2 update timing selection (Note 1) 0: Synchronous update (Note 2) 1: Asynchronous update (transferred immediately after writing)
10:8	TRG2MD[2:0]	R/W	PMDxTRGCMP2 mode setting 000: Trigger output disabled 001: Trigger output at match during down counting 010: Trigger output at match during up counting 011: Trigger output at match during up and down counting 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak and bottom 111: Trigger output disabled Selects the matching mode for the trigger output TRG2. When PWM mode 0 (sawtooth wave) is selected by PMDxMDCR<PMDMD>, TRG2 is output at match during up counting or PWM carrier peak even if down counting or PWM bottom are selected. When PMDxTRGCMP2<TRGCMP2[15:0]> is set to "0x0001", and mode 1 (triangle wave) is selected by PMDxMDCR<PWMMD>, setting PMDxTRGCR<TRG2MD[2:0]> to "011" makes the trigger output once a cycle.

7	TRG1BE	R/W	Subsequent stage of double buffer for PMDxTRGCMP1 update timing selection (Note 1) 0: Synchronous update (Note 2) 1: Asynchronous update (transferred immediately after writing)
6:4	TRG1MD[2:0]	R/W	PMDxTRGCMP1 mode setting 000: Trigger output disabled 001: Trigger output at match during down counting 010: Trigger output at match during up counting 011: Trigger output at match during up and down counting 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak and bottom 111: Trigger output disabled Selects the matching mode for the trigger output TRG1. When PWM mode 0 (sawtooth wave) is selected by PMDxMDCR<PMDMD>, TRG1 is output at match during up counting or PWM carrier peak even if down counting or PWM bottom are selected. When PMDxTRGCMP1<TRGCMP1[15:0]> is set to "0x0001", and mode 1 (triangle wave) is selected by PMDxMDCR<PWMMD>, setting PMDxTRGCR<TRG1MD[2:0]> to "011" makes the trigger output once a cycle.
3	TRG0BE	R/W	Subsequent stage of double buffer for PMDxTRGCMP0 update timing selection (Note 1) 0: Synchronous update (Note 2) 1: Asynchronous update (transferred immediately after writing)
2:0	TRG0MD[2:0]	R/W	PMDxTRGCMP0 mode setting 000: Trigger output disabled 001: Trigger output at match during down counting 010: Trigger output at match during up counting 011: Trigger output at match during up and down counting 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak and bottom 111: Trigger output disabled Selects the matching mode for the trigger output TRG0. When PWM mode 0 (sawtooth wave) is selected by PMDxMDCR<PMDMD>, TRG0 is output at match during up counting or PWM carrier peak even if down counting or PWM bottom are selected. When PMDxTRGCMP0<TRGCMP0[15:0]> is set to "0x0001", and mode 1 (triangle wave) is selected by PMDxMDCR<PWMMD>, setting PMDxTRGCR<TRG0MD[2:0]> to "011" makes the trigger output once a cycle.

Note1: When PMDxMDEN<PWMEN> is "0", the data is updated asynchronously regardless of the setting.

Note2: For the update timing, refer to "Table 12.4 Timing of Updating Subsequent Stage of Trigger Compare Register".

12.3.9.5. PMDxTRGMD (Trigger Output Mode Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	TRGOUT	EMGTGE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	TRGOUT	R/W	Trigger output mode 0: Trigger fixed output mode 1: Trigger selected output mode In the trigger fixed output mode, each trigger output PMDxTRG0 to PMDxTRG3 outputs a trigger signal when the PWM counter PMDxCNT<MDCNT[15:0]> matches PMDxTRGCMP0<TRGCMP0> to PMDxTRGCMP3<TRGCMP3>. PMDxTRG4 and PMDxTRG5 are not outputted. In the trigger selected output mode, the trigger signal which is output when the PWM counter PMDxCNT<MDCNT[15:0]> matches PMDxTRGCMP0<TRGCMP0> is output to any of the trigger output PMDxTRG0 to PMDxTRG5. Trigger out is selected by PMDxTRGSEL<TRGSEL [2:0]>. (Note)
0	EMGTGE	R/W	Output enable setting during EMG protection operation 0: Trigger output disabled during protection operation 1: Trigger output enabled during protection operation

Note: For the trigger output when PMDxTRGMD<TRGOUT> is set to "1", refer to "Table 12.5 Trigger Output".

12.3.9.6. PMDxTRGSEL (Trigger Output Select Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	TRGSEL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:3	-	R	Read as "0".
2:0	TRGSEL[2:0]	R/W	Trigger output selection 000: Output from PMDxTRG0 001: Output from PMDxTRG1 010: Output from PMDxTRG2 011: Output from PMDxTRG3 100: Output from PMDxTRG4 101: Output from PMDxTRG5 110: No trigger output 111: No trigger output This register is enabled when the trigger mode is trigger selected output mode (PMDxTRGMD <TRGOUT> = "1"). Selects the output destination of the trigger signal generated when the PWM counter PMDxCNT<MDCNT[15:0]> matches PMDxTRGCMP0<TRGCMP0>. (refer to "Table 12.5 Trigger Output".)

13. Vector Engine (VE)

13.1. Outline

13.1.1. Features

(1) The basic tasks for vector control (coordinate-axis conversion, phase conversion, and SIN/COS computation) use fixed-point format data.
→No complicated decimal point position management is required for software processing.

(2) Built in interface (output control, trigger generation, input processing) to control PMD and ADC

- Converts the computation result in fixed-point format to the data format of PMD.
- Generates timing data for interactive operation
- Converts AD conversion result to data in fixed-point format

(3) Calculates current, voltage, and rotation speed with normalized values based on their respective maximum values.
In this case, the decimal is in fixed-point format

(4) Built in PI control in current control

(5) Built in phase interpolation that integrates rotation speed

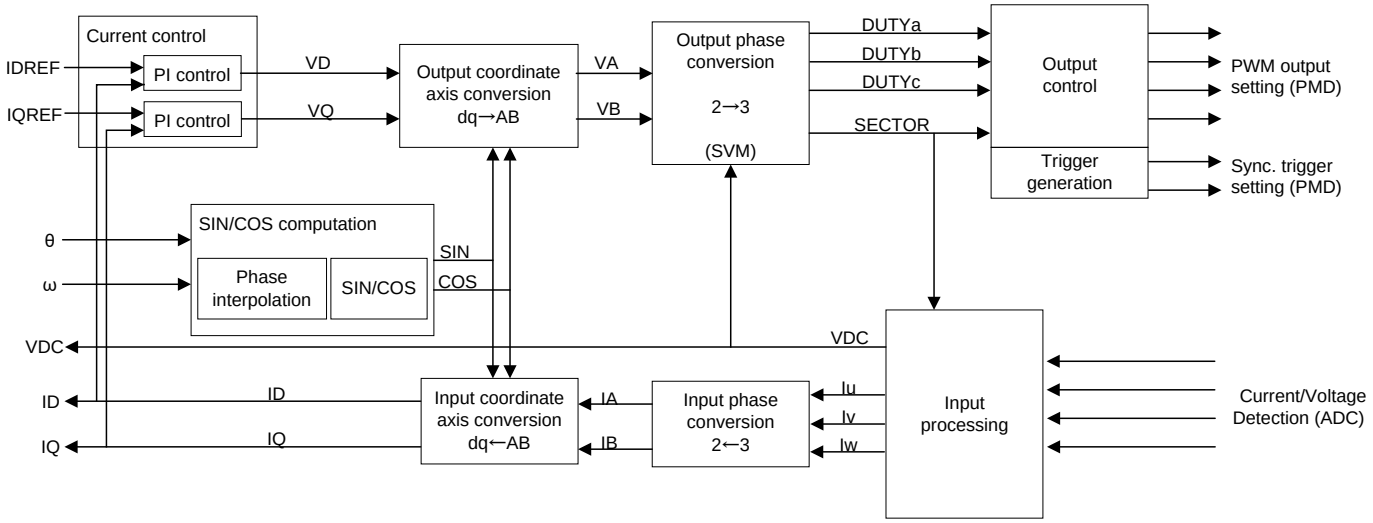


Figure 13.1 Block Diagram of Vector Control

13.1.2. Main Functions

(1) Space vector conversion is used for phase conversion from 2-phase to 3-phase. The conversion method supports 2-phase modulation and 3-phase modulation.

(2) In trigger generation, the sampling timing of the ADC corresponding to the sensorless current detection method can be generated. The current detection supports the 1-shunt method, 3-shunt method, and 2-sensor method.

(3) In current control, PI control is implemented independently for d-axis and q-axis. It is also possible to set voltage reference value directly without using current control processing.

(4) SIN/COS computations are performed with approximations using series expansion. For phase information, direct setting and phase interpolation that integrates with the PWM cycle from the rotation speed are possible.

Note 1: For using VE , the PMD and ADC must be set. For the PMD setting, select the VE mode using the mode select register (PMDxMODESEL).

Note 2: For the ADC setting, set the program (trigger enable, AIN selection, result register selection) for each synchronous trigger from the PMD.

13.2. Block Diagram

The configuration of VE is shown in Figure 13.2.

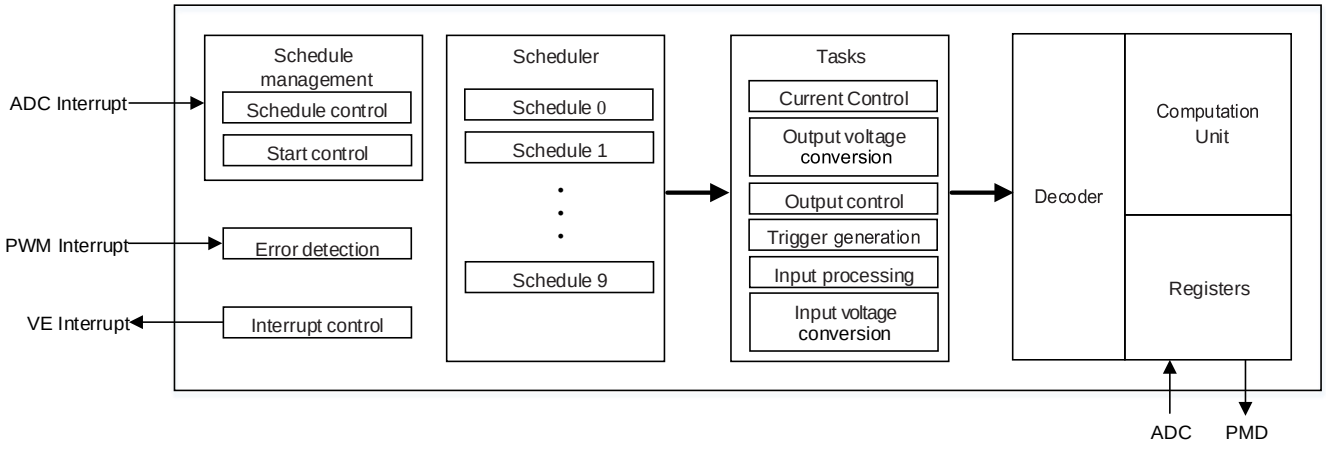


Figure 13.2 Configuration of VE

13.2.1. VE, PMD, and ADC Relationships

As shown in Figure 13.3, the VE allows direct interaction with the PMD and ADC.

When PMDxMODESEL is set to the VE mode, the PMD register PMDxCMPU, PMDxCMPV, PMDxCMPW, PMDxMDOUT, PMDxTRGCMP0, PMDxTRGCMP1 and PMDxTRGSEL switched to the VE registers VECMPUx, VECMPVx, VECMPWx, VEOUTCRx, VETRGCMP0x, VETRGCMP1x, VETRGSELx respectively. In this case, only data in these VE registers can control the motor, and the data in PMD registers which are written by the CPU cannot control. Other PMD registers have no read/write restrictions.

The ADC registers ADxREG0, ADxREG1, ADxREG2, ADxREG3 and ADxPSETn<UVWISn0[1:0]>, <UVWISn1[1:0]>, <UVWISn2[1:0]>, and <UVWISn3[1:0]> which are read from the VE as the VE registers VEADREG0x, VEADREG1x, VEADREG2x, VEADREG3x, VEPHNUM0x, VEPHNUM1x, VEPHNUM2x and VEPHNUM3x, respectively (These registers are dedicated that cannot be accessed by the CPU). These ADC registers can be written and read from the CPU.

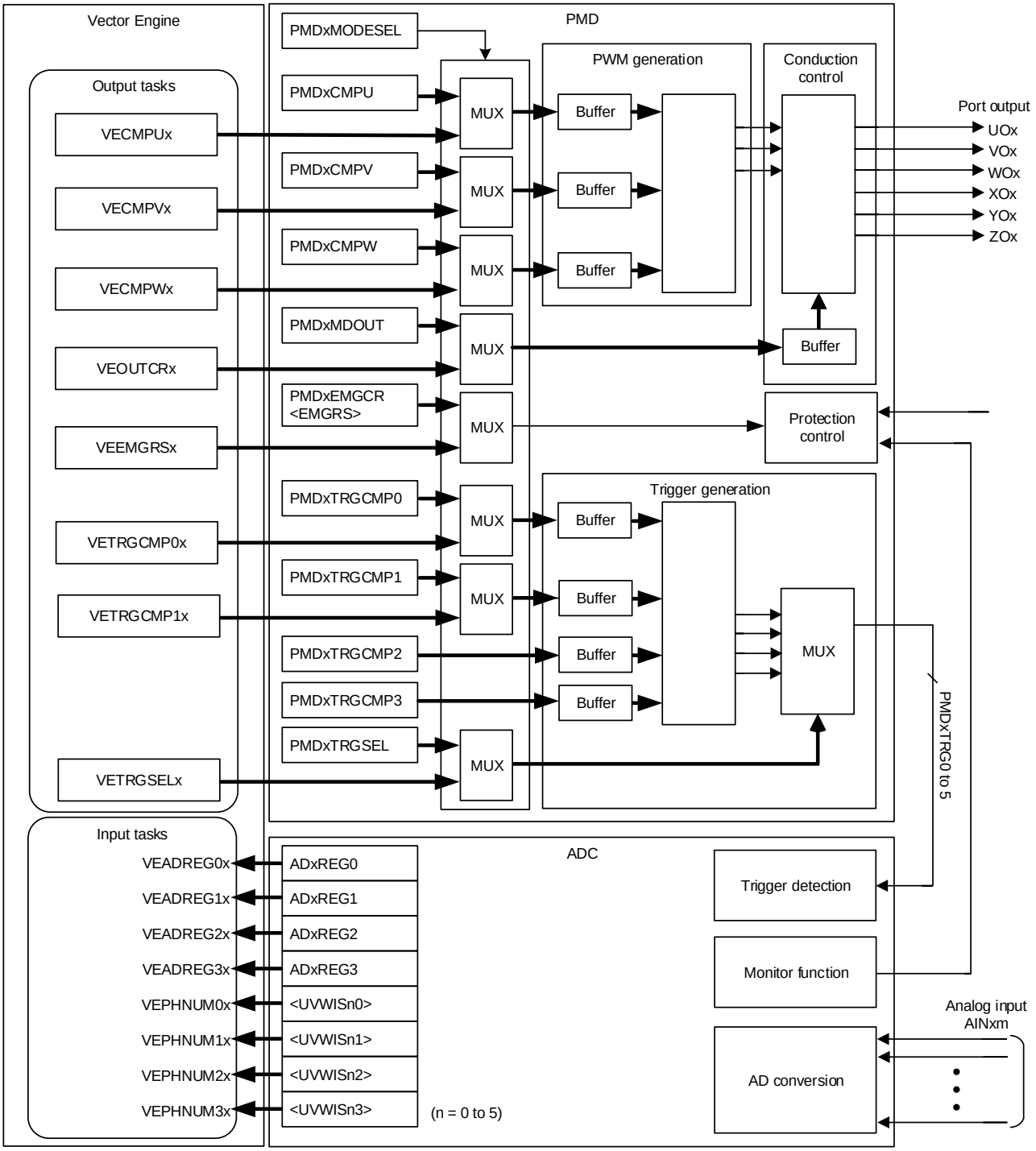


Figure 13.3 Interaction among VE, PMD and ADC

13.3. Registers

The VE registers are divided into the following three types.

- VE control registers
  VE control registers and temporary registers
- Common registers
  Registers commonly used for channel
- Dedicated registers
  Computation data and control registers for channel

13.3.1. List of Registers

13.3.1.1. VE Control Register

Register name		Address (Base+)
VE Operation Enable/Disable Register	VEEN	0x0000
CPU Start Trigger Selection Register	VECPURUNTRG	0x0004
Task Specification Register	VETASKAPP	0x0008
Operation Schedule Selection Register	VEACTSCH	0x000C
Operation Schedule Repetition Count Specification Register	VEREPTIME	0x0010
Start Trigger Mode Setting Register	VETRGMODE	0x0014
Error Interrupt Enable/Disable Setting Register	VEERRINTEN	0x0018
VE Forced Termination Register	VECOMPEND	0x001C
Error Detection Register	VEERRDET	0x0020
Schedule Execution Status/Executing Task Register	VESCHTASKRUN	0x0024
Reserved	-	0x0028
Temporary 0 Register	VETMPREG0	0x002C
Temporary 1 Register	VETMPREG1	0x0030
Temporary 2 Register	VETMPREG2	0x0034
Temporary 3 Register	VETMPREG3	0x0038
Temporary 4 Register	VETMPREG4	0x003C
Temporary 5 Register	VETMPREG5	0x0040
Reserved	-	0x01BC

Note: Do not access addresses with "Reserved".

13.3.1.2. Common Register

Register name		Address (Base+)
Reserved	-	0x0174
ADC-Conversion Time Setting (based on PWM clock) Register	VETADC	0x0178

Note: Do not access addresses with "Reserved".

13.3.1.3. Specific Registers

Register name		Address (Base+)
Status Flag Register	VEMCTLFx	0x0000
Task Control Mode Register	VEMODEx	0x0004
Flow Control Register	VEFMODEx	0x0008
PWM Cycle Rate (PWM cycle [s] × maximum speed (Note1) × 2^{16}) Setting Register	VETPWMx	0x000C
Rotation Speed (speed [Hz] / maximum speed (Note1) × 2^{15}) Setting Register	VEOMEGAx	0x0010
Motor Phase (motor phase [deg] / 360 × 2^{16}) Setting Register	VETHETAx	0x0014
d-Axis Reference Current Value (current [A] / maximum current (Note2) × 2^{15}) Setting Register	VEIDREFx	0x0018
q-Axis Reference Current Value (current [A] / maximum current (Note 2) × 2^{15}) Setting Register	VEIQREFx	0x001C
d-Axis Voltage (voltage [V] / maximum voltage (Note3) × 2^{31}) Setting Register	VEVDx	0x0020
q-Axis Voltage (voltage [V] / maximum voltage (Note3) × 2^{31}) Setting Register	VEVQx	0x0024
d-Axis Current Control Integral Coefficient for PI Control Setting Register	VECIDKIx	0x0028
d-Axis Current Control Proportional Coefficient for PI Control Setting Register	VECIDKPx	0x002C
q-Axis Current Control Integral Coefficient for PI Control Setting Register	VECIQKIx	0x0030
q-Axis Current Control Proportional Coefficient for PI Control Setting Register	VECIQKPx	0x0034
d-Axis Voltage Integral Term Hold (upper 32 bits of VDI) Register	VEVDIHx	0x0038
d-Axis Voltage Integral Term Hold (lower 32 bits of VDI) Register	VEVDILHx	0x003C
q-Axis Voltage Integral Term Hold (upper 32 bits of VQI) Register	VEVQIHx	0x0040
q-Axis Voltage Integral Term Hold (lower 32 bits of VQI) Register	VEVQILHx	0x0044
Rotation Speed When Shift PWM is Enabled with 2-phase Modulation Register	VEFPWMCHGx	0x0048
PWM Cycle Setting (setting the value as same as PMD's PWM cycle) Register	VEMDPRDx	0x004C
Minimum Pulse Width Difference Setting Register	VEMINPLSx	0x0050
Synchronous Trigger Correction Value Setting Register	VETRGCRCx	0x0054
Reserved	–	0x0058

Register name		Address (Base+)
Cosine Value by THETA for Output Conversion (Q15 data) Register	VECOSx	0x005C
Sine Value by THETA for Output Conversion (Q15 data) Register	VESINx	0x0060
Previous Cosine Value for Input Processing (Q15 data) Register	VECOSMx	0x0064
Previous Sine Value for Input Processing (Q15 data) Register	VESINMx	0x0068
Sector Information Register	VESECTORx	0x006C
Previous Sector Information Register	VESECTORMx	0x0070
AD Conversion Result at Detecting a-Phase Zero Current Register (Note4)	VEIAOx	0x0074
AD Conversion Result at Detecting b-Phase Zero Current Register (Note4)	VEIBOx	0x0078
AD Conversion Result at Detecting c-Phase Zero Current Register (Note4)	VEICOx	0x007C
a-Phase Current AD Conversion Result Register (Note4)	VEIAADCx	0x0080
b-Phase Current AD Conversion Result Register (Note4)	VEIBADCx	0x0084
c-Phase Current AD Conversion Result Register (Note4)	VEICADCx	0x0088
DC Supply Voltage (voltage [V] / maximum voltage (Note3) × 2^{15}) Register	VEVDCx	0x008C
d-Axis Current (current [A] / maximum current (Note2) × 2^{31}) Register	VEIDx	0x0090
q-Axis Current (current [A] / maximum current (Note2) × 2^{31}) Register	VEIQx	0x0094
PMD Control: U-Phase PWM Pulse Width Setting Register	VECMPUx	0x00C0
PMD Control: V-Phase PWM Pulse Width Setting Register	VECMPVx	0x00C4
PMD Control: W-Phase PWM Pulse Width Setting Register	VECMPWx	0x00C8
PMD Control: 6-Phase Output Control Register	VEOUTCRx	0x00CC
PMD Control: Trigger Timing Setting (TRGCMP0) Register	VETRGCMP0x	0x00D0
PMD Control: Trigger Timing Setting (TRGCMP1) Register	VETRGCMP1x	0x00D4
PMD Control: Synchronous Trigger Setting Register	VETRGSELx	0x00D8
PMD Control: EMG Return Setting Register	VEEMGRSx	0x00DC

Note 1: Maximum speed: Maximum rotation speed [Hz] that can be controlled or operation.

Note 2: Maximum current: The current value [A] when it becomes "0x7FF0"after the AD conversion result of the phase current corrected to zero level.

Note 3: Maximum voltage: Voltage value [V] when the AD conversion result of the supply voltage (VDC) becomes "0xFFF0".

Note 4: AD conversion result is stored in the upper 12 bits of 16-bit register.

Note 5: Do not access the address in "Reserved".

13.3.2. VE Control Register

13.3.2.1. VEEN (VE Operation Enable/Disable Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	VEIDLEN	VEEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	VEIDLEN	R/W	Clock operation control in IDLE mode 0: Stop 1: Operation
0	VEEN	R/W	VE operation control 0: Disabled 1: Enabled

13.3.2.2. VECPURUNTRG (CPU Start Trigger Selection Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	VCPURTB	VCPURTA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	VCPURTB	R	Read as "0".
		W	Starts VE ch1 by software (Note1) 0: - 1: Operation start
0	VCPURTA	R	Read as "0".
		W	Starts VE ch0 by software (Note1) 0: - 1: Operation start

Note1: When this bit is written to "1", it is cleared in the next cycle.

Note2: The task to be started is determined by the settings of the VEACTSCH and VETASKAPP.

Note3: When task of channel under executing schedule is restarted, it must be restarted by VECPURUNTRG after it must be terminated by VECOMPEND.

13.3.2.3. VETASKAPP (Task Specification Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VTASKB				VTASKA			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:4	VTASKB[3:0]	R/W	Specifies the started task of VE. 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 to 0xF: Prohibited Specifies the task to be started when VE ch1 is started by software.
3:0	VTASKB[3:0]	R/W	Specifies the started task of VE. 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 to 0xF: Prohibited Specifies the task to be started when VE ch0 is started by software.

Note: Only those tasks that are included in operating schedules can be specified.

13.3.2.4. VEACTSCH (Operation Schedule Selection Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VACTB				VACTA			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:4	VACTB[3:0]	R/W	Specify the operation schedule of VE ch1. 0x0: Task execution 0x1: Schedule 1 0x4: Schedule 4 0x9: Schedule 9 Others: Prohibited
3:0	VACTA[3:0]	R/W	Specify the operation schedule of VE ch0. 0x0: Task execution 0x1: Schedule 1 0x4: Schedule 4 0x9: Schedule 9 Others: Prohibited

13.3.2.5. VEREPTIME (Operation Schedule Repetition Count Specification Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VREPB				VREPA			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:4	VREPB[3:0]	R/W	Specifying the repetition number of the VE ch1 operation schedule (Note) 0: Do not execute the schedule 1 to 15: Execute schedule a specified number of times
3:0	VREPA[3:0]	R/W	Specifying the repetition number of the VE ch0 operation schedule (Note) 0: Do not execute the schedule 1 to 15: Execute schedule a specified number of times

Note: When this register is set to "0", no schedule is executed.

13.3.2.6. VETRGMODE (Start Trigger Mode Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	VTRGB		VTRGA	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:4	-	R	Read as "0".
3:2	VTRGB[1:0]	R/W	Trigger mode of VE ch1 00: Ignore both INTADAPDB (ADC unit A, PMD ch1 trigger conversion completion interrupt request) and INTADBPDB (ADC unit B, PMD ch1 trigger conversion completion interrupt request). 01: Start by INTADAPDB (ADC unit A, PMD ch1 trigger conversion completion interrupt request). 10: Start by INTADBPDB (ADC unit B, PMD ch1 trigger conversion completion interrupt request). 11: Start when both INTADAPDB (ADC unit A, PMD ch1 trigger conversion completion interrupt request) and INTADBPDB (ADC unit B, PMD ch1 trigger conversion completion interrupt request) occur.
1:0	VTRGA[1:0]	R/W	Trigger mode of VE ch0 00: Ignore both INTADAPDA (ADC unit A, PMD ch0 trigger conversion completion interrupt request) and INTADBPDA (ADC unit B, PMD ch0 trigger conversion completion interrupt request). 01: Start by INTADAPDA (ADC unit A, PMD ch0 trigger conversion completion interrupt request). 10: Start by INTADBPDA (ADC unit B, PMD ch0 trigger conversion completion interrupt request). 11: Start when both INTADAPDA (ADC unit A, PMD ch0 trigger conversion completion interrupt request) and INTADBPDA (ADC unit B, PMD ch9 trigger conversion completion interrupt request) occur.

13.3.2.7. VEERRINTEN (Error Interrupt Enable/Disable Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	VERRENB	VERRENA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	VERRENB	R/W	Interrupt control when VE ch1 error is detected 0: Disabled 1: Enabled
0	VERRENA	R/W	Interrupt control when VE ch0 error is detected 0: Disabled 1: Enabled

13.3.2.8. VECOMPEND (VE Forced Termination Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	VCENDB	VCENDA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	VCENDB	R	Read as "0".
		W	Forcefully terminates the schedule in operating VE ch1. (Note) 0: - 1: Terminated
0	VCENDA	R	Read as "0".
		W	Forcefully terminates the schedule in operating VE ch0. (Note) 0: - 1: Terminated

Note: When this register is set to "1", it is cleared in the next cycle.

13.3.2.9. VEERRDET (Error Detection Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	VERRDB	VERRDA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	VERRDB	R	VE ch1 Error Flag 0: No error detected 1: Error detected When a PWM interrupt is detected during execution of a schedule (excluding wait for a start trigger), the error flag is set to "1".
0	VERRDA	R	VE ch0 Error Flag 0: No error detected 1: Error detected When a PWM interrupt is detected during execution of a schedule (excluding wait for a start trigger), the error flag is set to "1".

Note: The error flag is cleared when the register is read.

13.3.2.10. VESCHTASKRUN (Schedule Execution Status/Executing Task Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	VRTASKB	
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VRTASKB		VRSCHB	VRTASKA				VRSCHA
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:10	-	R	Read as "0".
9:6	VRTASKB[3:0]	R	Task number currently executing in VE ch1 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 to 0xF: Prohibited
5	VRSCHB	R	Schedule execution status in VE ch1 0: Stop 1: Executing
4:1	VRTASKA[3:0]	R	Task number currently executing in VE ch0 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 to 0xF: Prohibited
0	VRSCHA	R	Schedule execution status in VE ch0 0: Stop 1: Executing

13.3.2.11. VETMPREGn (Temporary n Register) (n = 0 to 5)

	31	30	29	28	27	26	25	24
Bit symbol	TMPREGn							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	TMPREGn							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TMPREGn							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TMPREGn							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	TMPREGn[31:0]	R/W	Temporary n register

13.3.3. Common Register

13.3.3.1. VETADC (AD Conversion Time Setting (based on PWM clock) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TADC							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TADC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TADC[15:0]	R/W	AD conversion time "0x0000" to "0xFFFF": (Set the AD conversion time [s] / PWM counter clock cycle [s])

Note: This register is effective when the 1-shunt current detection mode is selected and shift PWM is enabled.

13.3.4. Specific Registers

13.3.4.1. VEMCTLFx (Status Flag Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	PLSLFM	PLSLF	-	LVTF	LAVFM	LAVF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:6	-	R/W	Write as "0".
5	PLSLFM	R/W	Previous value of <PLSLF>
4	PLSLF	R/W	Pulse with minimum flag 0: Minimum pulse width difference ≥ VEMINPLSx<MINPLS> 1: Minimum pulse width difference < VEMINPLSx<MINPLS >
3	-	R/W	Write as "0".
2	LVTF	R/W	Low supply voltage flag 0: VEVDCx<VDC> ≥ 1/128 1: VEVDCx<VDC> < 1/128
1	LAVFM	R/W	Previous value of VEMCTLFx<LAVF>
0	LAVF	R/W	Low speed Flag 0: VEOMEGAx<OMEGA> ≥ VEFPWMCHGx<FPWMCHG > (high speed) 1: VEOMEGAx<OMEGA> < VEFPWMCHGx<FPWMCHG > (low speed)

13.3.4.2. VEMODEx (Task Control Mode Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	OCRMD		ZIEN	PVIEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:4	-	R/W	Write as "0".
3:2	OCRMD[1:0]	R/W	Output control operation 00: Output off 01: Output enable 10: Prohibited 11: EMG return
1	ZIEN	R/W	Zero current detection control 0: Disabled 1: Enabled
0	PVIEN	R/W	Phase interpolation Control 0: Disabled 1: Enabled

13.3.4.3. VEFMODEx (Flow Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	MREGDIS	CRCEN
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	ADCSEL	-	PMDSEL	ADCSEL	IDMODE		SPWMEN	C2PEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:10	-	R/W	Write as "0".
9	MREGDIS	R/W	Keeping previous value of SIN/COS/SECTOR selection 0: Enabled 1: Disabled When disabled, VESINMx = VESINx, VECOSMx = VECOSx, VESECTORMx = VESECTORx
8	CRCEN	R/W	Trigger correction enable 0: Disabled 1: Enabled
7:6	ADCSEL	R/W	Used ADC unit selection 00: Unit A 01: Unit B 10: Unit A, B 11: Unit A, B ADC unit must be selected shown below by VE channel. VE ch0: Used ADC unit A or A, B VE ch1: Used ADC unit B or A, B
5	-	R/W	Write as "0".
4	PMDSEL	R/W	Used PMD channel selection 0: Channel 0 1: Channel 1 PMD channel must be selected shown below by VE channel. VE ch0: Used PMD channel channel 0 VE ch1: Used PMD channel channel 1
3:2	IDMODE	R/W	Current detection mode 00: 3 shunts 01: 2 sensors 10: 1 shunt (PMDTRG up counter (Note)) 11: 1 shunt (PMDTRG down counter (Note))
1	SPWMEN	R/W	Shift PWM enable 0: Disabled 1: Enabled
0	C2PEN	R/W	Modulation mode selection 0: 3-phase modulation 1: 2-phase modulation

Note: When the 1-shunt mode is used, refer to below for PMDTRG setting value.

VEFMODEx <IDMODE[1:0]>	PMDxTRGCR <TRG0MD[2:0]>	PMDxTRGCR <TRG1MD[2:0]>
10	010 (up count)	010 (up count)
10	101 (carrier bottom)	010 (up count)
11	001 (down count)	001 (down count)
11	001 (down count)	101 (carrier bottom)

13.3.4.4. VETPWMx (PWM Cycle Rate (PWM cycle [s] × maximum speed × 2^{16}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TPWM							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TPWM							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TPWM[15:0]	R/W	Set the PWM cycle rate and integration unit during phase interpolation. 16-bit fixed point data "0.0" to "1.0". "0x0000" to "0xFFFF": PWM cycle [s] × Max_Hz × 2^{16} Indicates the ratio between the PWM frequency and the maximum rotation speed. (Max_Hz: Maximum rotation speed)

13.3.4.5. VEOMEGAx (Rotation Speed (speed [Hz] / maximum speed × 2^{15}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	OMEGA							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	OMEGA							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	OMEGA[15:0]	R/W	Rotation speed setting, 16-bit fixed-point data "-1.0" to "1.0" "0x0000" to "0xFFFF": Rotation speed [Hz] / Max_Hz × 2^{15} (Max_Hz: Maximum rotation speed)

13.3.4.6. VETHETAx (Motor Phase (motor phase [deg] / 360 × 2^{16}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	THETA							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	THETA							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	THETA[15:0]	R/W	Phase setting, 16-bit fixed-point data "0.0" to "1.0" Calculation formula: Phase [deg] / 360 × 2^{16}

13.3.4.7. VEIDREFx (d-Axis Reference Current Value (current [A] / maximum current × 2^{15}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	IDREF							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	IDREF							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	IDREF[15:0]	R/W	d-axis current reference value 16-bit fixed-point data "-1.0" to "1.0" "0x0000" to "0xFFFF" (d-axis current reference value [A] / Max_I × 2^{15}) Max_I: (When ADC converted value changes 1 LSB, amount of change in phase current [A]) × 2^{11}

13.3.4.8. VEIQREFx (q-Axis Reference Current Value (current [A] / maximum current × 2^{15}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	IQREF							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	IQREF							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	IQREF[15:0]	R/W	q-axis current reference value 16-bit fixed-point data "-1.0" to "1.0" "0x0000" to "0xFFFF" (q-axis current reference value [A] / Max_I × 2^{15}) Max_I: (When ADC converted value changes 1 LSB, amount of change in phase current [A]) × 2^{11}

13.3.4.9. VEVDx (d-Axis Voltage (voltage [V] / maximum voltage × 2^{31}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	VD							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	VD							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VD							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VD							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	VD[31:0]	R/W	d-axis voltage, 32-bit fixed-point data "-1.0" to "1.0" "0x0000_0000" to "0xFFFF_FFFF" (d-axis voltage / Max_V × 2^{31}) Max_V: (When ADC converted value changes 1 LSB, amount of change in supply voltage [V])× 2^{12}

13.3.4.10. VEVQx (q-Axis Voltage (voltage [V] / maximum voltage × 2^{31}) Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	VQ							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	VQ							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VQ							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VQ							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	VQ[31:0]	R/W	q-axis voltage, 32-bit fixed-point data "-1.0" to "1.0" "0x0000_0000" to "0xFFFF_FFFF" (q-axis voltage / Max_V × 2^{31}) Max_V: (When ADC converted value changes 1 LSB, amount of change in supply voltage [V]) × 2^{12}

13.3.4.11. VECIDKIx (d-Axis Current Control Integral Coefficient for PI Control Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CIDKI							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CIDKI							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0"
15:0	CIDKI[15:0]	R/W	Integral coefficient for PI control of d-axis: "0x0000" to "0xFFFF"

13.3.4.12. VECIDKPx (d-Axis Current Control Proportional Coefficient for PI Control Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CIDKP							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CIDKP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0"
15:0	CIDKP[15:0]	R/W	Proportional coefficient for PI control of d-axis: "0x0000" to "0xFFFF"

13.3.4.13. VEVCIQKIx (q-Axis Current Control Integral Coefficient for PI Control Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CIQKI							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CIQKI							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0"
15:0	CIQKI[15:0]	R/W	Integral coefficient for PI control of q-axis: "0x0000" to "0xFFFF"

13.3.4.14. VECIQKPx (q-Axis Current Control Proportional Coefficient for PI Control Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	CIQKP							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	CIQKP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0"
15:0	CIQKP[15:0]	R/W	Proportional coefficient for PI control of q-axis: "0x0000" to "0xFFFF"

13.3.4.15. VEVDIHx (d-Axis Voltage Integral Term Hold (upper 32-bit of VDI) Register)

	31	30	29	28	27	26	25	24
Bit symbol	VDIH							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	VDIH							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VDIH							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VDIH							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	VDIH[31:0]	R/W	Upper 32-bit integral term (VDI) for PI control of d-axis

13.3.4.16. VEVDILHx (d-Axis Voltage Integral Term Hold (lower 32-bit of VDI) Register)

	31	30	29	28	27	26	25	24
Bit symbol	VDILH							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	VDILH							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	VDILH[15:0]	R/W	Lower 32-bit integral term (VDI) for PI control of d-axis, but lower 16 bits are fixed to "0x0000".
15:0	-	R	Read as "0".

Note: VDI: 64-bit fixed-point data (fractional of 63bits "-1.0"to"1.0")

13.3.4.17. VEVQIHx (q-Axis Voltage Integral Term Hold (upper 32-bit of VQI) Register)

	31	30	29	28	27	26	25	24
Bit symbol	VQIH							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	VQIH							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VQIH							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VQIH							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	VQIH[31:0]	R/W	Upper 32-bit integral term (VQI) for PI control of q-axis

13.3.4.18. VEVQILHx (q-Axis Voltage Integral Term Hold (lower 32-bit of VQI) Register)

	31	30	29	28	27	26	25	24
Bit symbol	VQILH							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	VQILH							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	VQILH[15:0]	R/W	Lower 32-bit integral term (VQI) for PI control of q-axis, but lower 16 bits are fixed to "0x0000"
15:0	-	R	Read as "0".

Note: VQI: 64-bit fixed point data (fractional of 63bits "-1.0" to "1.0")

13.3.4.19. VEFPWMCHGx (Rotation Speed When Shift PWM is Enabled with 2-Phase Modulation Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	FPWMCHG							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	FPWMCHG							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	FPWMCHG[15:0]	R/W	PWM rotation speed setting when shift PWM is enabled Set the rotation speed [Hz] / Max_Hz × 2^{15} (Max_Hz: Maximum rotation speed [Hz])

13.3.4.20. VEMDPRDx (PWM Cycle Setting (setting the value as same as PMD's PWM cycle.) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VMDPRD							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VMDPRD							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VMDPRD[15:0]	R/W	PWM cycle setting Set the value of the PMD's PMDxMDPRD

13.3.4.21. VEMINPLSx (Minimum Pulse Width Difference Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	MINPLS							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	MINPLS							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	MINPLS[15:0]	R/W	Set the minimum pulse width difference. (Minimum value of duty difference of 3-Phase PWM (VECMPUx, VECMPVx, VECMPWx)) Set value: minimum pulse width difference [s] / PWM counter clock cycle [s]

13.3.4.22. VETRGCRCx (Synchronous Trigger Correction Value Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	TRGCRC							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	TRGCRC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	TRGCRC[15:0]	R/W	Correct the synchronizing trigger timing. Set value: Correction time [s] / PWM counter clock cycle [s]

13.3.4.23. VECOSx (Cosine Value by THETA for Output Conversion (Q15 data) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	COS							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	COS							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	COS[15:0]	R/W	Cosine value by THETA value, 16-bit fixed point data "-1.0" to "1.0" "0x0000" to "0xFFFF"

13.3.4.24. VESINx (Sine Value by THETA for Output Conversion (Q15 data) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	SIN							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	SIN							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	SIN[15:0]	R/W	Sine value by THETA value, 16-bit fixed point data "-1.0" to "1.0" "0x0000" to "0xFFFF"

13.3.4.25. VECOSMx (Previous Cosine Value for Input Processing (Q15 data) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	COSM							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	COSM							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	COSM[15:0]	R/W	Store previous VECOSx value "0x0000" to "0xFFFF"

13.3.4.26. VESINMx (Previous Sine Value for Input Processing (Q15 data) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	SINM							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	SINM							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	SINM[15:0]	R/W	Store previous VESINx value "0x0000" to "0xFFFF"

13.3.4.27. VESECTORx (Sector Information Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	SECTOR			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:4	-	R	Read as "0".
3:0	SECTOR[3:0]	R/W	Sector information Set value: "0x0" to "0xF" Indicates the rotation position at the time of output divided by 12 sectors each 30 degrees.

13.3.4.28. VESECTORMx (Previous Sector Information Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	SECTORM			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:4	-	R	Read as "0".
3:0	SECTORM[3:0]	R/W	Previous sector information Set value: "0x0" to "0xF" Used in input processing.

13.3.4.29. VEIAOx (AD Conversion Result at Detecting a-Phase Zero Current Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	IAO							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	IAO							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	IAO[15:0]	R/W	Store AD conversion result of a-phase zero current. (Stores the AD conversion result of a-phase current when a motor is stopped.)

Note1: The AD conversion result is automatically stored when zero current detection is enabled.

Note2: The AD conversion result is stored in the upper 12 bits. The lower 4 bits are always "0".

13.3.4.30. VEIBOx (AD Conversion Result at Detecting b-Phase Zero-Current Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	IBO							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	IBO							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	IBO[15:0]	R/W	Store AD conversion result of b-phase zero current. (Stores the AD conversion result of b-phase current when a motor is stopped.)

Note1: The AD conversion result is automatically stored when zero current detection is enabled.

Note2: The AD conversion result is stored in the upper 12 bits. The lower 4 bits are always "0".

13.3.4.31. VEICOx (AD Conversion Result at Detecting c-Phase Zero-Current Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	ICO							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	ICO							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	ICO[15:0]	R/W	Store AD conversion result of c-phase zero current. (Stores the AD conversion result of c-phase current when a motor is stopped.)

Note1: The AD conversion result is automatically stored when zero current detection is enabled.

Note2: The AD conversion result is stored in the upper 12 bits. The lower 4 bits are always "0".

13.3.4.32. VEIAADCx (a-Phase Current AD Conversion Result Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	IAADC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	IAADC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	IAADC[15:0]	R/W	Store the AD conversion result of a-phase current "0x0000" to "0xFFF0"

Note: The AD conversion result is stored in the upper 12 bits. The lower 4 bits are always "0".

13.3.4.33. VEIBADCx (b-Phase Current AD Conversion Result Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	IBADC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	IBADC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	IBADC[15:0]	R/W	Store the AD conversion result of b-phase current "0x0000" to "0xFFF0"

Note: The AD conversion result is stored in the upper 12 bits. The lower 4 bits are always "0".

13.3.4.34. VEICADCx (c-Phase Current AD Conversion Result Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ICADC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ICADC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	ICADC[15:0]	R/W	Store the AD conversion result of c-phase current "0x0000" to "0xFFF0"

Note: The AD conversion result is stored in the upper 12 bits. The lower 4 bits are always "0".

13.3.4.35. VEVDCx (DC Supply Voltage (voltage [V] / maximum voltage × 2^{15}) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	VDC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	VDC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VDC[15:0]	R/W	Supply voltage, 16-bit fixed-point data "0" to "1.0" Set value: "0x0000" to "0x7FFF" The actual voltage value: VDC value × Max_V / 2^{15} (Max_V: (When ADC converted value changes 1 LSB, amount of change in supply voltage [V]) × 2^{12})

13.3.4.36. VEIDx (d-Axis Current (current [A] / maximum current × 2^{31}) Register)

	31	30	29	28	27	26	25	24
Bit symbol	ID							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	ID							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	ID							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	ID							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	ID[31:0]	R/W	d-axis current, 32-bit fixed-point data "-1.0" to "1.0" Setting value: "0x0000_0000" to "0xFFFF_FFFF" The actual current value: ID value × Max_I / 2^{31} (Max_I: (When ADC converted value changes 1 LSB, amount of change in phase current [A]) × 2^{11})

13.3.4.37. VEIQx (q-Axis Current (current [A] / maximum current x. 2^{31}) Register)

	31	30	29	28	27	26	25	24
Bit symbol	IQ							
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	IQ							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	IQ							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	IQ							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:0	IQ[31:0]	R/W	q-axis current, 32-bit fixed-point data "-1.0" to "1.0" Setting value: "0x0000_0000" to "0xFFFF_FFFF" The actual current value: IQ value × Max_I / 2^{31} (Max_I: (When ADC converted value changes 1 LSB, amount of change in phase current [A]) × 2^{11})

13.3.4.38. VECMPUx (PMD Control: U-Phase PWM Pulse Width Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VCMPU							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VCMPU							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VCMPU[15:0]	R/W	U-phase PWM pulse width setting Set value: "0x0000" to "0xFFFF"

13.3.4.39. VECMPVx (PMD Control: V-Phase PWM Pulse Width Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VCMPV							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VCMPV							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VCMPV[15:0]	R/W	V-phase PWM pulse width setting Set value: "0x0000" to "0xFFFF"

13.3.4.40. VECMPWx (PMD Control: W-Phase PWM Pulse Width Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VCMPW							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VCMPW							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VCMPW[15:0]	R/W	W-phase PWM pulse width setting Set value: "0x0000" to "0xFFFF"

13.3.4.41. VEOUTCRx (PMD Control: 6-Phase Output Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	WPWM
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VPWM	UPWM	WOC		VOC		UOC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:9	-	R	Read as "0".
8	WPWM	R/W	PWM of W-phase 0: ON/OFF output 1: PWM output
7	VPWM	R/W	PWM of V-phase 0: ON/OFF output 1: PWM output
6	UPWM	R/W	PWM of U-phase 0: ON/OFF output 1: PWM output
5:4	WOC[1:0]	R/W	Output control of W-phase 00: WO OFF, ZO OFF (Note) 01: WO OFF, ZO ON 10: WO ON, ZO OFF 11: WO ON, ZO ON
3:2	VOC[1:0]	R/W	Output control of V-phase 00: VO OFF, YO OFF (Note) 01: VO OFF, YO ON 10: VO ON, YO OFF 11: VO ON, YO ON
1:0	UOC[1:0]	R/W	Output control of U-phase 00: UO OFF, XO OFF (Note) 01: UO OFF, XO ON 10: UO ON, XO OFF 11: UO ON, XO ON

Note: When VEOUTCRx<WPWM>, <VPWM>, <UPWM >= "1", both are ON.

Output control of U, V, and W phases of PMD is shown below. (Only the combination used in VE is shown.)

Table 13.1 VEOUTCRx<UPWM>, VEOUTCRx<UOC[1:0]> PMD setting: Output control of U-phase (UOx, XOx)

Setting		Output	
VEOUTCRx <UPWM>	VEOUTCRx <UOC[1:0]>	UOx	XOx
0	00	OFF output	OFF output
1	00	PWMU invert output	PWMU output
1	11	PWMU output	PWMU invert output

Table 13.2 VEOUTCRx<VPWM>, VEOUTCRx<VOC[1:0]> PMD setting: Output control of V-phase (VOx, YOx)

Setting		Output	
VEOUTCRx <VPWM>	VEOUTCRx <VOC[1:0]>	VOx	YOx
0	00	OFF output	OFF output
1	00	PWMV invert output	PWMV output
1	11	PWMV output	PWMV invert output

Table 13.3 VEOUTCRx<WPWM>, VEOUTCRx<WOC[1:0]> PMD setting: Output control of W-phase (WOx, ZOx)

Setting		Output	
VEOUTCRx <WPWM>	VEOUTCRx <WOC[1:0]>	WOx	ZOx
0	00	OFF output	OFF output
1	00	PWMW invert output	PWMW output
1	11	PWMW output	PWMW invert output

13.3.4.42. VETRGCMP0x (PMD Control: Trigger Timing Setting (TRGCMP0) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VTRGCMP0							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VTRGCMP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VTRGCMP0 [15:0]	R/W	Sets the trigger timing (PMD setting) for sampling ADC in synchronization with PMD. 0 x 0000: Prohibited 0x0001 to (PMDxMDPRD<MDPRD[15:0]> - 1): Trigger timing PMDxMDPRD<MDPRD[15:0]> to "0xFFFF": Prohibited

Note1: Enabled when one of the following is selected as the trigger mode of the PMD.

- Match during down counting
- Match during up counting
- Match during up and down counting

Note2: It is invalid when trigger selected output (PMDxTRGMD<TRGOUT> = "1") is selected as trigger output mode of PMD.

13.3.4.43. VETRGCMP1x (PMD Control: Trigger Time Setting (TRGCMP1) Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	VTRGCMP1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	VTRGCMP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	VTRGCMP1 [15:0]	R/W	Sets the trigger timing (PMD setting) for sampling ADC in synchronization with PMD. 0 x 0000: Prohibited 0x0001 to (PMDxMDPRD<MDPRD[15:0]> - 1): Trigger timing PMDxMDPRD<MDPRD[15:0]> to "0xFFFF": Prohibited

Note1: Enabled when one of the following is selected as the trigger mode of the PMD.

- Match during down counting
- Match during up counting
- Match during up and down counting

Note2: It is invalid when trigger selected output (PMDxTRGMD<TRGOUT> = "1") is selected as trigger output mode of PMD.

13.3.4.44. VETRGSELx (PMD Control: Synchronous Trigger Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	VTRGSEL		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:3	-	R	Read as "0".
2:0	VTRGSEL[2:0]	R/W	Specifying the synchronous trigger number which is output at the setting timing of VETRGCMP0x<VTRGCMP0[15:0]>. (Note) 0 to 5: Output trigger number 6 to 7: Prohibited

Note: Enabled when the trigger selection output (PMDxTRGMD<TRGOUT> = "1") is selected as the trigger output mode of the PMD.

13.3.4.45. VEEMGRSx (PMD Control: EMG Return Setting Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	-	EMGRS
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0".
0	EMGRS	R/W	EMG return command (PMD setting) 0: - 1: EMG return command

13.4. Operations

13.4.1. Schedule Management

Motor control is performed in the flow shown in Figure 13.4. VE transitions the each operation status by switching between scheduled setting and MODE setting.

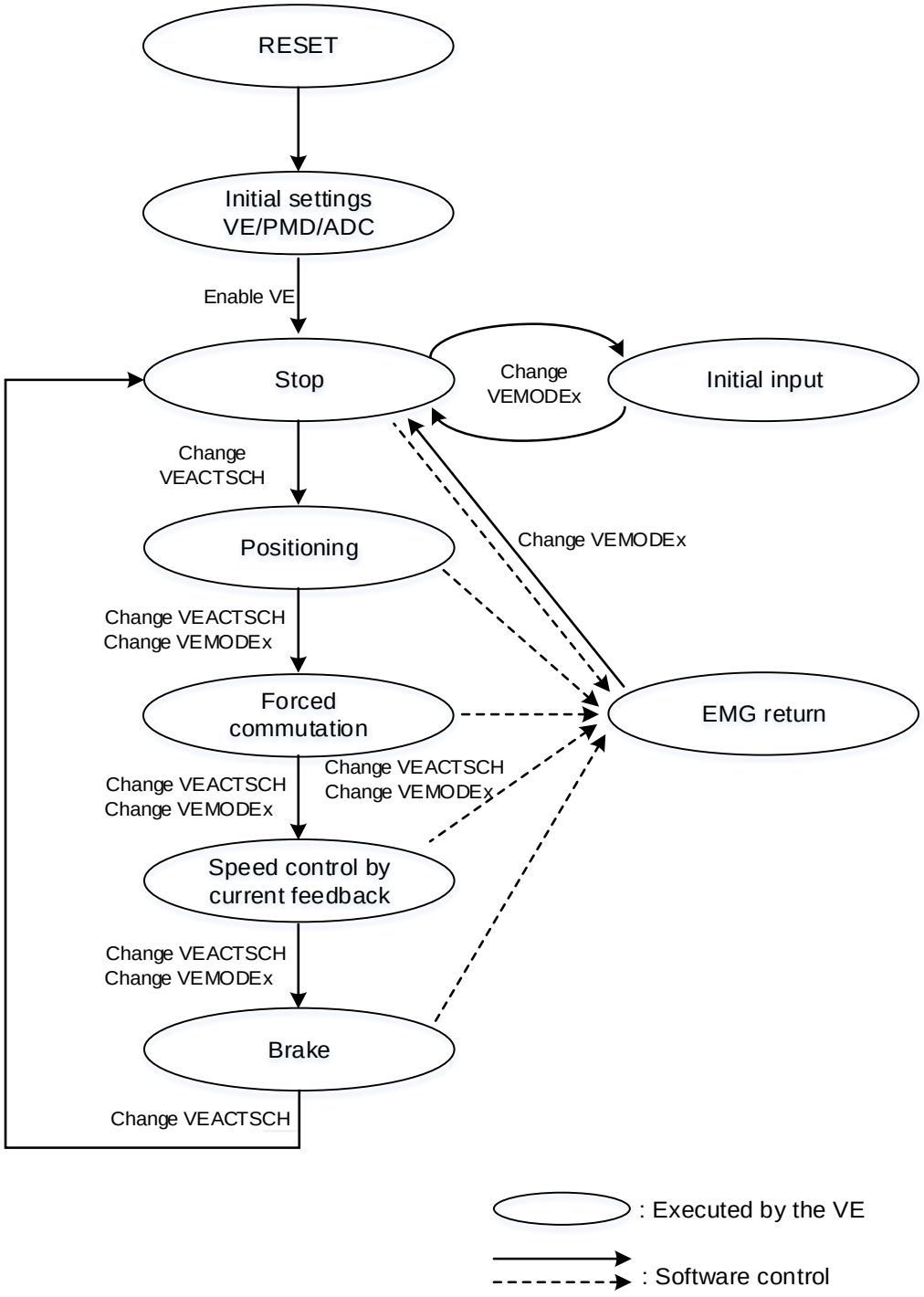


Figure 13.4 Motor Control Operation Flow Example

RESET	MCU reset
Initial Settings	Initial setting by user software
Stop	Motor stop
Initial input	Sample and store zero-current data at motor stop
Positioning	Positioning control at motor start
Forced commutation	Motor is rotated at the set speed without feedback control for the starting specified time
Speed control by current feedback	Control motor rotation by current feedback
Brake	Deceleration control
EMG return	Return from EMG protection status

13.4.1.1. Schedule Control

The operation schedules are selected with VEACTSCH.

A schedule is comprised of an output schedule handling output processing and an input schedule handling input processing. An output schedule consists of output-related tasks, and an input schedule consists of input-related tasks. Table 13.4 lists the relationship between schedules and tasks to operate.

In addition, phase interpolation enable, output control operation, and zero current detection are switched in VEMODEx setting according to the motor control flow (Table 13.5).

Table 13.4 Tasks to be Executed in Each Schedule

Schedule Selection VEACTSCH	Output Schedule						Input Schedule		
	Current control	SIN/COS computation	Output coordinate axis conversion	Output phase conversion	Output control	Trigger generation	Input processing	Input phase conversion	Input coordinate axis conversion
0: Individual operation	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)
1: Schedule 1	✓	✓ (Note2)	✓	✓	✓ (Note3)	✓	✓ (Note4)	✓	✓
4: Schedule 4	-	✓ (Note2)	✓	✓	✓ (Note3)	✓	✓ (Note4)	✓	✓
9: Schedule 9	-	-	-	-	✓ (Note3)	✓	✓ (Note4)	-	-

✓: A task that is executed according to the schedule.,-: A task that is independent of the schedule.

Note1: Only specified task is executed.

Note2: Phase interpolation setting

Note3: Output off setting: VEEMGRSx<EMGRS>

Note4: Task operation to be switched by zero current detection.

Table 13.5 Example Setting in Typical Operation Flow

Setting / Motor Control Flow	Schedule setting VEACTSCH <VACTB[3:0]>	Task specification VETASKAPP <VTASKB[3:0]>	Phase interpolation enable VEMODEx <PVIEN>	Output control operation VEMODEx <OCRMD[1:0]>	Zero current detection VEMODEx <ZIEN>
Stop	1001	0000	-	00	0
Initial input	1001	0000	-	00	1
Positioning	0001	0101	0	01	0
Forced commutation	0001	0101	1	01	0
Speed control by current feedback	0001	0101	1	01	0
Brake	0100	0110	0	01	0
EMG return	1001	0000	-	11	0

-: Not set

An output schedule begins executing by the command (VECPURUNTRG). When all output related tasks are completed, the VE enters a standby state and waits for a start trigger. At this time, schedule of other channels can be executed.

An input schedule begins executing by a start trigger. When all input related tasks are completed, the VE generates an interrupt to the CPU and enters a halt state. However, if the schedule repeat count (VEREPTIME) is set to 2 or more times, the output schedule is started without generating an interrupt until the set count is reached.

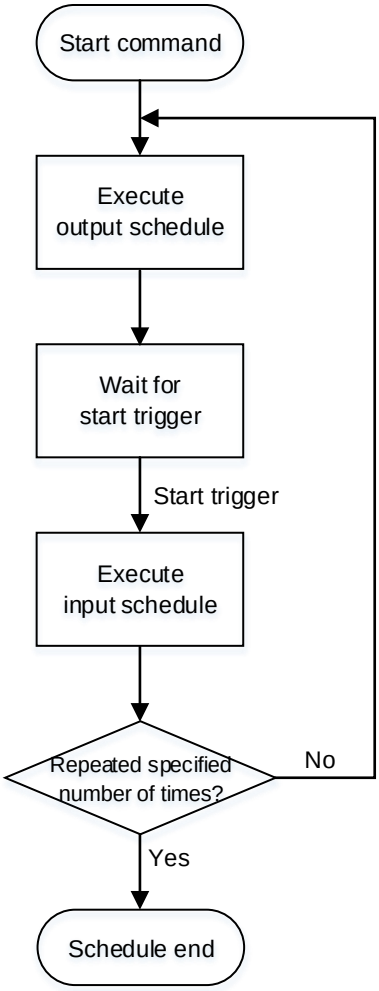


Figure 13.5 Operation Transition in Operation Schedule

13.4.1.2. Start Control

Enable the VE (VEEN<VEEN>= "1"). Set the operation schedule specification (VEACTSCH), task specification (VETASKAPP), and repetition count (VEREPTIME). The operation schedule of the VE is comprised of output schedule and input schedule. Basically, the VE executes output schedule first, enters the standby state, and then starts executing the input schedule by the start trigger.

The output schedule and the input schedule are started by each following conditions.

- Starting the output schedule
  - (a) Starting the specified task (VETASKAPP) with the command start (VECPURUNTRG)
  - (b) Repeat start (VEREPTIME ≥ 2) is executed after the input schedule is completed.

- Starting the input schedule
  - (a) Starting the input processing task by the start trigger (Trigger input selected in VETRGMODE) after the output schedule completes
  - (b) Starting the specified task (VETASKAPP) with the command start (VECPURUNTRG)

13.4.2. Task Overview

The following is the overview of each task that operates on the schedule.

To specify an individual operation or startup task, specify the task number of the Table 13.6.

Table 13.6 Task Overview

Task		Task function	Task number
Output Schedule	Current control	dq current control	5
	SIN/COS computation	Sine/cosine computation, phase interpolation	6
	Output coordinate axis conversion	Convert from dq coordinate axis to αβ coordinate axis	7
	Output phase conversion	Convert from 2-phase to 3-phase	8
	Output control	Data conversion to PMD setting format Shift PWM switching	0
	Trigger generation	Synchronous trigger timing generation	1
Input Schedule	Input processing	Capture AD conversion results Data conversion to fixed-point data	2
	Input phase conversion	Convert from 3-phases to 2-phases	3
	Input coordinate axis conversion	Convert from αβ coordinate axis to dq coordinate axis	4

13.4.2.1. Current Control

The current control unit is comprised of a PI control unit for d-axis and a PI control unit for q-axis. It calculates d-axis and q-axis voltages.

(1) PI control of d-axis current

<Formula>

$\Delta ID = VEIDREFx - VEIDx<ID[31:0]>$: Difference between current reference value and current feedback

$VDIx = VECIDKIx \times \Delta ID + VDIx$: Integral term operation

$VEVDx = VECIDKPx \times \Delta ID + VDIx$: Voltage calculation using proportional term

Table 13.7 PI Control of d-axis Current

	Register name	Function	Data format
Input	VEIDx	d-axis current	32-bit fixed-point data (31 bits after the decimal point)
	VEIDREFx	d-axis current reference value	16-bit fixed-point data (15 bits after the decimal point)
	VECIDKPx	Proportional coefficient	16-bit data
	VECIDKIx	Integral coefficient	16-bit data
Output	VEVDx	d-axis voltage	32-bit fixed-point data (31 bits after the decimal point)
Internal	VDIx	d-axis voltage integral term hold	64-bit fixed-point data (63 bits after the decimal point)

(2) PI control of q-axis current

<Formula>

$\Delta IQ = VEIQREFx - VEIQx<IQ[31:0]>$: Difference between current reference value and current feedback

$VQIx = VECIQKIx \times \Delta IQ + VQIx$: Integral term operation

$VEVQx = VECIQKPx \times \Delta IQ + VQIx$: Voltage calculation using proportional term

Table 13.8 PI Control of q-axis Current

	Register name	Function	Data format
Input	VEIQx	q-axis current	32-bit fixed-point data (31 bits after the decimal point)
	VEIQREFx	q-axis current reference value	16-bit fixed-point data (15 bits after the decimal point)
	VECIQKPx	Proportional coefficient	16-bit data
	VECIQKIx	Integral coefficient	16-bit data
Output	VEVQx	q-axis voltage	32-bit fixed-point data (31 bits after the decimal point)
Internal	VQI1x	q-axis voltage integral term hold	64-bit fixed-point data (63 bits after the decimal point)

13.4.2.2. SIN/COS Computation

The SIN/COS computation unit is comprised of phase interpolation unit and SIN/COS computation unit.

Phase interpolation calculates the rotation speed by integral with the PWM cycle. It is executed only when phase interpolation is enabled.

(1) phase interpolation

<Formula>

VETHETAx = VEOMEGAx × VETPWMx + VETHETAx : Integral of rotation speed, only when phase interpolation is enabled

Table 13.9 Phase Interpolation

	Register name	Function	Data format
Input	VETHETAx	Phase θ	16-bit fixed-point data ("0.0" to "1.0", 16 bits after the decimal point)
	VEOMEGAx	Rotation speed	16-bit fixed-point data ("-1.0" to "1.0", 15 bits after the decimal point)
	VETPWMx	PWM cycle rate	16-bit fixed-point data ("0.0" to "1.0", 16 bits after the decimal point)
	VEMODEx	Phase Interpolation Enable	Setting
Output	VETHETAx	Phase θ	16-bit fixed-point data ("0.0" to "1.0", 16 bits after the decimal point)

(2) SIN/COS computation

<Formula>

VESINMx = VESINx : Saves previous value (for input processing)

VECOSMx = VECOSx : Saves previous value (for input processing)

VESINx = sin(VETHETAx × π) : SIN/COS computation

VECOSx = sin((VETHETAx + 1 / 4) × π) : SIN/COS computation

Table 13.10 SIN/COS Computation

	Register name	Function	Data format
Input	VETHETAx	Phase θ	16-bit fixed-point data ("0.0" to "1.0", 16 bits after the decimal point)
Output	VESINx	Sine value at θ	16-bit fixed-point data ("-1.0" to "1.0", 15 bits after the decimal point)
	VECOSx	Cosine value at θ	
	VESINMx	Previous sine value	
	VECOSMx	Previous cosine value	

13.4.2.3. Output Voltage Conversion (Coordinate Axis Conversion/Phase Conversion)

Output voltage conversion is comprised of dq-αβ coordinate axis conversion and 2-3-phase conversion. The dq-αβ coordinate axis conversion calculates Vα, Vβ from Vd, Vq, VESINx, and VECOSx.

In the 2-3 phase conversion, the sector is decided from Vα and Vβ and Va, Vb, and Vc are calculated by executing space vector conversion according to the decided sector.

In 2-3 phase conversion, 2-phase modulation or 3-phase modulation can be selected as the conversion method.

(1) dq-αβ coordinate axis conversion

<Formula>

VETMPREG3 = VECOSx × VEVDx - VESINx × VEVQx : Calculates Vα

VETMPREG4 = VESINx × VEVDx + VECOSx × VEVQx : Calculates Vβ

Table 13.11 dq-αβ Coordinate Axis Conversion

	Register name	Function	Data format
Input	VEVDx	d-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VEVQx	q-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VESINx	Sine value at θ	16-bit fixed-point data ("-1.0" to "1.0", 15 bits after the decimal point)
	VECOSx	Cosine value at θ	16-bit fixed-point data ("-1.0" to "1.0", 15 bits after the decimal point)
Output	VETMPREG3	α-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG4	β-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)

(2) 2-3 phase conversion (space vector conversion)

(a) sector decision

<Formula>

VESECTORMx = VESECTORx : Stores previous sector

if (Vα ≥ 0 & Vβ ≥ 0) : Calculates Vβ

```
    if (|Vα| ≥ |Vβ| / √3)
            if (|Vα| / √3 ≥ |Vβ|)      SECTOR= "0"
            else                       SECTOR = "1"
    else                               SECTOR = "2"
else if (Vα < 0 & Vβ ≥ 0)
    if (|Vα| < |Vβ| / √3)              SECTOR = "3"
    else if (|Vα| / √3 < |Vβ|)         SECTOR = "4"
        else                           SECTOR = "5"
else if (Vα < 0 & Vβ < 0)
    if (|Vα| ≥ |Vβ| / √3)
            if (|Vα| / √3 ≥ |Vβ|)      SECTOR = "6"
            else                       SECTOR = "7"
    else                               SECTOR = "8"
else if (Vα ≥ 0 & Vβ < 0)
    if (|Vα| < |Vβ| / √3)              SECTOR = "9"
    else if (|Vα| / √3 < |Vβ|)         SECTOR = "10"
        else                           SECTOR = "11"
```

Table 13.12 2-3 Phase Conversion (Space Vector Conversion)

	Register name	Function	Data format
Input	VETMPREG3	α-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG4	β-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
Output	VESECTORx	Sector	4-bit data
	VESECTORMx	Previous sector	4-bit data

(b) 3 phase voltage calculation (when VESECTORx<SECTOR[3:0]>= "0")

<Formula>

if (VESECTORx<SECTOR[3:0]>= "0")

t1 = (√3) / (VEVDCx) × ((√3) / 2 × Vα - 1 / 2 × Vβ) : Calculates V1 period

t2 = (√3) / (VEVDCx) × (Vβ) : Calculates V2 period

t3 = 1 − t1 − t2 : Calculates V0 + V7 period

if (VEFMODEx<C2PEN> = "0") : 3-phase modulation

VETMPREG0 = t1 + t2 + t3 / 2 : Calculates Va

VETMPREG1 = t2 + t3 / 2 : Calculates Vb

VETMPREG2 = t3 / 2 : Calculates Vc

else : 2-phase modulation

VETMPREG0 = t1 + t2 : Calculates Va

VETMPREG1 = t2 : Calculates Vb

VETMPREG2 = "0" : Calculates Vc

Table 13.13 3 Phase Voltage Calculation (When VESECTORx<SECTOR[3:0]>= "0")

	Register name	Function	Data format
Input	VETMPREG3	α-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG4	β-axis voltage	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VEVDCx	Supply voltage	16-bit fixed-point data ("0.0" to "1.0", 15 bits after the decimal point)
	VESECTORx	Sector	4-bit data
	VEFMODEx	Modulation mode	VEFMODEx<C2PEN> 0: 3-phase modulation 1: 2-phase modulation
Output	VETMPREG0	a-phase voltage	32-bit fixed-point data ("0.0" to "1.0", 31 bits after the decimal point)
	VETMPREG1	b-phase voltage	32-bit fixed-point data ("0.0" to "1.0", 31 bits after the decimal point)
	VETMPREG2	c-phase voltage	32-bit fixed-point data ("0.0" to "1.0", 31 bits after the decimal point)

13.4.2.4. Output Control

The output control unit converts the 3-phase voltage value to VECMPUx, VECMPVx, VECMPWx of the PWM setting format and sets VEOUTCRx according to the operation mode.

When 1-shunt current detection and 2-phase modulation are selected and shift PWM is enabled, if the rotation speed is slower than the shift PWM switching reference value, output is switched to shift PWM output.

Table 13.14 Output Control

	Register name	Function	Data format
Input	VETMPREG0	a-phase voltage	32-bit fixed-point data ("0.0" to "1.0", 31 bits after the decimal point)
	VETMPREG1	b-phase voltage	32-bit fixed-point data ("0.0" to "1.0", 31 bits after the decimal point)
	VETMPREG2	c-phase voltage	32-bit fixed-point data ("0.0" to "1.0", 31 bits after the decimal point)
	VEMDPRDx	PWM cycle setting	16-bit data (PMD PWM cycle setting)
	VESECTORx	Sector	4-bit data
	VEOMEGAx	Rotational speed	16-bit fixed-point data ("-1.0" to "1.0", 15 bits after the decimal point)
	VEFPWMCHGx	Shift PWM switching reference	16-bit fixed-point data ("0.0" to "1.0", 15 bits after the decimal point)
	VEMODEx	Output control operation	Setting
	VEFMODEx	PMD channel selection/Shift PWM enable /modulation mode /detection mode	Setting
Output	VECMPUx	PMD U-phase PWM setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VECMPVx	PMD V-phase PWM setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VECMPWx	PMD W-phase PWM setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VEOUTCRx	PMD output control setting	9-bit setting
	VEEMGRSx	PMD EMG return	1-bit setting
	VEMCTLFx	Shift Switching Flag	Status

13.4.2.5. Trigger Generation

The trigger generator unit calculates the trigger timing according to the current detection method from VECMPUx, VECMPVx, VECMPWx and sets VETRGCMP0x and VETRGCMP1x to the calculated trigger timing.

Table 13.15 Trigger Generation

	Register name	Function	Data format
Input	VECMPUx	PMD U-phase PWM setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VECMPVx	PMD V-phase PWM setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VECMPWx	PMD W-phase PWM setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VEMDPRDx	PWM cycle setting	16-bit data (PMD PWM cycle setting)
	VETADC	AD conversion time	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VETRGCRCx	Trigger Compensation Value	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VESECTORx	Sector	4-bit data
	VEMODEx	Output control operation	Setting
	VEFMODEx	PMD channel selection/Shift PWM enable /modulation mode /detection mode/trigger correction enable	Setting
	VEMCTLFx	Shift Switching Flag	Status
Output	VETRGCMP0x	PMD trigger 0 timing setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VETRGCMP1x	PMD trigger 1 timing setting	16-bit data (0 - VEMDPRDx<VMDPRD[15:0]>value)
	VETRGSELx	PMD Trigger Select	3-bit data

13.4.2.6. Input Processing

In the input processing task, the current conversion result is judged and stored as 3-phase component, and the conversion result of current and voltage is converted to fixed-point data. And, the result of zero current conversion is stored in initial input operation.

Table 13.16 Input Processing

	Register name	Function	Data format
Input	VEADREG0x	AD conversion result 0	16-bit data (results held in the upper 12 bits)
	VEADREG1x	AD conversion result 1	
	VEADREG2x	AD conversion result 2	
	VEADREG3x	AD conversion result 3	
	VEPHNUM0x	ADREG0x detection phase information	2-bit data
	VEPHNUM1x	ADREG1x detection phase information	
	VEPHNUM2x	ADREG2x detection phase information	
	VEPHNUM3x	ADREG3x detection phase information	
	VESECTORMx	Sector information	4-bit data
	VEMODEx	Zero current detection	Setting
	VEFMODEx	PMD channel selection/Current detection mode/shift PWM enable	Setting
	VEMCTLFx	Shift Switching Flag	Status
Output	VEVDCx	Supply voltage	16-bit fixed-point data ("0.0" to "1.0", 15 bits after the decimal point)
	VETMPREG0	a-phase current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG1	b-phase current	
	VETMPREG2	c-phase current	
Internal	VEIAOx	a-phase zero current conversion result	16-bit data (results held in the upper 12 bits)
	VEIBOx	b-phase zero current conversion result	
	VEICOx	c-phase zero current conversion result	
	VEIAADCx	a-phase current conversion result	16-bit data (results held in the upper 12 bits)
	VEIBADCx	b-phase current conversion result	
	VEICADCx	c-phase current conversion result	

13.4.2.7. Input Current Conversion (Phase Conversion/Coordinate Axis Conversion)

The input current conversion consists of 3-2 phase conversion and αβ-dq coordinate axis conversion.

The 3-2 phase conversion calculates Iα and Iβ from Ia, Ib and Ic.

The αβ-dq coordinate conversion calculates Id and Iq from Iα, Iβ, and VESINMx, VECOSMx.

(1) 3-2 phase conversion

<Formula>

VETMPREG3 = VETMPREG0 : Calculates Iα

VETMPREG4 = 1 / √3 × VETMPREG1 - 1 / √3 × VETMPREG2 : Calculates Iβ

Table 13.17 3-2-phase Conversion

	Register name	Function	Data format
Input	VETMPREG0	a-phase current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG1	b-phase current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG2	c-phase current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
Output	VETMPREG3	α-axis current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG4	β-axis current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)

(2) αβ-dq coordinate axis conversion

<Formula>

VEIDx = VECOSMx × VETMPREG3 + VESINMx × VETMPREG4 : Calculates Id

VEIQx = - VESINMx × VETMPREG3 + VECOSMx × VETMPREG4 : Calculates Iq

Table 13.18 αβ-dq Coordinate Axis Conversion

	Register name	Function	Data format
Input	VETMPREG3	α-axis current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VETMPREG4	β-axis current	
	VESINMx	Sine value at θ	16-bit fixed-point data ("-1.0" to "1.0", 15 bits after the decimal point)
	VECOSMx	Cosine value at θ	
Output	VEIDx	d-axis current	32-bit fixed-point data ("-1.0" to "1.0", 31 bits after the decimal point)
	VEIQx	q-axis current	

13.5. VE and AD Conversion Result Register

The combination of PMD channel and ADC unit which can be used has restriction depending on the used VE channel.

And the values of AD conversion result registers 0 to 2 (ADxREG0 to 2, x = A, B) are calculated as current values, and the values of AD conversion result register 3 (ADxREG3, x = A, B) are calculated as voltage values. Allocate the AD conversion result register according to each mode of VE as shown in Table 13.19.

Table 13.19 Combination of VE Channel and PMD Channel

VE channel	PMD channel
ch0	ch0
ch1	ch1

Table 13.20 Combination of VE Channel, ADC Unit and AD Conversion Result Register

VE			ADC unit A				ADC unit B			
Channel	Current detection mode VEFMODEx <IDMODE[1:0]>	ADC unit selection VEFMODEx <ADCSEL[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	0*	00	Current data 1	Current data 2	(Note1)	VDC data	-	-	-	-
		1*	Current data 1	-	(Note1)	VDC data	Current data 2	-	-	-
	1*	00	Current data 1	Current data 2	-	VDC data	-	-	-	-
1	0*	00	-	-	-	-	Current data 1	Current data 2	(Note1)	VDC data
		1*	-	Current data 2	-	-	-	Current data 1	(Note1)	VDC data
	1*	01	-	-	-	-	Current data 1	Current data 2	-	VDC data

Note1: No ADxREG2 conversion is required, but phase information must be set. Refer to "11. 12-Bit Analog-to-Digital Converter (ADC)".

Note2: *: Don't care

Note3: Prohibit except shown above combination of current detection mode and AD conversion result register.

14. Encoder Input Circuit (ENC)

14.1. Outline

The encoder input circuit supports four operation modes including encoder mode, sensor mode (two types) and timer mode. And the functions are as follows:

- Supports incremental encoders and Hall sensor ICs. (signals of Hall sensor IC can be input directly)
- 24-bit general-purpose timer mode
- Built-in multiply-by-4 (multiply-by-6) circuit
- Built-in rotational direction detection circuit
- Built-in 24-bit counter
- Compare enable/disable
- Interrupt request output: 1
- Built-in digital noise filters for input signals

The encoder input circuit can obtain the absolute position of the motor, based on input signals from the incremental encoder.

14.2. Block Diagram

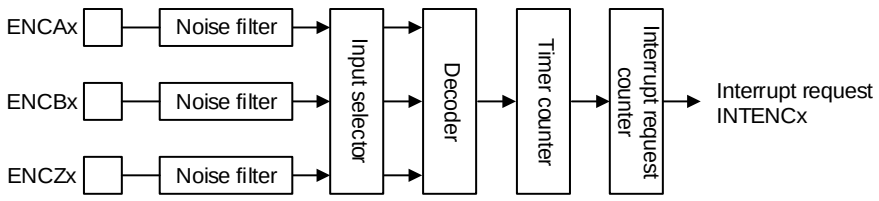


Figure 14.1 Block diagram of encoder input circuit

14.3. Registers

14.3.1. List of Registers

The following is control registers and addresses.

Register name		Address (Base+)
Encoder Input Control Register	ENxTNCR	0x0000
Encoder Counter Reload Register	ENxRELOAD	0x0004
Encoder Compare Register	ENxINT	0x0008
Encoder Counter Register	ENxCNT	0x000C

14.3.2. ENxTNCR (Encoder Input Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	MODE		P3EN
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	ZEN	ENRUN	NR		INTEN	ENDEV		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:19	-	R	Read as "0".
18:17	MODE[1:0]	R/W	Encoder input mode setting 00: Encoder mode 01: Sensor mode (event count) 10: Sensor mode (timer count) 11: Timer mode
16	P3EN	R/W	2-phase/3-phase input selection (sensor mode) (Note 1) 0: 2-phase input 1: 3-phase input Sets the number of input signals.
15	CMP	R	Compare flag 0: - 1: Compare (Clear by read) When comparing is executed, <CMP> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0", always "0" is set. Writing to this bit is no effect.

14	REVERR	R	Reverse error flag (Sensor mode (at timer count)) (Note 2) 0: - 1: Error (Clear by read) In sensor mode (at timer count), when a reverse error occurs, <REVERR> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0", always "0" is set. Writing to this bit is no effect. In the encoder mode, sensor mode (event count) and timer mode, this bit has no meaning.
13	UD	R	Rotation direction 0: CCW direction 1: CW direction <UD> is set to "1" when the CW direction is indicated. <UD> is cleared to "0" when the CCW direction is indicated. When <ENRUN> = "0", <UD> is always set to "0".
12	ZDET	R	Z- phase detected 0: Not detected 1: Z-phase detected <ZDET> is set to "1" on the first edge of Z input signal (ENCZx) after <ENRUN> is written from "0" to "1". This occurs on a rising edge of the signal Z-phase (CW direction) or on a falling edge of Z-phase (CCW direction). <ZDET> is always set to "0" when <ENRUN> = "0". <ZEN> has no influence on the value of <ZDET>. <ZDET> is always set to "0" in the sensor mode (event count) and sensor mode (timer count)
11	SFTCAP	W	Executes software capture (timer mode/sensor mode (at timer count)) 0: - 1: Software capture When <SFTCAP> is set to "1", the value of the encoder counter is captured into the ENxCNT register. Writing "0" to <SFTCAP> has no effect. Reading <SFTCAP> always returns "0". In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; writing "1" to this bit is no meaning.
10	ENCLR	W	Encoder pulse counter clear 0: - 1: Clear Writing a "1" to <ENCLR> clears the encoder counter to "0". Once cleared, the encoder counter restarts counting from "0". Writing "0" to <ENCLR> has no effect. Reading <ENCLR> always returns "0".
9	ZESEL	R/W	Edge selection of ENCZx (timer mode) 0: Rising edge 1: Falling edge In timer mode, this bit selects inputs edge of ENCZx used as external trigger. In the other mode, this bit has no meaning.
8	CMPEN	R/W	Compare enable 0: Disable 1: Enable When <CMPEN> is set to "1", counter values of encoder counter and register value of ENxINT are compared. When <CMPEN> is set to "0" , this compare is disabled.
7	ZEN	R/W	Z-phase enable (Encoder mode/timer mode) 0: Disable 1: Enable In the other mode, this bit has no meaning. <Encoder mode>: Sets clearing of encoder counter using ENCZx input When <ZEN> = "1", if a rising edge of ENCZx is detected during CW rotation, the encoder counter is cleared to "0". When a falling edge of ENCZx is detected during CCW rotation, the encoder counter is cleared to "0". When the timing of ENCLK (clock derived by multiplying the decoded A-phase and B-phase signals by 4) and the edge of ENCZx coincide, the encoder counter is cleared to "0" without incrementing or decrementing (i.e., the clear takes precedence). <Timer mode>: Sets ENCZx input to use as an external trigger. When <ZEN> = "1", the value of the encoder counter is captured and cleared to "0" on the edge of ENCZx selected by <ZESEL>.

6	ENRUN	R/W	Encoder operation enable 0: Disable 1: Enable When setting <ENRUN> to "1", clears <ZDET> to "0" and enables the encoder operation. Clearing <ENRUN> to "0" disables the encoder operation. There are counters and flags that are cleared and not cleared when <ENRUN> bit is cleared to "0".
5:4	NR[1:0]	R/W	Noise filter 00: No filtering 01: Filters out pulses narrower than 31 / fsys as noise 10: Filters out pulses narrower than 63 / fsys as noise 11: Filters out pulses narrower than 127 / fsys as noise Sets the width of the pulse to be removed as noise with the digital noise filter.
3	INTEN	R/W	Encoder interrupt enable 0: Disable 1: Enable Setting <INTEN> to "1" enables interrupt generation. Setting <INTEN> to "0" disables interrupt generation.
2:0	ENDEV[2:0]	R/W	Sets Encoder pulse division 000: divided by 1 100: divided by 16 001: divided by 2 101: divided by 32 010: divided by 4 110: divided by 64 011: divided by 8 111: divided by 128 Sets encoder pulse division. The encoder pulse is divided by <ENDEV[2:0]>. The divided signal is used for the interval of the event interrupt.

Note1: In the encoder mode or timer mode, <P3EN> must be set to "0".

Note2: When changing the mode, read the flag to clear to "0".

The operation mode is determined by <MODE[1: 0]>, <P3EN>, and <ZEN>, and there are 8 modes in total. The operation mode settings are as follows:

<MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Mode
00	0	0	A, B	Encoder mode
	1		A, B, Z	Encoder mode (use of Z-phase)
01	0	0	U, V	Sensor mode (event count, 2-phase input)
		1	U, V, W	Sensor mode (event count, 3-phase input)
10	0	0	U, V	Sensor mode (timer count, 2-phase input)
		1	U, V, W	Sensor mode (timer count, 3-phase input)
11	0	0	-	Timer mode
	1		Z	Timer mode (use of Z-phase)

The following is the status of <ENRUN> and each signal.

Counter/flag	<ENRUN> = "0" (After reset)	<ENRUN> = "1" (Operating)	<ENRUN> = "0" (Stopping)	<ENRUN> = "0" Object flag/counter clear procedure
Encoder counter	0x000000	Count operation	Maintains a value when stopping	Software clear (<ENCLR> = "1" WR)
Noise filter counter	0b0000000	Count-up operation	Count-up operation (Always filtering)	Only reset
Encoder pulse division counter	0x00	Count-down operation	Stopped and cleared	Clear when <ENRUN> = "0"
Compare flag <CMP>	0	"1" is set when comparing Clear when read.	Cleared	Clear when <ENRUN> = "0"
Reverse error flag <REVERR>	0	"1" is set when error occurs. Clear when read.	Cleared	Clear when <ENRUN> = "0"
Z detection flag <ZDET>	0	"1" is set when Z is detected.	Cleared	Clear when <ENRUN> = "0"
Rotation direction bit <UD>	0	"0"/"1" is set depending on the direction	Cleared	Clear when <ENRUN> = "0"

14.3.3. ENxRELOAD (Encoder Counter Reload Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	RELOAD							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	RELOAD							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:16	-	R	Read as "0".
15:0	RELOAD[15:0]	R/W	Sets the Encoder counter period (after multiplied by 4 or 6) "0x0000" to "0xFFFF" Z-phase is used : Sets the number of count pulses for one rotation Z-phase is not used : Sets the number of count pulses minus one for one rotation <RELOAD[15:0]> defines the encoder counter period multiplied by 4 or 6. When the encoder counter is counting up, when the value of the counter becomes equal to the value of <RELOAD [15: 0]>, it will be cleared to "0" at the next ENCLK timing. When down counting is performed, the value of <RELOAD [15: 0]> is loaded into the encoder counter at the timing of the next ENCLK when the value of the counter becomes "0".

Note1: ENxRELOAD register is only used in Encoder mode.

Note2: ENxRELOAD register should be accessed in word size.

14.3.4. ENxINT (Encoder Compare Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	INT							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	INT							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	INT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:24	-	R	Read as "0".
23:0	INT[23:0]	R/W	Counter compare value setting <Encoder mode> Interrupt condition setting of the encoder pulse position: "0x0000" to "0xFFFF" While <CMPEN> = "1", when an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". When <INTEN> = "1", an interrupt request (INTENCx) occurs. However when <ZEN> = "1", an interrupt request does not occur until <ZDET> = "1". <Sensor mode (event count)> Interrupt condition setting of the encoder pulse position: "0x0000" to "0xFFFF" While <CMPEN> = "1", when an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". When <INTEN> = "1" , an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>. <Sensor mode (Timer count)> Interrupt condition setting of abnormal pulse detection time: "0x000000" to "0xFFFFFF" When <CMPEN> = "1", when an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". When <INTEN> = "1", an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>. <Timer mode> Interrupt condition setting of timer compare: "0x000000" to "0xFFFFFF" When <CMPEN> = "1", when an internal counter value matches a value of <INT[23:0]>, <CMP> is set to "1". When <INTEN> = "1", an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.

Note 1: <INT[23:16]> is used only in Sensor mode (timer count) and Timer mode.

Note 2: ENxINT register should be accessed in word size.

14.3.5. ENxCNT (Encoder Counter Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	CNT							
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	CNT							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	CNT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:24	-	R	Read as "0".
23:0	CNT[23:0]	R/W	Encoder counter/captured value <Encoder mode> Counter value of encoder pulse: "0x0000" to "0xFFFF" The value of encoder count can be read. In Encoder mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to the value of <RELOAD[15:0]>, it is cleared to "0" on the next ENCLK timing. During CCW rotation, encoder counter counts down; when it has reached to "0", it is loaded with the value of <RELOAD[15:0]> on the next ENCLK timing. <Sensor mode (event count)> Counter value of encoder pulse: "0x0000" to "0xFFFF" The value of encoder count can be read. In Sensor Event Count mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to "0xFFFF", it is cleared to "0" on the next ENCLK timing. During CCW rotation, encoder counter counts down; when the encoder counter has reached to "0", "0xFFFF" is loaded to the counter on the next ENCLK timing. <Sensor mode (Timer count)> Pulse detection time or captured value by software: "0x000000" to "0xFFFFFF" This bit can read the value captured by the encoder counter by the encoder pulse (ENCLK) or the value captured for the encoder counter by writing "1" in <SFTCAP>. The captured value is cleared to "0" by system reset. It can also be cleared by software capture after clearing the counter by setting <ENCLR> to "1" and then setting <SFTCAP> to "1". In Sensor mode (Timer Count), the encoder counter is configured as a free-running counter that counts up with fsys. The encoder counter is cleared to "0" when the encoder pulse (ENCLK) is detected. When it has reached to "0xFFFFFF", it is cleared to "0" automatically. <Timer mode> Captured value of internal counter or captured value by software: "0x000000" to "0xFFFFFF" The value of encoder counter captured by software can be read by writing "1" to <SFTCAP>.When <ZEN> = "1", the value of the encoder counter is also captured into <CNT[23:0]> on the Z-phase edge selected by <ZESEL>. The captured value is cleared to "0" by reset. After the counter is cleared by setting <ENCLR> to "1", it can also be cleared by software capture. In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached to "0xFFFFFF", it is cleared to "0" automatically.

Note 1: <CNT[23:16]> is used only in the sensor mode (Timer counting) or timer mode. In the encoder mode or

sensor mode (event counting), always reads as "0".

Note 2: ENxCNT register should be accessed in word size.

14.4. Operational Description

14.4.1. Encoder Mode

It supports AB encoder input and ABZ encoder input in High-speed position sensor (phase judgment).

- Event detection (rotation pulse) → Interrupt generation
- Event count → Match detection interrupt generation (measures the amount of transferring)
- Detects rotation direction
- Up/down-count (changeable in operation)
- Settable counter cycle

14.4.2. Sensor Mode

It supports UV Hall sensor input and UVW Hall sensor input in low-speed position sensor (zero cross judgment).

There are two kinds of sensor modes such as event count mode and timer count mode (counts with fsys).

14.4.2.1. Event Count Mode

- Event detection (rotation pulse) → Interrupt generation
- Event count → Match interrupt occurs (measuring the amount of transfer)
- Rotation direction detection

14.4.2.2. Timer Count Mode

- Event detection (rotation pulse) → Interrupt generation
- Timer count
- Rotation direction detection
- Capture function → Event capture (measures event intervals) → Interrupt generation
  Software capture
- Abnormal detection time error (timer compare) → Match detection interrupt generation
- Reverse detection error → Error flag caused by changing rotation direction

14.4.3. Timer Mode

This mode can be used as a general-purpose 24-bit timer.

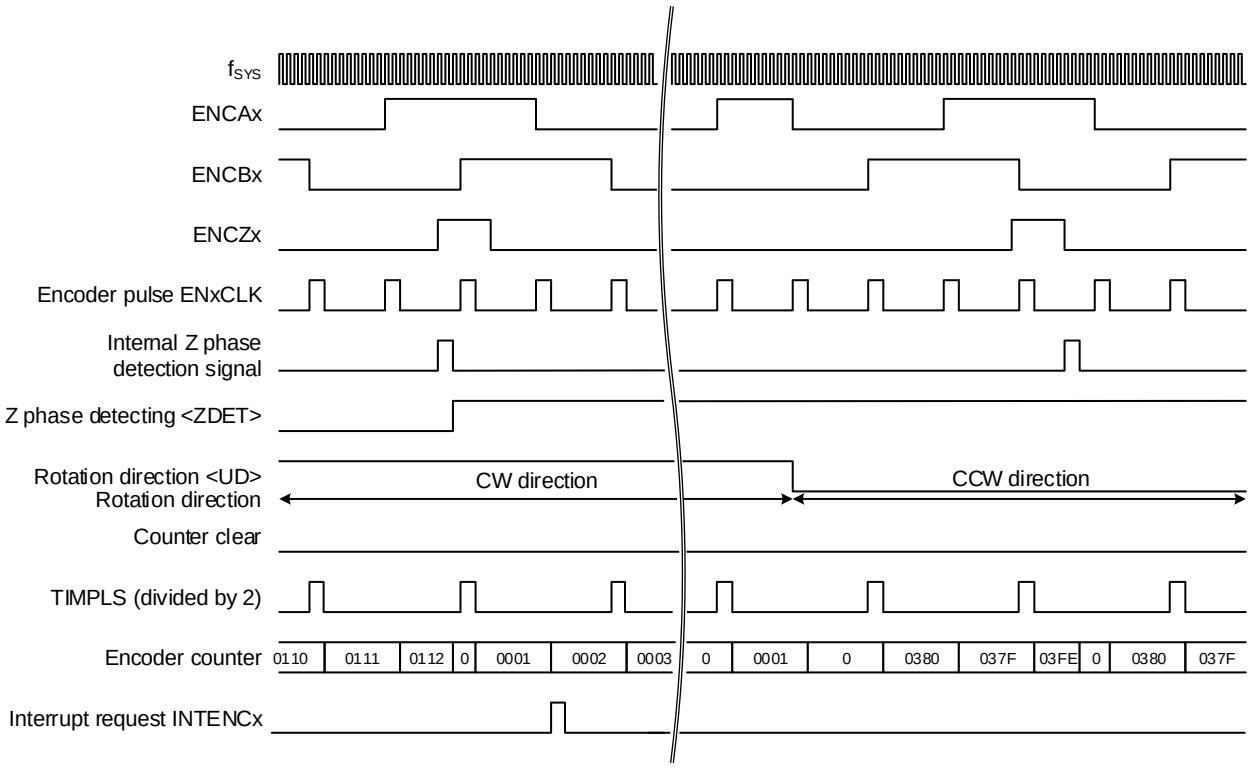
- 24-bit up counter
- Counter clear control (software clear, timer clear, external trigger and free-run count)
- Compare function → Match detection interrupt generation
- Capture function → External trigger capture → Interrupt generation
  Software capture

14.5. Function

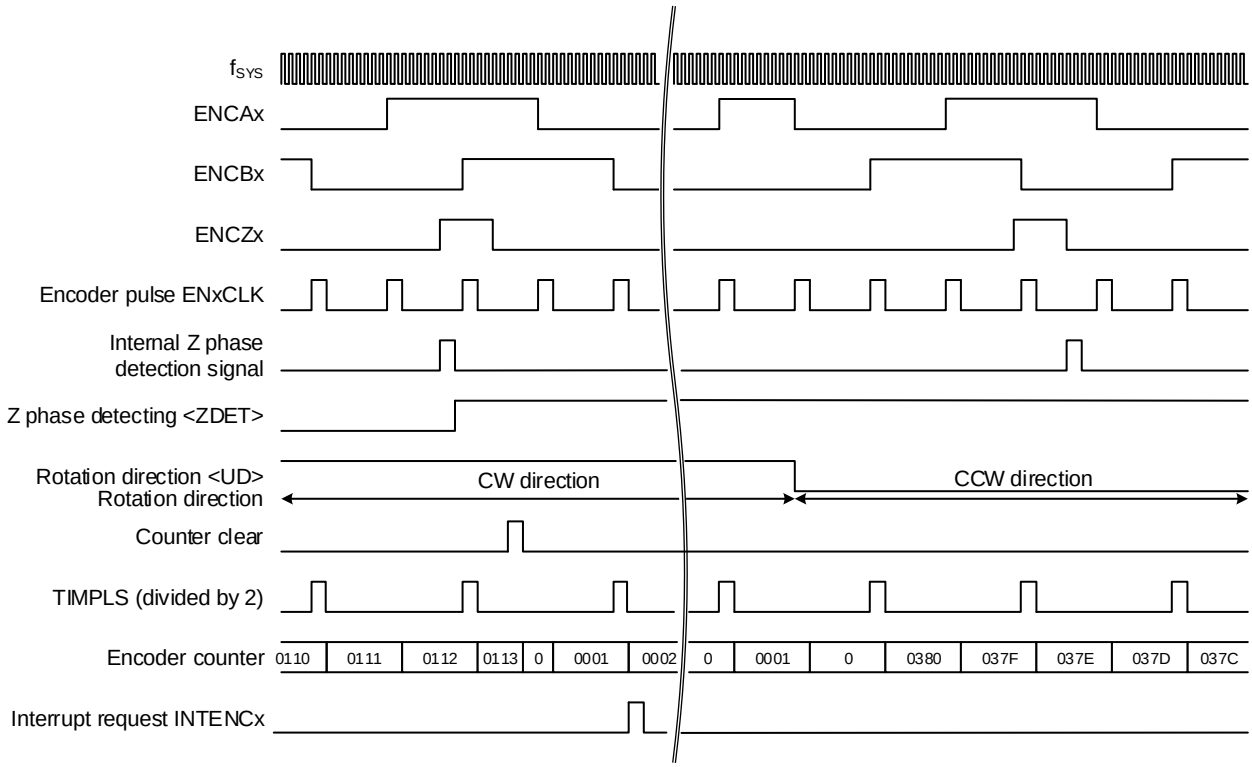
14.5.1. Mode Operation Outline

14.5.1.1. Encoder Mode

(1) When <ZEN> = "1" (<RELOAD[15:0]> = "0x0380", <INT[15:0]> = "0x0002")



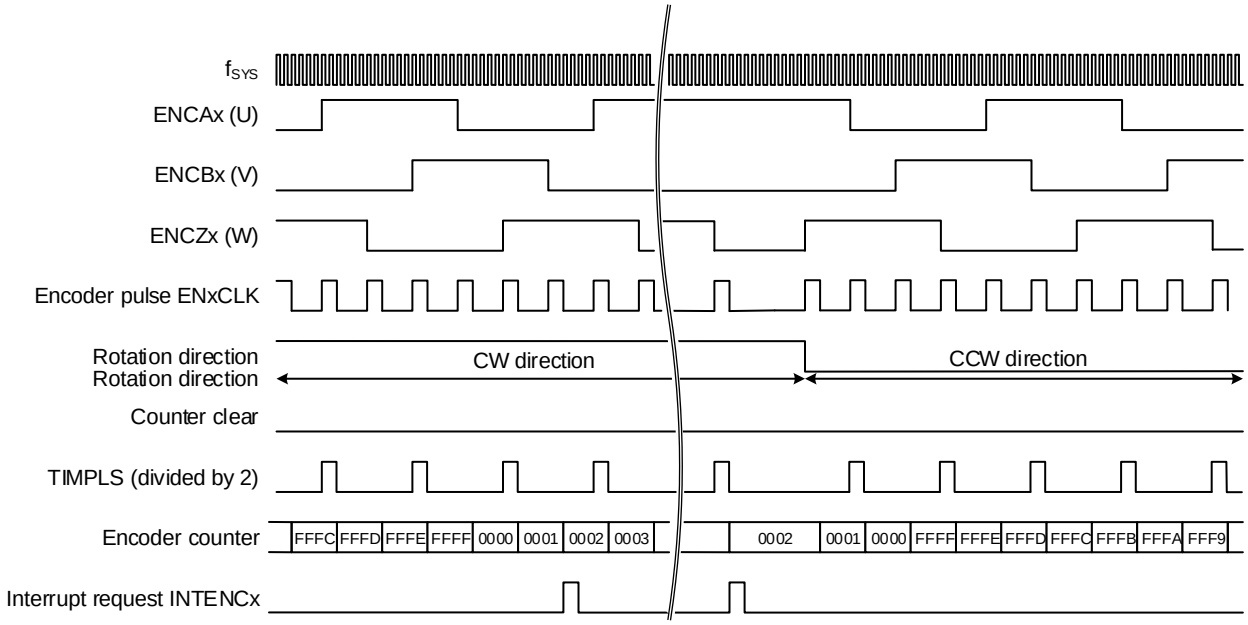
(2) When <ZEN> = "0" (<RELOAD[15:0]> = "0x0380", <INT[15:0]> = "0x0002")



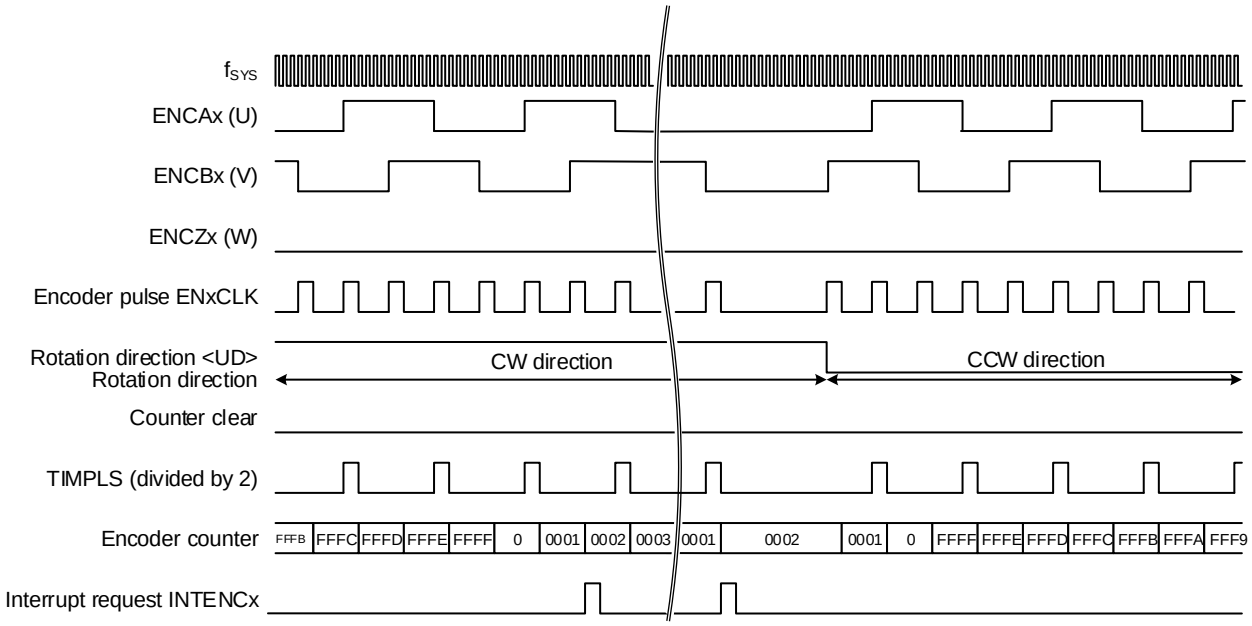
- The incremental encoder inputs of the MCU should be connected to the A, B and Z phase. The encoder counter counts pulses of ENCLK, which is multiplied by 4 clock derived from A and B signals.
- During CW rotation (i.e., A phase is 90 degrees ahead of B phase), the encoder counter counts up; when it has reached to the value of <RELOAD[15:0]>, it cleared to "0" on the next ENCLK.
- During CCW rotation (i.e., A phase is 90 degrees behind B phase), the encoder counter counts down; when it has reached to "0x0000", the value of <RELOAD [15: 0]> is set in the counter. on the next ENCLK.
- Additionally, when <ZEN> = "1", the encoder counter is cleared to "0" on the rising edge of Z-phase during CW rotation and on the falling edge of Z-phase during CCW rotation. When the ENCLK edge matches Z-phase edge, the encoder counter is cleared to "0" without counting.
- When <ENCLR> is set to "1", the encoder counter is cleared to "0".
- <UD> is set to "1" during CW rotation and is set to "0" during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK, can be taken out.
- When <CMPEN> is set to "1", an interrupt is generated when the value of the encoder counter has reached to the value of <INT[15:0]>. When <ZEN> = "1", however, an interrupt does not occur while <ZDET> = "0".
- When <ENRUN> is set to "0", <ZDET> and <UD> are cleared to "0".

14.5.1.2. Sensor Mode (Event count)

(1) When <P3EN> = "1" (<INT[15:0]> = "0x0002")



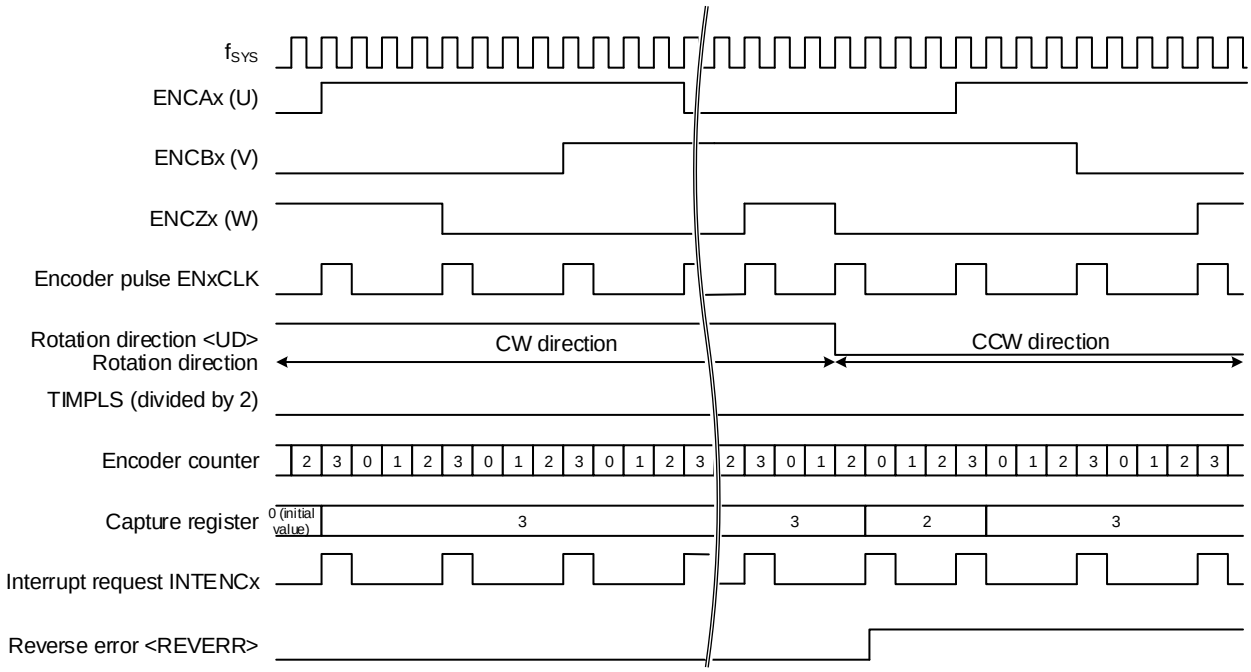
(2) When <P3EN> = "0" (<INT[15:0]> = "0x0002")



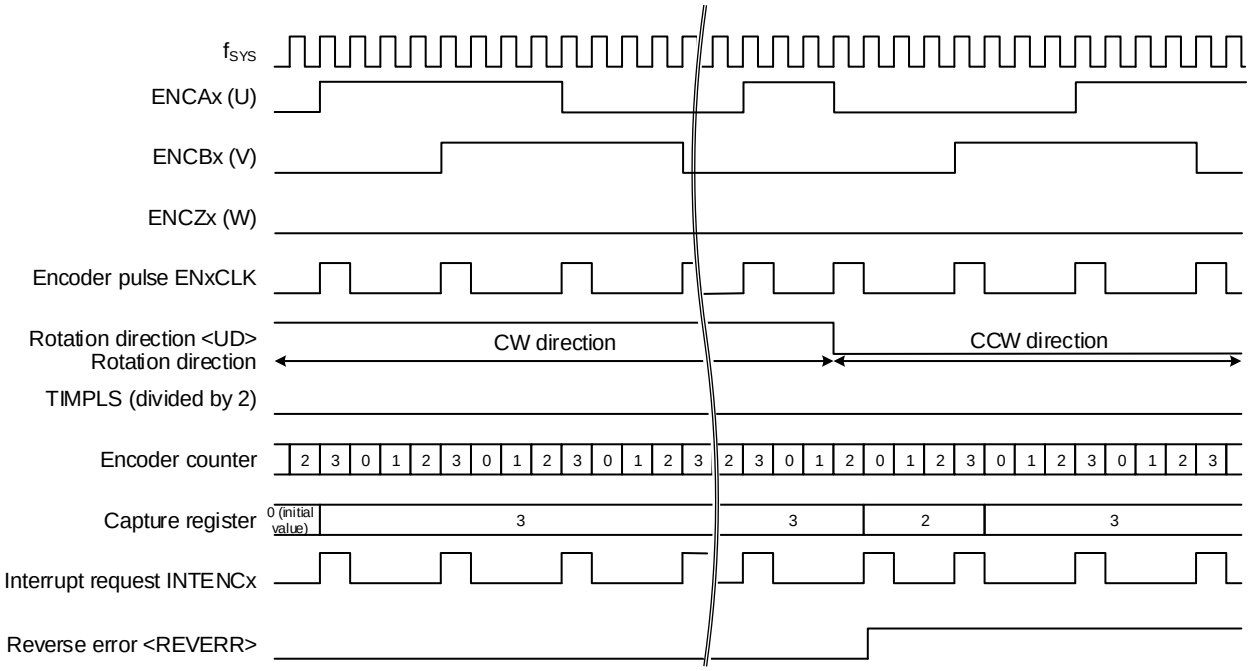
- The Hall sensor inputs of the MCU should be connected to the U, V and W -phase. The encoder counter counts the pulses of Hall sensor, which is either multiplied by 4 clock (when <P3EN> = "0") derived from U and V signals or multiplied by 6 clock (when <P3EN> = "1") derived from U, V and W signals.
- During CW rotation (i.e., U-phase is 90 degrees ahead of V-phase, V-phase is 90 degrees ahead of W-phase.), the encoder counter counts up; when it has reached to "0xFFFF", it cleared to "0" on the next ENCLK.
- During CCW rotation (i.e., U-phase is 90 degrees behind V-phase, V-phase is 90 degrees behind W-phase), the encoder counter counts down; when it has reached to "0x0000", it is set to "0xFFFF" on the next ENCLK.
- When <ENCLR> is set to "1", the counter is cleared to "0".
- <UD> is set to "1" during CW rotation and is set to "0" during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK, can be taken out.
- When <CMPEN> is set to "1", an interrupt is generated when the value of the counter has reached to the value of <INT[15:0]>.
- When <ENRUN> are set to "0", <UD> is cleared to "0".

14.5.1.3. Sensor Mode (Timer count)

(1) When <P3EN> = "1" (<INT[23:0]> = "0x0002")



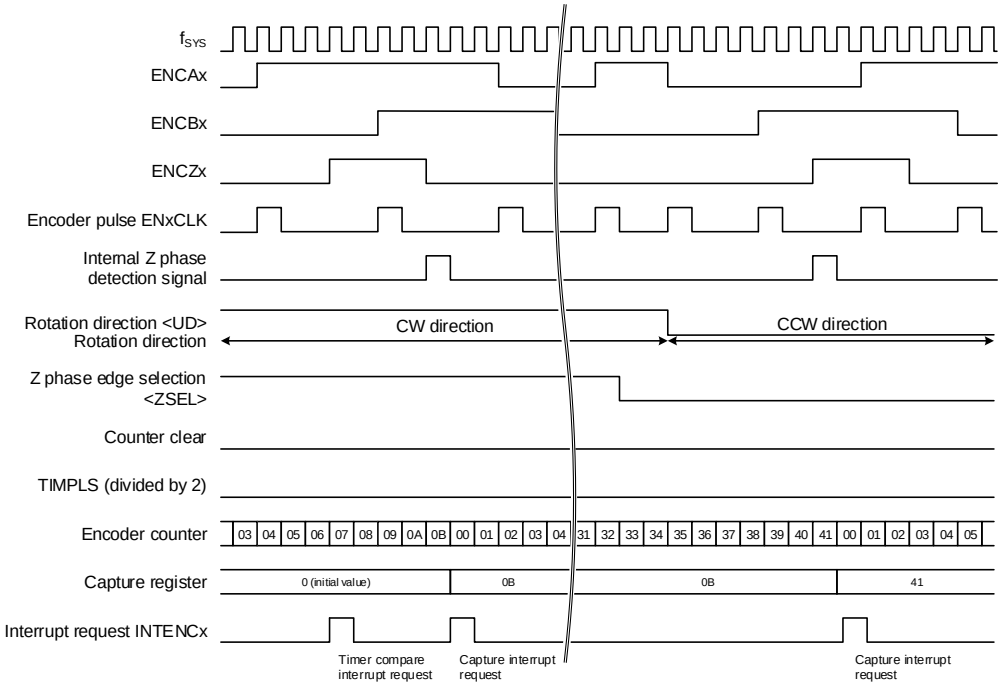
(2) When <P3EN> = "0" (<INT[23:0]> = "0x0002")



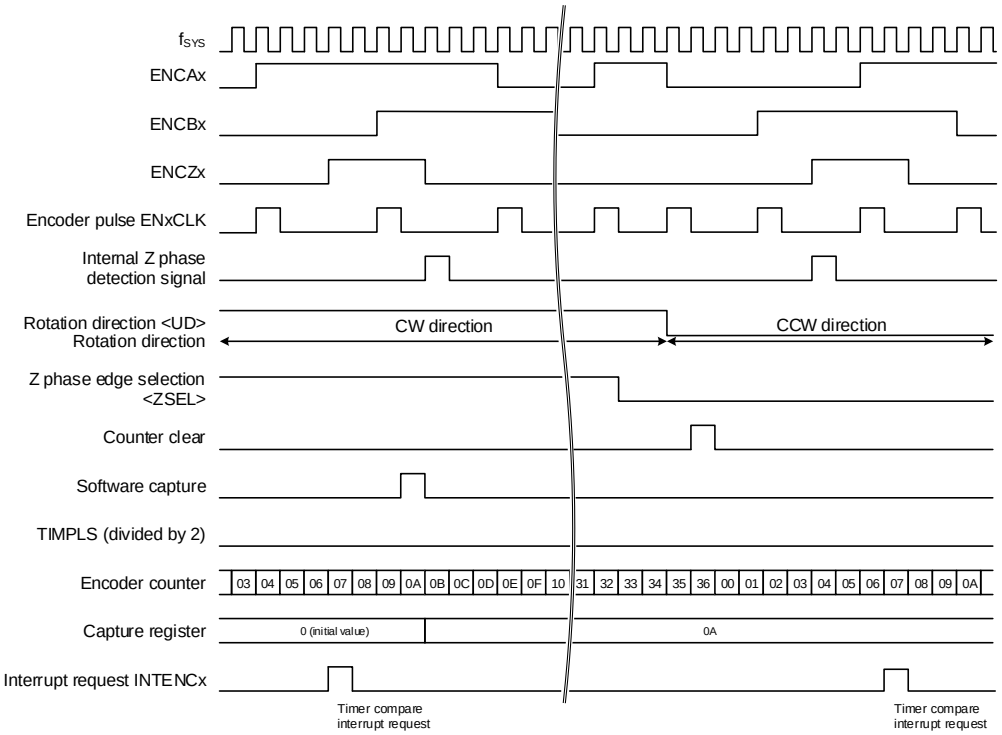
- The Hall sensor inputs of the MCU should be connected to the U, V and W -phase. The encoder counter counts the pulses of Hall sensor, which is either multiplied by 4 clock (when <P3EN> = "0") derived from U and V signals or multiplied by 6 clock (when <P3EN> = "1") derived from U, V and W signals.
- The encoder counter always counts up; it is cleared to "0" on ENCLK. When the encoder counter has reached to "0xFFFFFF", it cleared to "0".
- When <ENCLR> is set to "1", the encoder counter is cleared to "0".
- The counter value at the time is captured by ENCLK. The captured counter value can be read out from ENxCNT.
- Setting <SFTCAP> to "1", the counter value at the time is captured. This capture operation can be performed at any time. The captured counter value can be read out from ENxCNT.
- <UD> is set to "1" during CW rotation and is set to "0" during CCW rotation.
- When <CMPEN> is set to "1", an interrupt is generated when the value of the encoder counter has reached to the value of <INT[23:0]>.
- When <ENRUN> is set to "0", <UD> is cleared to "0".
- <REVERR> is set to "1" when the rotation direction has changed. This bit is cleared to "0" on reading.
- The value of the ENxCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENxCNT register is only cleared by a reset.

14.5.1.4. Timer Mode

(1) When <ZEN> = "1" (<INT[23:0]> = "0x0006")



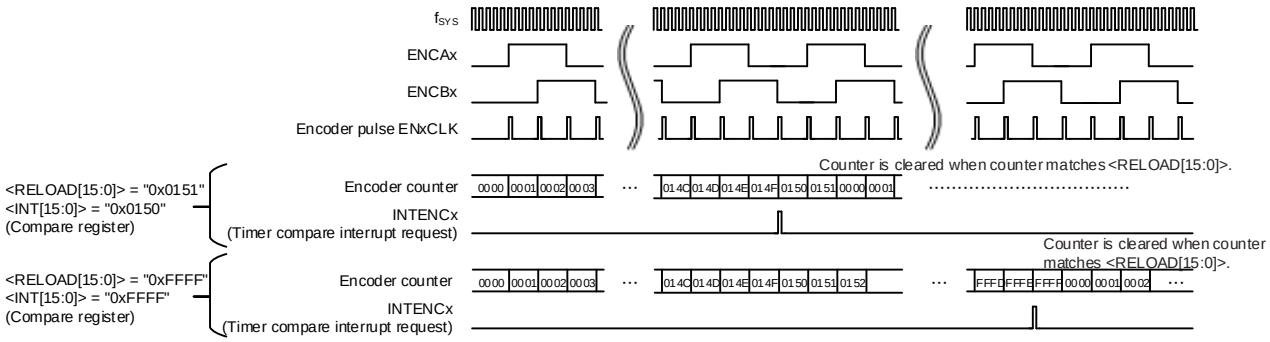
(2) When <ZEN> = "0" (<INT[23:0]> = "0x0006")



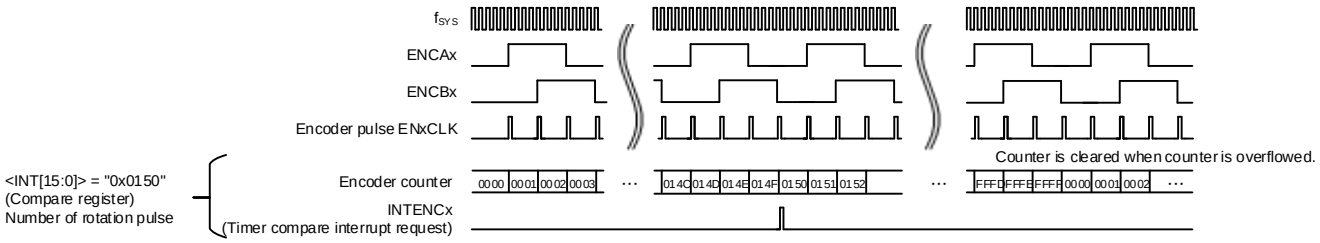
- When <ZEN> = "1", the Z input pin is used as an external trigger. When <ZEN> = "0", no external input is used.
- The encoder counter always counts up. When <ZEN> = "1", the counter is cleared to "0" on the rising edge of Z-phase when <ZESEL> is set to "0" and on the falling edge when <ZESEL> is set to "1". When the encoder counter has reached to "0xFFFFFF", it is cleared to "0".
- When <ENCLR> is set to "1", the encoder counter is cleared to "0".
- The counter value is captured by Z-phase detection at the time. The captured counter value can be read out from ENxCNT.
- Setting <SFTCAP> to "1", the encoder counter value at the time is captured. This capture operation can be performed at any time. The captured counter value can be read out from ENxCNT.
- <UD> is set to "1" during CW rotation and is set to "0" during CCW rotation.
- When <CMPEN> is set to "1", an interrupt is generated when the value of the encoder counter has reached to the value of <INT[23:0]>.
- When <ENRUN> is set to "0", <UD> is cleared to "0".
- The value of the ENxCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENxCNT register is only cleared by a reset.

14.5.2. Counter and interrupt generate operation when <CMPEN> = "1"

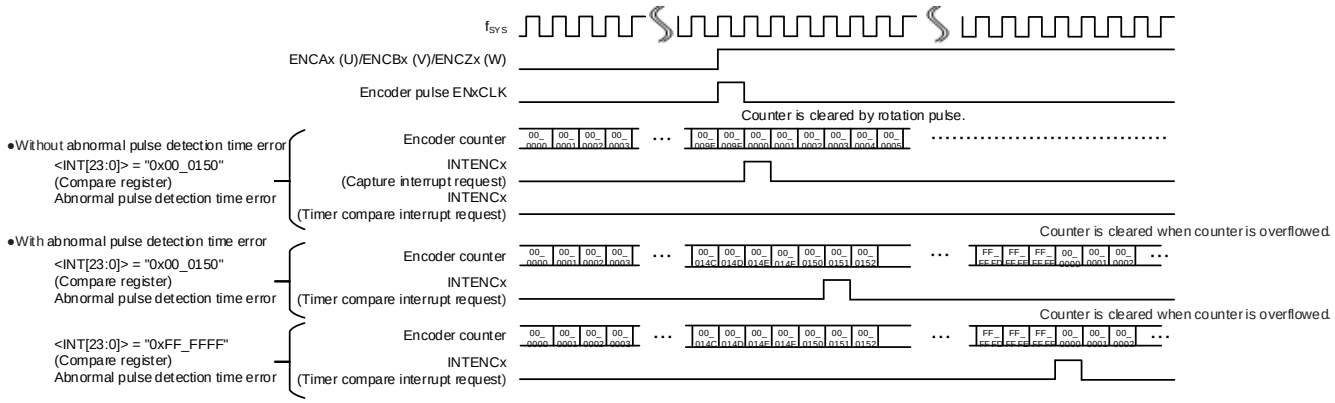
14.5.2.1. Encoder Mode



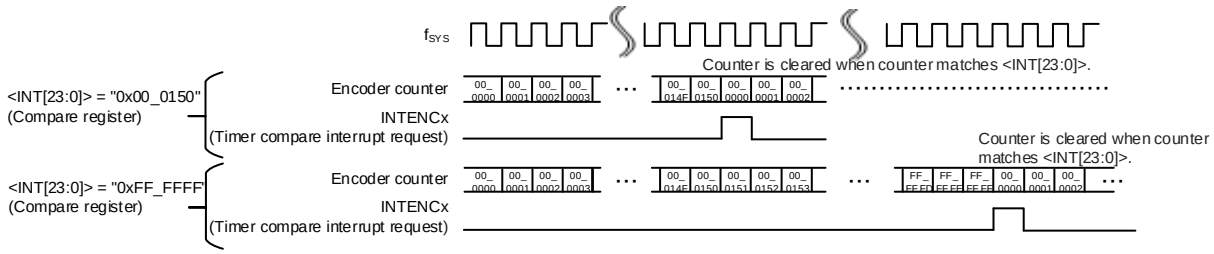
14.5.2.2. Sensor Mode (Event count)



14.5.2.3. Sensor Mode (Timer count)

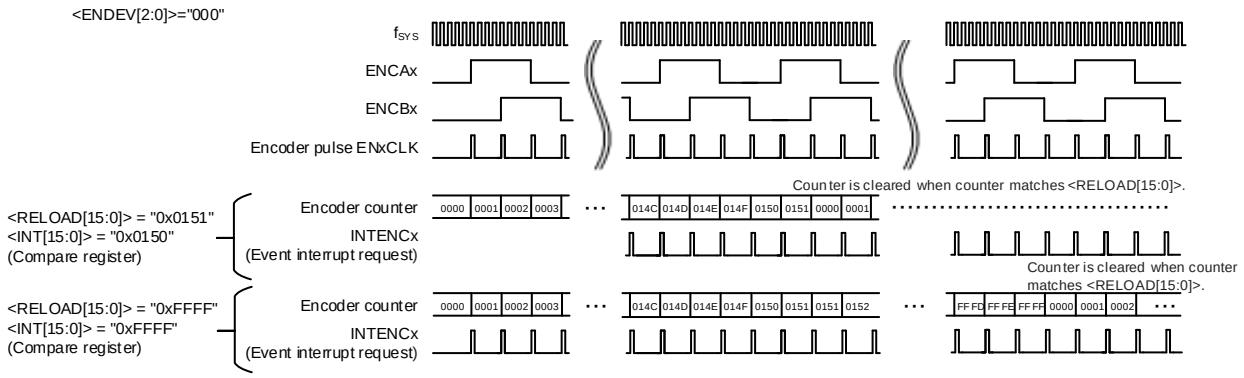


14.5.2.4. Timer mode

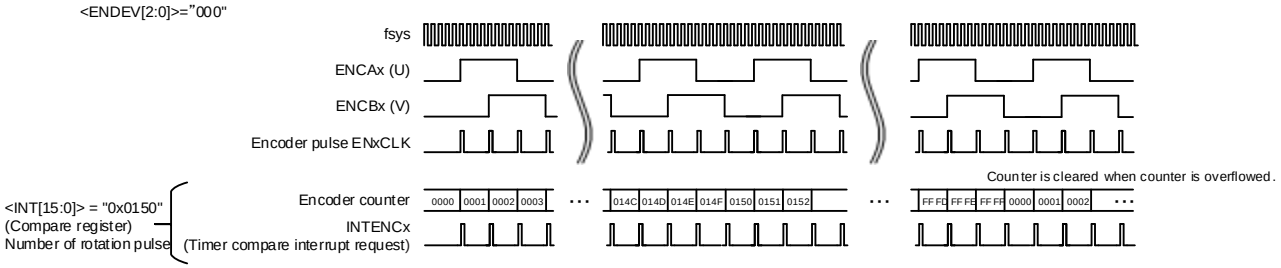


14.5.3. Counter and interrupt generate operation when <CMPEN> = "0"

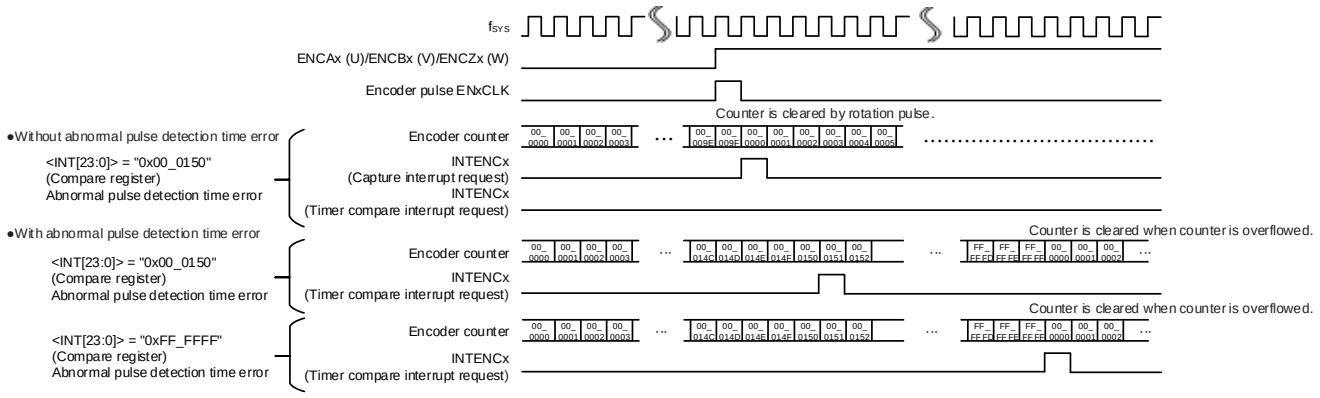
14.5.3.1. Encoder Mode



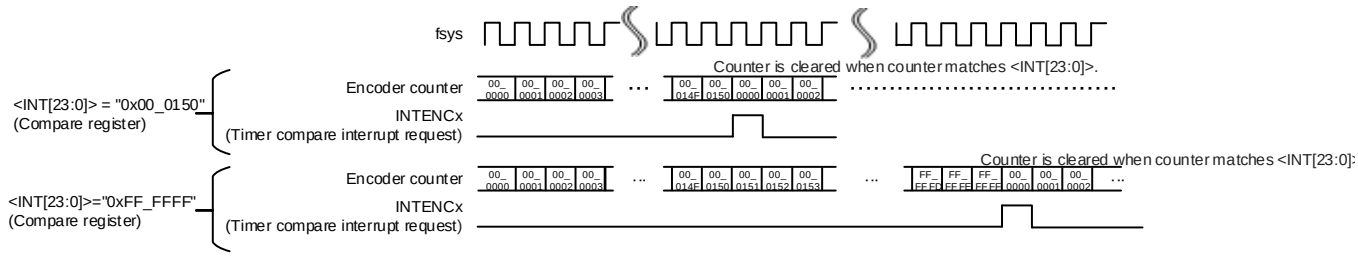
14.5.3.2. Sensor Mode (Event count)



14.5.3.3. Sensor Mode (Timer count)



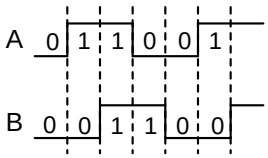
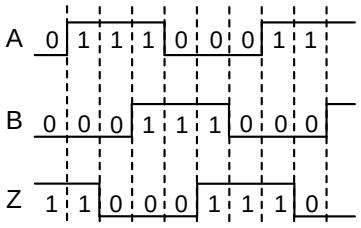
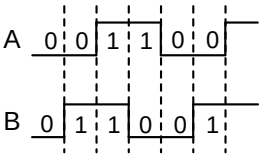
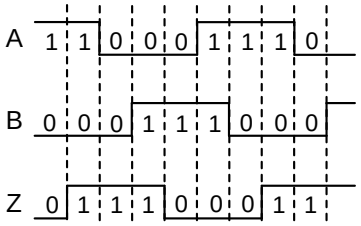
14.5.3.4. Timer Mode



14.5.4. Encoder Rotation Direction

This circuit determines a phase either A-, B- or Z-phase.

It is used as 2-phase input (A, B) and 3-phase input (A, B, Z) in common. When 3-phase input is used, set <P3EN> to "1".

	2-phase input	3-phase input
CW direction		
CCW direction		

14.5.5. Counter Circuit

The counter circuit has a 24-bit up/down counter and controls counter.

14.5.5.1. Operation Description

Depending on the operation modes, counting, clearing and reloading operation are controlled as described in Table 14.1.

Table 14.1 Counter control

Mode <MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Count	Opera-tion	Counter clear condition	Counter reload condition	Operating range of counter
Encoder mode 00	0	0	A, B	Encoder pulse (ENCLK)	UP	[1] Sets <ENCLR> to "1". [2] Matches with <RELOAD[15:0]>	-	"0x0000" to <RELOAD[15:0]>
					DOWN	[1] Sets <ENCLR> to "1".	[1] Matches with "0x0000"	
	1		A, B, Z		UP	[1] Sets <ENCLR> to "1". [2] Matches with <RELOAD[15:0]> [3] Z-trigger	-	
					DOWN	[1] Sets <ENCLR> to "1".	[1] Matches with "0x0000"	
Sensor mode (event count) 01	0	0	U, V		UP	[1] Sets <ENCLR> to "1". [2] Matches with "0xFFFF"	-	"0x0000" to "0xFFFF"
					DOWN	[1] Sets <ENCLR> to "1".	[1] Matches with "0x0000"	
		1	U, V, W		UP	[1] Sets <ENCLR> to "1". [2] Matches with "0xFFFF"	-	
					DOWN	[1] Sets <ENCLR> to "1".	[1] Matches with "0x0000"	
Sensor mode (Timer count) 10	0	0	U, V	fsys	UP	[1] Sets <ENCLR> to "1". [2] Matches with "0xFF_FFFF" [3] Encoder pulse (ENCLK)	-	"0x00_0000" to "0xFF_FFFF"
		1	U, V, W		UP		-	
Timer mode 11	0	Don't care	-		UP	[1] Sets <ENCLR> to "1". [2] Matches with "0xFF_FFFF" [3] Matches with <INT[23:0]>	-	"0x00_0000" to "0xFF_FFFF"
	1		Z		UP	[1] Sets <ENCLR> to "1". [2] Matches with "0xFF_FFFF" [3] Matches with <INT[23:0]> [4] Z-trigger	-	

Note: The counter value is not cleared by writing "0" to <ENRUN>. When <ENRUN> = "1" is set again, the counter restarts from the counter value which has stopped. If clear the counter value, write "1" to <ENCLR> to execute software clear.

14.5.6. Interrupt

The interrupt consists of including Event (divide pulse and capture) interrupt, Abnormal detecting time interrupt, Timer compare interrupt and Capture interrupt.

14.5.6.1. Operational Description

When <INTEN> is set to "1", interrupts occurs by counter value and encoder pulses.

Interrupt factor consists of following six kinds of setting for operation modes and the setting of <CMPEN> and <ZEN>.Table 14.2 shows interrupt factors.

Table 14.2 Interrupt factors

	Interrupt factor	Description	Mode	Interrupt output	Status flag
1	Event count interrupt	When <CMPEN> = "1", the encoder counter uses a counter that counts the events (rotational pulses). Then, notifies that the counter has reached to set number of times (= <INT[15:0]>).	Encoder mode and Sensor mode (event count)	<INTEN> = "1" and <CMPEN> = "1"	<CMP>
2	Event interrupt (divide pulse)	The occurrence of an event (encoder pulse) is divided by 1 to 128 according to the <ENDEV> setting and notified.		<INTEN> = "1"	Not available
3	Event interrupt (capture interrupt)	Notifies that an event (encoder pulse) has occurred and that a capture has executed in an event (rotational pulse).	Sensor mode (Timer count)	<INTEN> = "1"	Not available
4	Abnormal detection time error interrupt	When <CMPEN> = "1", the ENC uses a counter that counts up with fsys and is cleared by an event (encoder pulse). Then, notifies that the event does not occur for more than certain period of time (= <INT[23:0]>).		<INTEN> = "1" and <CMPEN> = "1"	<CMP>
5	Timer compare interrupt	When <CMPEN> = "1", notifies that the counter has reached to set time (= <INT[23:0]>).	Timer mode	<INTEN> = "1" and <CMPEN> = "1"	<CMP>
6	Capture interrupt	Notifies that the capture was done with an external trigger (ENCZx input).		<INTEN> = "1"	Not available

In Sensor mode (Timer Count) and Timer mode, the captured operation of the encoder counter is possible.

The captured counter value can be read out from the <CNT[23:0]>.

In Sensor mode (Timer Count), when the event (encoder pulse) occurs, the value of the counter is captured. Software capture is also possible by writing "1" to <SFTCAP>.

In Timer mode, software capture is possible by writing "1" to <SFTCAP>. When <ZEN> is set to "1", external trigger capture at the edge according to <ZESEL> is also possible by using the ENCZx input.

15. Power-on Reset circuit (POR)

The Power-on reset circuit (POR) generates the Power-on reset signal when power-on.
It also generates the Power-on reset signal when the power supply voltage is lower than it's detection voltage.
The power supply voltage means DVDD5 and RVDD5.

Note1: The POR may not operate properly due to fluctuations of supply voltage. It requires consideration based on the electrical characteristic when designing the equipment.

Note2: The Power-on reset signal is undefined when the power supply voltage is lower than the operating limit voltage (the voltage which the reference voltage generation circuit cannot operate).

15.1. Block Diagram

The POR consists of the reference voltage generation circuit, comparators, and the Power-on counter.
This circuit compares a voltage divided by the ladder resistor with a reference voltage generated in the reference voltage generation circuit by the comparator.

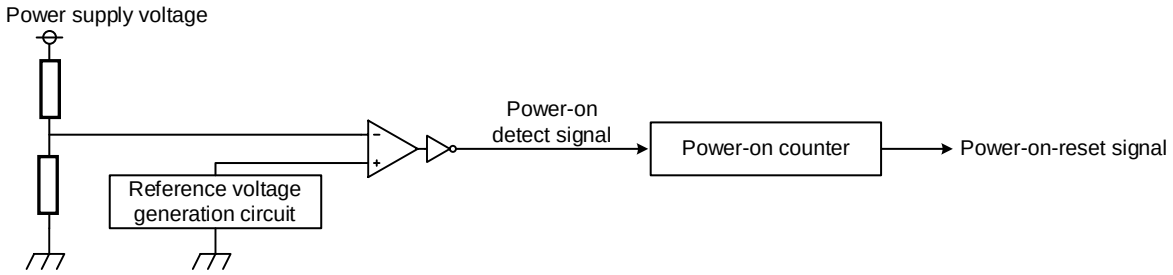


Figure 15.1 Block Diagram of POR

15.2. Function

At power on, the Power-on detection signal generates while the power supply voltage is the POR release voltage or lower. When the power supply voltage is higher than the POR release voltage and the Power-on detect signal is released, the Power-on counter operates. The power-on reset signal is released after up to 3.7ms elapses.
At power off, the Power-on reset signal is generated when the power supply voltage is the detect voltage of the POR or lower.
While the Power-on reset signal is generated, the Power-on counter, CPU, and peripheral functions are reset.

When only Power-on reset is used without RESET by $\overline{\text{RESET}}$ pin, the power supply voltage must be increased to the operating voltage range within 3ms after the power supply voltage exceeds the POR release voltage. If not, TMPM370FYDFG/TMPM370FYFG will not operate correctly.

Note: When the supply voltage rises, until power supply voltage reaches the operating voltage range (4.5V to 5.5V) and 200μs elapses, Port L (PL0 and PL1) must be opened or input voltage to Port L must be 0.5 V or less.

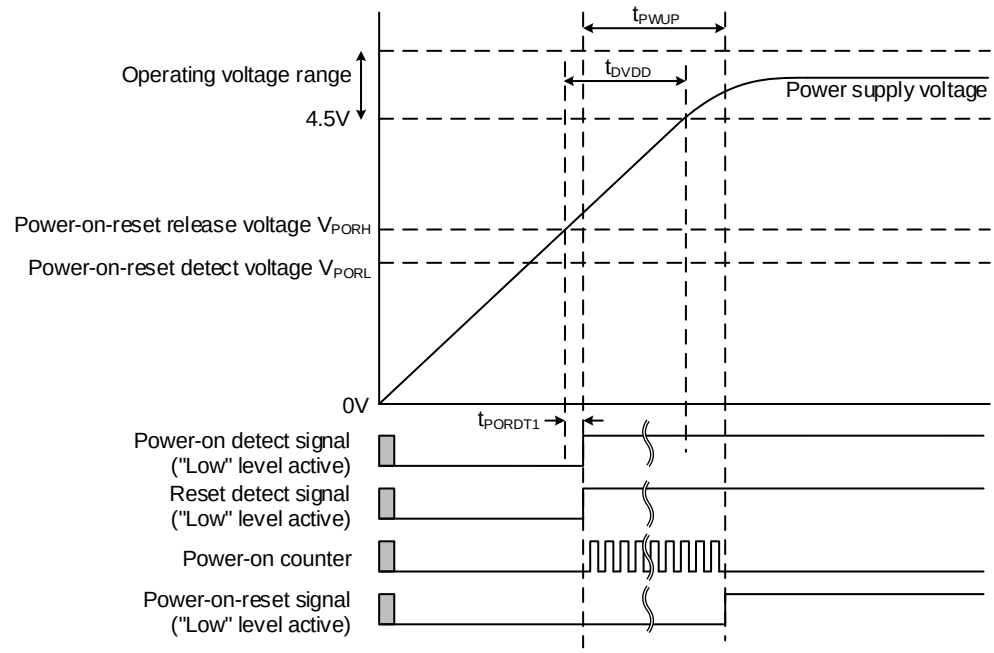


Figure 15.2 Operation Timing of Power-on reset (1/2)

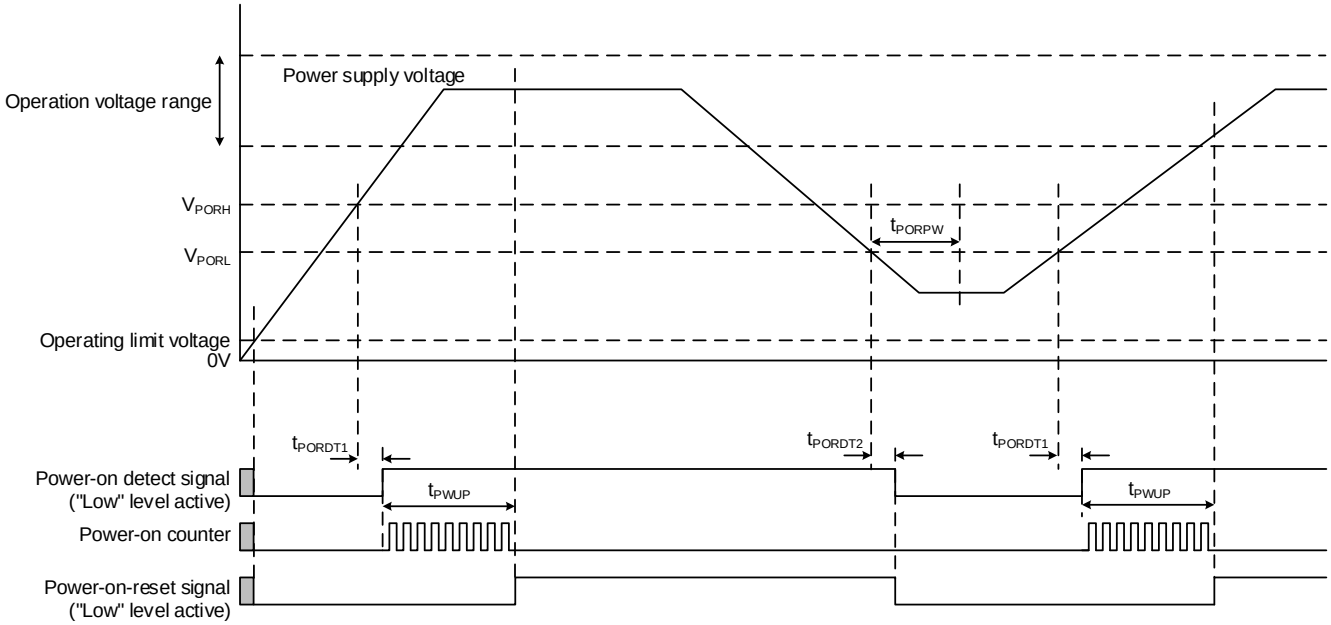


Figure 15.3 Operation Timing of Power-on reset (2/2)

Parameter	Symbol	Min	Typ.	Max	Unit
Power-on reset release time	t_{PWUP}	-	-	3.7	ms
Rising time of power supply voltage	t_{DVDD}	-	-	3	
Power-on reset release voltage	V_{PORH}	2.8	3	3.2	V
Power-on reset detect voltage	V_{PORL}	2.6	2.8	3	V
Power-on reset release response time	t_{PORDT1}	-	30	-	μs
Power-on reset detect response time	t_{PORDT2}	-	30	-	μs
Minimum pulse width of Power-on reset	t_{PORPW}	45	-	-	μs

Note: The Power-on reset release voltage (V_{PORH}) is always higher than the Power-on reset detect voltage (V_{PORL}), since they relatively change.

16. Voltage Detection circuit (VLTD)

16.1. Outline

The voltage detection circuit detects a decrease in the power supply voltage and generates a reset signal.
Power supply voltage means DVDD5, RVDD5.

Note: Depending on the fluctuation of the power supply voltage, the voltage detection circuit may not operate properly. When designing the device, careful consideration must be given to the electrical characteristics.

16.2. Block Diagram

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator, and control registers.
The power supply voltage is divided by the ladder resistor and input to the detection voltage selection circuit. A voltage corresponding to the detection voltage (VDCR<VDLVL[1:0]>) is selected by the detection voltage selection circuit and compared with the reference voltage by the comparator. VLTD generates a voltage detection reset when the power supply voltage is less than the detection voltage (VDCR<VDLVL[1:0]>).

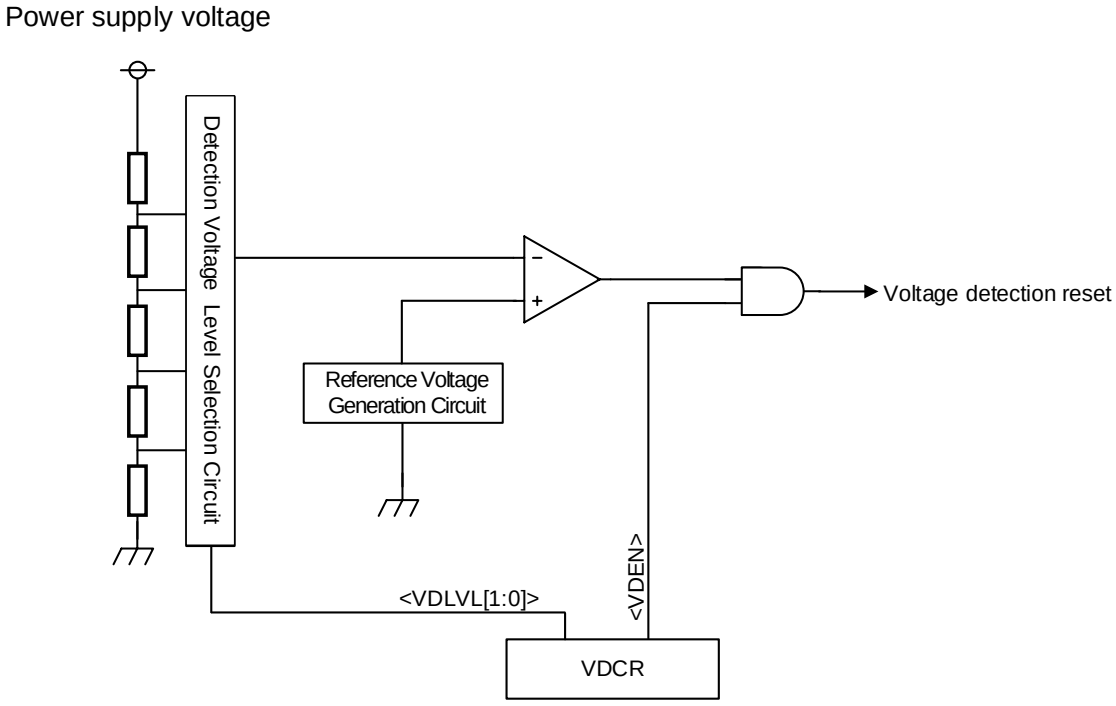


Figure 16.1 Voltage Detection Circuit

16.3. Registers

16.3.1. List of Registers

Register name		Address (Base+)
Voltage detection control register	VDCR	0x0000

16.3.2. VDCR (Voltage Detection Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	VDLVL		VDEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:3	-	R	Read as "0".
2:1	VDLVL[1:0]	R/W	Detection voltage level selection 00: Prohibited 01: 4.1 ± 0.2V 10: 4.4 ± 0.2V 11: 4.6 ± 0.2V
0	VDEN	R/W	Voltage detection enabled/disabled 0: Disabled 1: Enabled

Note: VDCR is initialized by the POR/external reset input.

16.4. Operations

16.4.1. Control

The voltage detection circuit is controlled by the voltage detection control register.

16.4.2. Function

The voltage detection circuit is set by setting VDCR<VDLVL[1:0]> and VDCR<VDEN >. When the voltage detection is enabled, if the power supply voltage is less than the detection voltage (VDCR<VDLVL[1:0]>), voltage detection reset is generated.

16.4.2.1. Enable/disable voltage detection operation

VDCR<VDEN> is cleared to "0" (i.e., VLTD is disabled) after power-on reset and external reset release. When this bit is set to "1", voltage detection operation is enabled.

Note: When the value of VDCR<VDEN> is changed from "0" (disabled) to "1" (enabled) with the power supply voltage < detection voltage VDCR<VDLVL[1:0]>, a voltage detection reset occurs immediately.

16.4.2.2. Detection voltage level selection

Select the detect voltage by VDCR<VDLVL[1:0]>.

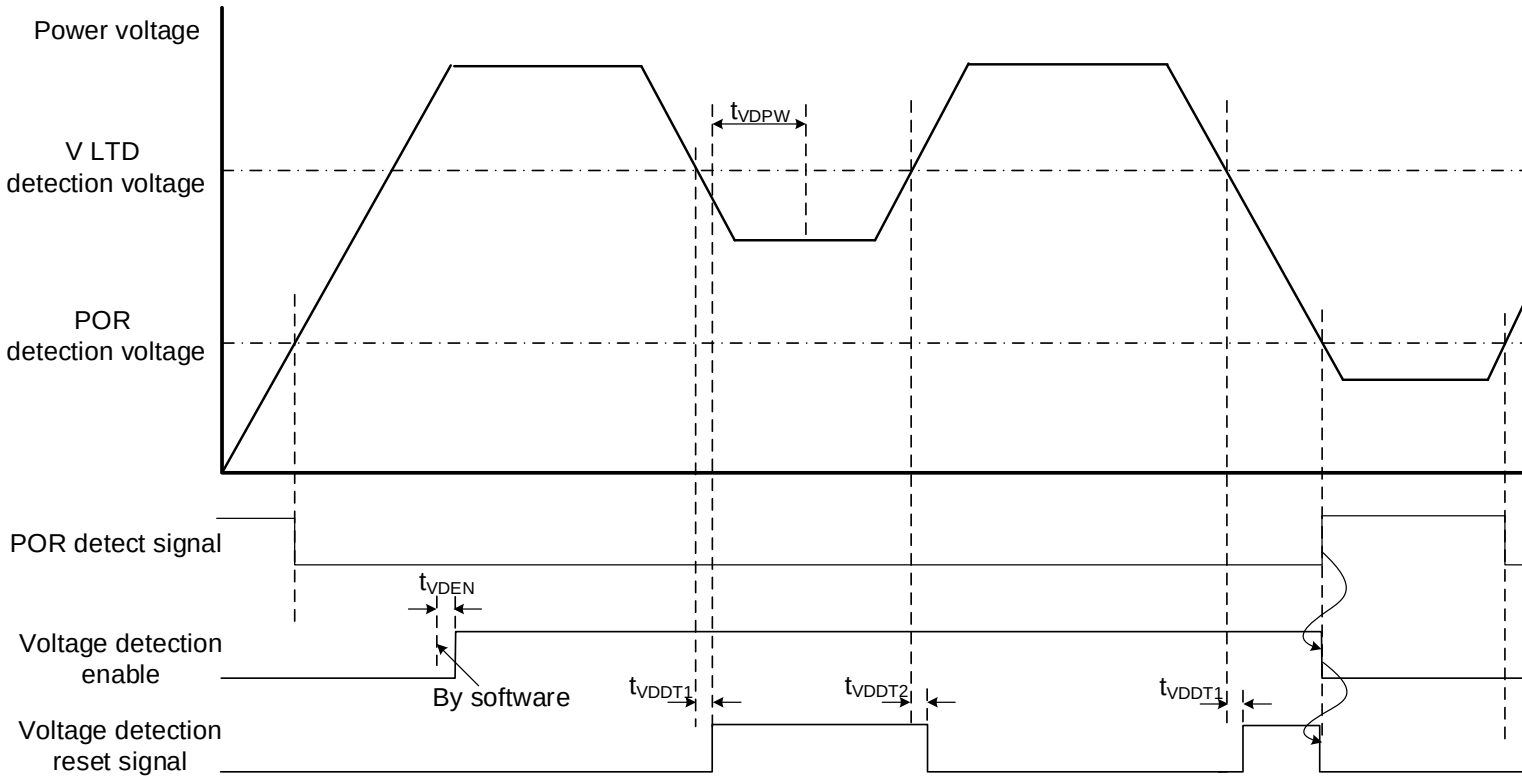


Figure 16.2 Voltage detection timing

Table 16.1 Voltage detection circuit characteristics

Parameter	Symbol	Min	Typ.	Max	Unit
Time when the VLTD is enabled	t_{VDEN}	-	40	-	μs
VLTD detection response time	t_{VDDT1}	-	40	-	
VLTD detection release time	t_{VDDT2}		40	-	
VLTD minimum pulse width	t_{VDPW}	45	-	-	

17. Oscillation Frequency Detector Circuit (OFD)

17.1. Configuration

The oscillation frequency detector circuit (OFD) generates a reset if the external high-speed frequency for CPU clock is out of the detection frequency range.

The upper and lower limits of the frequency to be detected are set by the OFDMXPLLOFF and OFDMNPLLOFF. Refer to Figure 17.1, Figure 17.2 for the initial detection frequency of TMPM370FYDFG/TMPM370FYFG.

When the OFD is enabled, writing to OFDMNPLLOFF/OFDMNPLLON/ OFDMXPLLOFF/OFDMXPLLON is disabled. Therefore, writing the detection frequency to these registers should be done when the OFD is disabled.

And to write to OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON, the write enable code "0xF9" should be written to OFDCR1 beforehand.

The OFD is disabled by the external reset input to the $\overline{\text{RESET}}$ pin.

To enable the OFD, set OFDCR2 to "0xE4" after setting OFDCR1 to "0xF9".

When the TMPM370FYDFG/TMPM370FYFG detects the out of frequency range by OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF, and OFDMXPLLON, a frequency detection reset occurs. Then, all I/O port become high impedance. Since the internal circuit such as the CPU is initialized by the reset synchronized with the clock, when the frequency detection reset occurs due to the stop of the high frequency clock, the internal circuit will not be initialized until the oscillation of the high frequency clock is restarted.

All registers of the frequency detection circuit (OFDCR1, OFDCR2, OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF, OFDMXPLLON) are not initialized by the frequency detection reset, and the detection operation continues even during the period when the frequency detection reset occurs. Therefore, when the frequency detection reset occurs, the reset will not be released until the high frequency clock becomes normal.

Note1: It is not guaranteed that OFD can detect all defects at any time, and it is not a circuit to measure error frequency.

Note2: The OFD is enabled only in NORMAL and IDLE modes. In STOP mode, the OFD is disabled automatically.

Note3: When the PLL is controlled by the CGPLLSEL, the OFD must be disabled beforehand. If OFD reset is generated with PLL-ON, the detection frequency setting registers (OFDMNPLLON/ OFDMXPLLON) are automatically switched over to OFDMNPLLOFF/OFDMXPLLOFF.

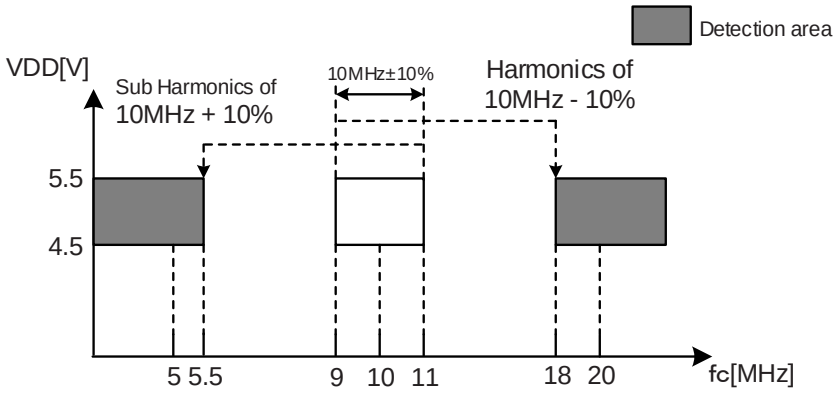


Figure 17.1 Initial Value of Detection Frequency Range (PLL OFF)

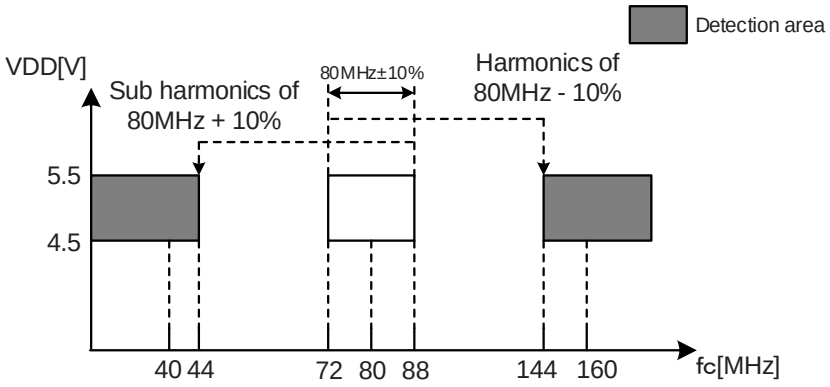


Figure 17.2 Initial Value of Detection Frequency Range (PLL ON)

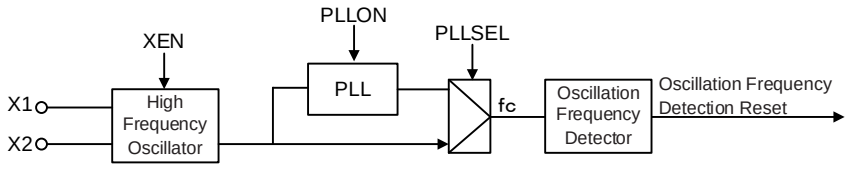


Figure 17.3 Oscillation Frequency Detector Circuit

Note: When setting the PLL by the CGPLLSEL register, be sure to disable OFD.

17.2. Registers

17.2.1. Registers List

The control registers and its addresses are as follows.

Register name		Address (Base+)
Oscillation frequency detection control register 1	OFDCR1	0x0000
Oscillation frequency detection control register 2	OFDCR2	0x0004
Lower detection frequency setting register (PLL OFF)	OFDMNPLLOFF	0x0008
Lower detection frequency setting register (PLL ON)	OFDMNPLLON	0x000C
Higher detection frequency setting register (PLL OFF)	OFDMXPLLOFF	0x0010
Higher detection frequency setting register (PLL ON)	OFDMXPLLON	0x0014

17.2.2. OFDCR1 (Oscillation frequency detection control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	OFDWEN							
After reset	0	0	0	0	0	1	1	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:0	OFDWEN[7:0]	R/W	Register write control 0x06: Disable 0xF9: Enable Writing "0xF9" enables to write registers except OFDCR1. When writing a value except "0x06" or "0xF9", "0x06" is written. Even if writing to registers is disabled, reading from each register is enabled

Note: OFDCR1 is initialized by the external reset ("Low" level input to the $\overline{\text{RESET}}$ pin).

17.2.3. OFDCR2(Oscillation frequency detection control register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	OFDEN							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:8	-	R	Read as "0".
7:0	OFDEN[7:0]	R/W	Frequency detection operation control 0x00: Disable 0xE4: Enable When writing a value except "0x00" or "0xE4", the written value is invalid and will not be changed.

Note: OFDCR2 is initialized by the external reset ("Low" level input to the $\overline{\text{RESET}}$ pin).

17.2.4. OFDMNPLLOFF (Lower detection frequency setting register (PLL OFF))

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	OFDMNPLL OFF
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	OFDMNPLLOFF							
After reset	0	0	0	1	1	1	1	1

Bit	Bit Symbol	Type	Function
31:9	-	R	Read as "0".
8:0	OFDMNPLL OFF[8:0]	R/W	Sets lower limit detection frequency.

Note1: This register cannot be written when frequency detection operation is permitted.

Note2: OFDMNPLLOFF is initialized by the external reset ("Low" level input to the $\overline{\text{RESET}}$ pin).

Note3: Specify an appropriate value to OFDMNPLLOFF and OFDMXPLLOFF depending on the clock frequency to be used under the condition of OFDMNPLLOFF<OFDMXPLLOFF. For how to calculate the value, refer to "17.3.2. Setting of Detection Frequency".

Note4: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

17.2.5. OFDMNPLLON (Higher detection frequency setting register (PLL ON))

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	OFDMNPLLON
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	OFDMNPLLON							
After reset	1	1	1	1	0	0	0	1

Bit	Bit Symbol	Type	Function
31:9	-	R	Read as "0".
8:0	OFDMNPLLON[8:0]	R/W	Sets lower limit detection frequency.

Note1: This register cannot be written when frequency detection operation is permitted.

Note2: OFDMNPLLON is initialized by the external reset ("Low" level input to the $\overline{\text{RESET}}$ pin).

Note3: Specify an appropriate value to OFDMNPLLON and OFDMXPLLON depending on the clock frequency to be used under the condition of OFDMNPLLON<OFDMXPLLON. For how to calculate the value, refer to "17.3.2. Setting of Detection Frequency"

Note4: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

17.2.6. OFDMXPLLOFF (Higher detection frequency setting register (PLL OFF))

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	OFDMXPLL OFF
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	OFDMXPLLOFF							
After reset	0	0	1	1	0	1	1	1

Bit	Bit Symbol	Type	Function
31:9	-	R	Read as "0".
8:0	OFDMXPLL OFF[8:0]	R/W	Sets higher limit detection frequency.

Note1: This register cannot be written when frequency detection operation is permitted.

Note2: OFDMXPLLOFF is initialized by the external reset ("Low" level input to the $\overline{\text{RESET}}$ pin).

Note3: Specify an appropriate value to OFDMNPLLOFF and OFDMXPLLOFF depending on the clock frequency to be used under the condition of OFDMNPLLOFF<OFDMXPLLOFF. For how to calculate the value, refer to "17.3.2. Setting of Detection Frequency".

Note4: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

17.2.7. OFDMXPLLON (Higher detection frequency setting register (PLL ON))

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	OFDMXPLL ON
After reset	0	0	0	0	0	0	0	1
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	OFDMXPLLON							
After reset	1	1	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:9	-	R	Read as "0".
8:0	OFDMXPLL ON[8:0]	R/W	Sets higher limit detection frequency.

Note1: This register cannot be written when frequency detection operation is permitted.

Note2: OFDMXPLLON is initialized by the external reset ("Low" level input to the $\overline{\text{RESET}}$ pin).

Note3: Specify an appropriate value to OFDMNPLLON and OFDMXPLLON depending on the clock frequency to be used under the condition of OFDMNPLLON<OFDMXPLLON. For how to calculate the value, refer to "17.3.2. Setting of Detection Frequency"

Note4: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

17.3. Function

17.3.1. Operation Control of Oscillation Frequency Detection Circuit

Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection, and writing "0x00" to OFDCR2 with OFDCR1="0xF9" disables the oscillation frequency detection.
When the write-protect code "0x06" is set in OFDCR1, writing to OFDCR2 will not be possible. OFDCR2 can be read regardless of the OFDCR1 setting. OFDCR1 is initialized to "0x06" and OFDCR2 is initialized to "0x00" by the external reset input ("L" input to the $\overline{\text{RESET}}$ pin), so the frequency detection operation is stopped after the external reset is released and OFD becomes register writing disable state. OFDCR1 and OFDCR2 are not initialized by internal factor reset (SYSRESETREQ reset, watchdog timer reset) and frequency detection reset.

Note: After writing data to OFDCR2, set "0x06" to OFDCR1 to protect registers.

When the frequency detection operation is enabled (OFDCR2 = "0xE4") and the STOP mode is executed, the oscillation frequency detection circuit is automatically disabled. When the STOP mode is released in this state, the frequency detection circuit will be permitted after the STOP mode is released and the warming-up period elapses.

The oscillation frequency detection circuit is available only in NORMAL and IDLE mode. Refer to Table 17.1 for details.

Table 17.1 Status of OFD in each Operation Mode

Operation mode or state	Oscillation Frequency Detector operation (OFDCR2 = "0xE4")	Pin status by frequency detection reset (Except power supply, $\overline{\text{RESET}}$, X1, X2 pins)
NORMAL	Available	Hi-Z
IDLE	Available	Hi-Z
STOP (Including warming-up period)	The OFD is disabled automatically.	
Reset by OFD	Available	Hi-Z
Watchdog timer reset SYSRESETREQ reset	Available	Hi-Z
RESET by external reset ("Low" level input to the $\overline{\text{RESET}}$ pin)	Unavailable	-

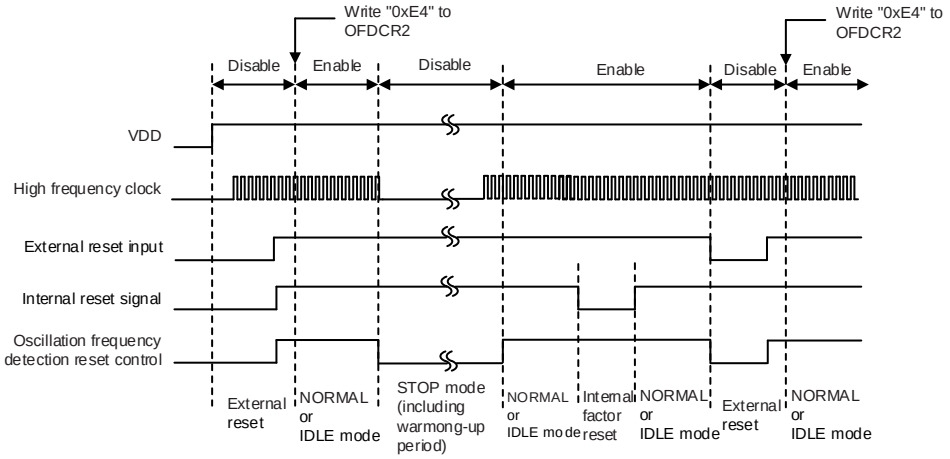


Figure 17.4 Operation of Oscillation Frequency Detection Circuit in each Mode

17.3.2. Setting of Detection Frequency

Table 17.2 shows OFDMXPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMNPLLON and setting value.

Table 17.2 Used Frequency and Setting Value

Oscillation frequency		Detection range [MHz]	Undetectable range [MHz]	Setting value	Decimal	(Hex.)
8MHz (PLL OFF)	lower limit	≤4.12	≥7.76	OFDMNPLLOFF	24	(0x18)
	upper limit	≥15.52	≤8.24	OFDMXPLLOFF	47	(0x2F)
10MHz (PLLOFF)	lower limit	≤5.15	≥9.7	OFDMNPLLOFF	30	(0x1E)
	upper limit	≥19.4	≤10.3	OFDMXPLLOFF	59	(0x3B)
64MHz (PLL ON)	lower limit	≤32.96	≥62.08	OFDMNPLLON	190	(0xBE)
	upper limit	≥124.16	≤65.92	OFDMXPLLON	379	(0x17B)
80MHz (PLL ON)	lower limit	≤41.2	≥77.6	OFDMNPLLON	238	(0xEE)
	upper limit	≥155.2	≤82.4	OFDMXPLLON	473	(0x1D9)

17.3.3. Oscillation Frequency Detection Reset

When a frequency lower than the value set by OFDMNPLLON/OFDMNPLLOFF or higher than the value set by OFDMXPLLON/OFDMXPLLOFF is detected, TMPM370FYDFG/TMPM370FYFG will generate a frequency detection reset that initializes all input/output ports.

(1) When the high frequency oscillation becomes abnormal

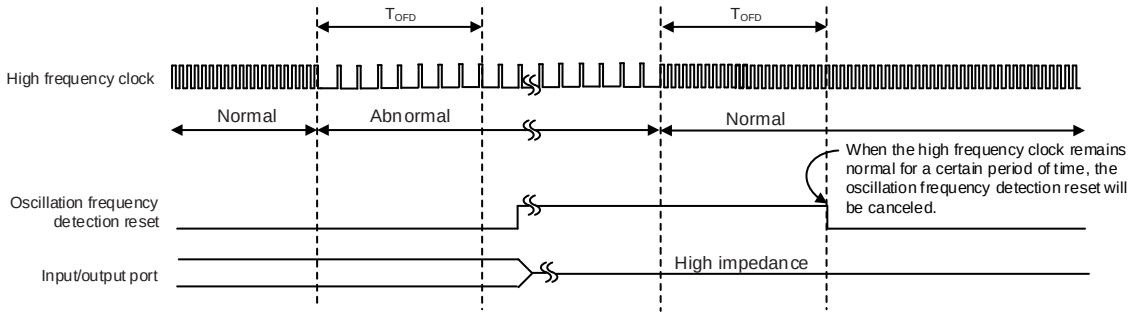
When the high frequency becomes abnormal for a certain period (T_{OFD}), the oscillation frequency detection reset will occur. Frequency detection reset initializes all input/output ports except the power supply pin and $\overline{\text{RESET}}$ pin to the high impedance state, and also initializes the internal circuit such as CPU.

(2) When the high frequency oscillation stops

When the high frequency is stopped for a certain period of time (T_{OFD}), a frequency detection reset will occur. Frequency detection reset initializes all input/output ports except the power supply pin and $\overline{\text{RESET}}$ pin to put them in a high impedance state. Since the internal circuit such as the CPU is initialized by the signal latched by the high frequency clock, the internal circuit retains the state when the frequency detection reset occurs while the high frequency is stopped. After that, when the high frequency oscillation restarts, the internal circuit is also initialized.

When the high frequency remains normal for a certain period of time (T_{OFD}), the oscillation frequency detection reset will be canceled.

- When the high frequency clock becomes abnormal



- When the high frequency clock stops

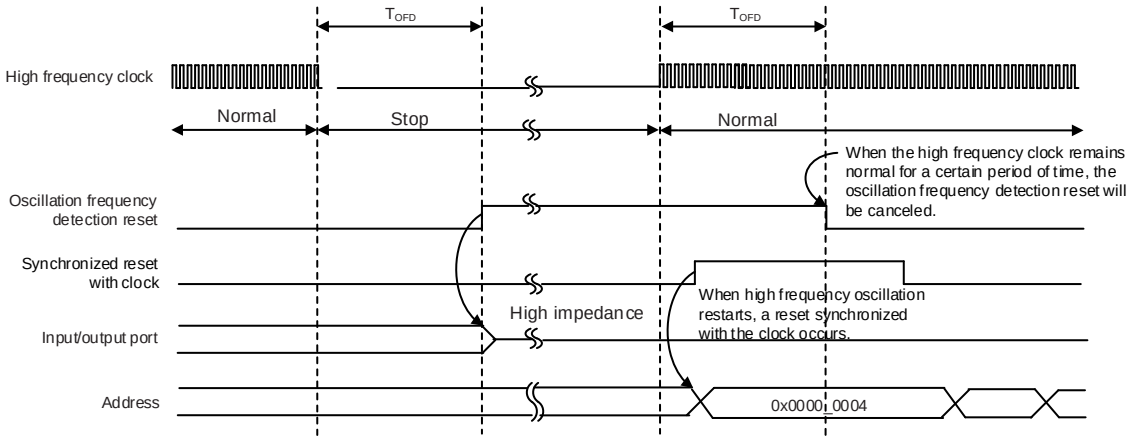


Figure 17.5 Oscillation Frequency Detection Reset Timing

18. Watchdog Timer (WDT)

18.1. Outline

The watchdog timer (WDT) is intended to detect malfunction (runaway) and return it to a normal state when the CPU begins to malfunction (runaway) due to noise, etc.

The WDT generates interrupt request (INTWDT) or resetting when a malfunction (runaway) is detected.

18.2. Block Diagram

The WDT consists of a binary counter and a watchdog timer output control circuit. The WDT is controlled by the watchdog timer mode register and the watchdog timer control register.

Note: TMPM370FYDFG/TMPM370FYFG does not have $\overline{\text{WDTOUT}}$ pin.

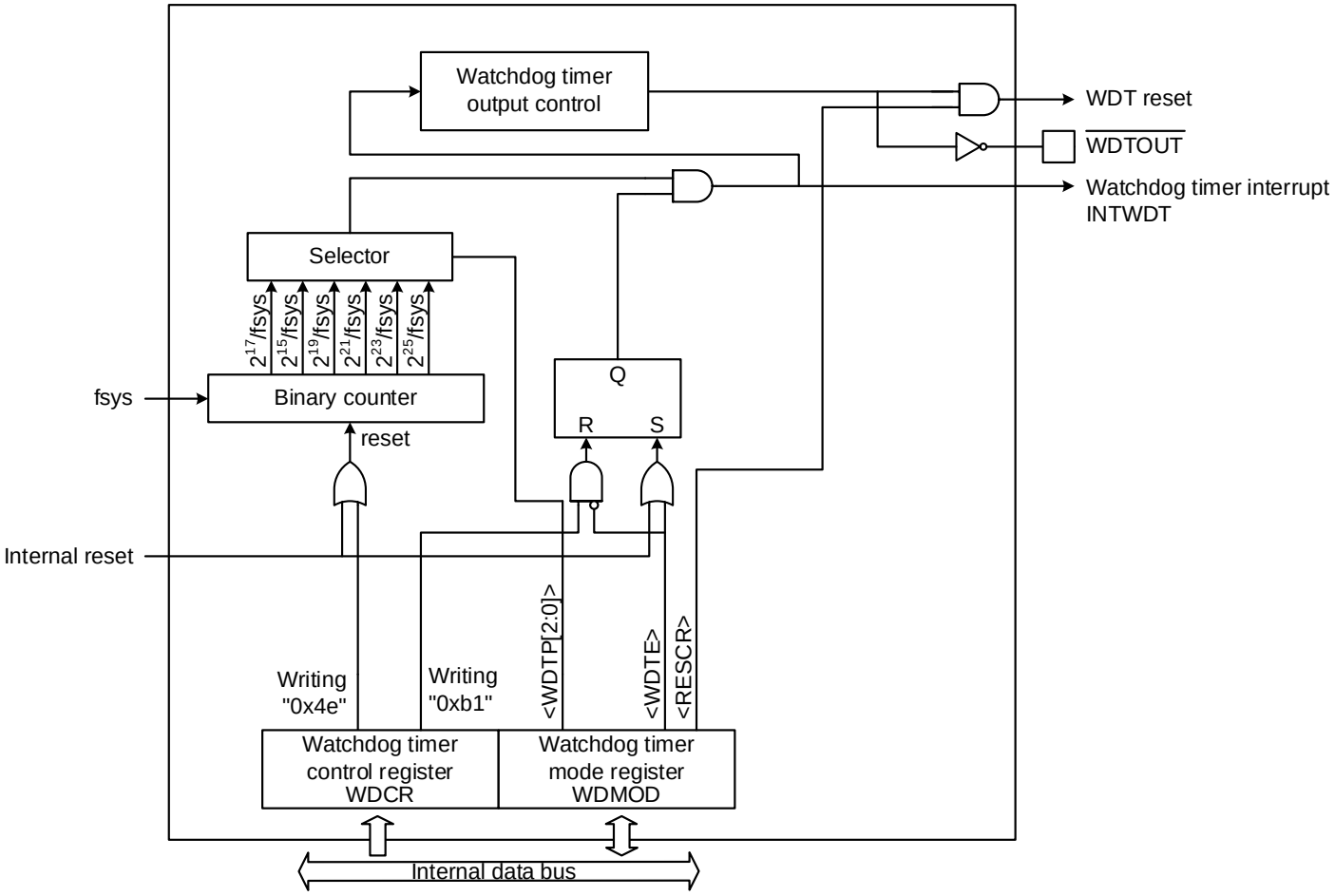


Figure 18.1 Block Diagram of WDT

18.3. Registers

18.3.1. List of Registers

The control registers and addresses are listed below.

Register name		Base+ (Address)
Watchdog timer mode register	WDMOD	0x0000
Watchdog timer control register	WDCR	0x0004

18.3.2. WDMOD (Watchdog Timer Mode Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	WDTE	WDTP			-	I2WDT	RESCR	-
After reset	1	0	0	0	0	0	1	0

Bit	Bit symbol	Type	Function
31:8	-	R	Read as "0".
7	WDTE	R/W	WDT operation control 0: Disabled 1: Enabled
6:4	WDTP[2:0]	R/W	Select the detection time (refer to Table 18.1.)Table 18.1 Watchdog timer detection time (fc = 80MHz) 000: 2^{15} / fsys 100: 2^{23} / fsys 001: 2^{17} / fsys 101: 2^{25} / fsys 010: 2^{19} / fsys 110: Prohibited 011: 2^{21} / fsys 111: Prohibited
3	-	R	Read as "0".
2	I2WDT	R/W	Operation at IDLE mode 0: Stop 1: Operation
1	RESCR	R/W	Operation Selection after Runaway Detection 0: Generates a watchdog timer interrupt (INTWDT). (Note) 1: Generates a watchdog timer reset.
0	-	R/W	Write as "0".

Note: The INTWDT is a non-maskable interrupt (NMI).

Table 18.1 Watchdog timer detection time (fc = 80MHz)

CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.41ms	1.64ms	6.55ms	26.21ms	104.86ms	419.43ms
100 (fc / 2)	0.82ms	3.28ms	13.11ms	52.43ms	209.72ms	838.86ms
101 (fc / 4)	1.64ms	6.55ms	26.21ms	104.86ms	419.43ms	1.68s
110 (fc / 8)	3.28ms	13.11ms	52.43ms	209.72ms	838.86ms	3.36s
111 (fc / 16)	6.55ms	26.21ms	104.86ms	419.43ms	1.68s	6.71s

18.3.3. WDCR (Watchdog Timer Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	WDCR							
After reset	(Note)	(Note)	(Note)	(Note)	(Note)	(Note)	(Note)	(Note)

Bit	bit symbol	Type	Function
31:8	-	R	Read as "0".
7:0	WDCR[7:0]	W	Disable/Clear code 0xb1: Disable code 0x4e: Clear code Others: Prohibited

Note: Only the initial value after reset is a value that is not the disable code (0xb1) or the clear code (0x4e).

18.4. Operations

18.4.1. Basic Operation

The WDT consists of a binary counter which counts with the system clock fsys as the source clock.

The detection time of the WDT is selected from 2^{15}/fsys, 2^{17}/fsys, 2^{19}/fsys, 2^{21}/fsys, 2^{23} / fsys and 2^{25} / fsys by WDMOD<WDTP [2:0]>. Watchdog timer interrupt (INTWDT) or watchdog timer reset occurs after the detection time has elapsed. At this time, "Low" level is output from the watchdog timer out pin ($\overline{\text{WDTOUT}}$ pin).

To detect that the CPU begins to malfunction (runaway) due to noise, etc., the program clears the binary counter before the detection time has elapsed.

The binary counter is not cleared when the CPU malfunctions (runaway) and the program fails to operate normally. At this time, the INTWDT occurs, a watchdog timer reset occurs, or a "Low" level is output from $\overline{\text{WDTOUT}}$ pin to detect the CPU malfunction (runaway).

When the INTWDT occurs, execute the anti-runaway program in the interrupt service routine of INTWDT to operate the CPU normally.

When a watchdog timer reset occurs, the MCU is reset and CPU operation changes to normal.

Also, malfunction (runaway) of the CPU can be detected by using $\overline{\text{WDTOUT}}$ pin as follows.

Allows the external device to detect the "Low" level of $\overline{\text{WDTOUT}}$ pin in advance. $\overline{\text{WDTOUT}}$ pin connects to an external device to detect a CPU malfunction (runaway). $\overline{\text{WDTOUT}}$ pin is set to "High" level by writing the clear code (0x4e) to WDCR.

Note: TMPM370FYDFG/TMPM370FYFG does not have $\overline{\text{WDTOUT}}$ pin.

18.4.2. Operating Mode and State

The WDT immediately starts operation after the reset is released. When not in use, write "0xb1" to WDCR after setting WDMOD<WDTE> to "0" to disable WDT operation.

Note that the watchdog timer cannot be used in the operation mode which fc stops. Disable WDT operation before transitioning to the operation mode shown below.

- STOP Mode

WDT operation during IDLE mode follows the setting of WDMOD<I2WDT>.

In addition, the binary counter stops automatically during the halt mode.

18.5. Operation when Runaway is Detected

18.5.1. INTWDT Generation

Figure 18.2 shows the operation when the binary counter overflows and an INTWDT occurs (WDMOD<RESCR> = "0").

Since INTWDT is NMI, the CPU detects NMI.

CGNMIFLG<NMIFLG0> is set to "1" when an INTWDT occurs.

$\overline{\text{WDTOUT}}$ pin outputs "Low" level at the same time as an INTWDT is generated.

$\overline{\text{WDTOUT}}$ pin is set to "High" level by writing the clear code (0x4e) to WDCR. At this time, the binary counter is cleared and restarts count up.

Note: TMPM370FYDFG/TMPM370FYFG does not have $\overline{\text{WDTOUT}}$ pin.

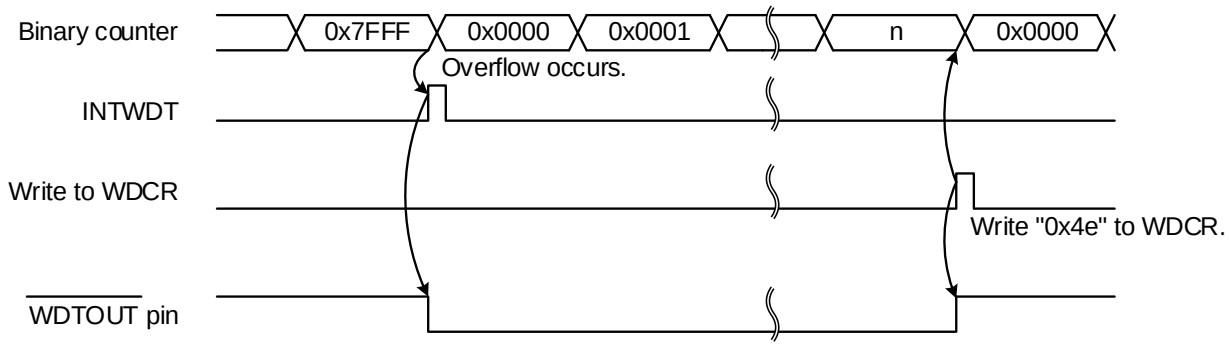


Figure 18.2 Timing of INTWDT Generation

18.5.2. Watchdog Timer Reset Generation

Figure 18.3 shows the operation when the binary counter overflows and a watchdog timer reset occurs (WDMOD<RESCR > = "1").

Watchdog timer reset is outputted for 32 / fsys.

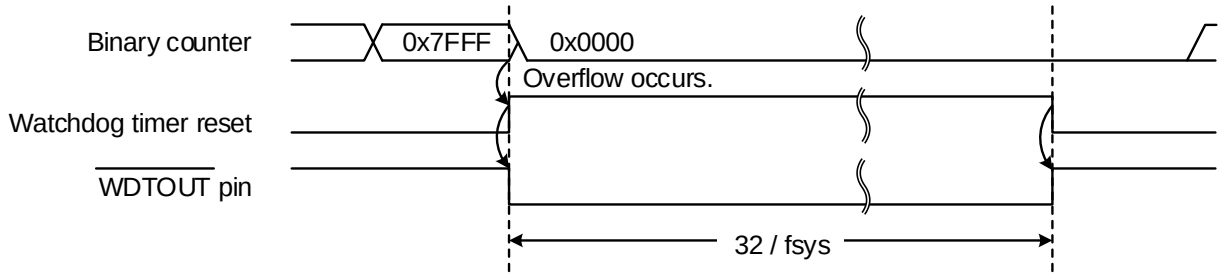


Figure 18.3 Timing of Watchdog Timer Reset Generation

18.6. Control register

The WDT is controlled by two control registers.

18.6.1. Watchdog Timer Mode Register (WDMOD)

18.6.1.1. Selection of Detection Time: WDMOD<WDTP[2:0]>

Selects the detection time. WDMOD<WDTP[2:0]> is initialized to "000" after reset release.

18.6.1.2. WDT Operation Control: WDMOD<WDTE>

WDMOD<WDTE> is initialized to "1" after reset is released. The WDT is enabled.

To prevent erroneous writing due to CPU malfunction (runaway), write the disable code (0xb1) to WDCR after setting WDMOD<WDTE > to "0" to disable WDT.

To enable WDT from disabled, set WDMOD<WDTE > to "1".

18.6.1.3. Operation Selection when Malfunction (runaway) is Detected: WDMOD<RESCR>

Selects the WDT operation with the binary counter overflowed.

WDMOD<RESCR> is initialized to "1" after reset is released. When the binary counter overflows, the WDT generates a watchdog timer reset. To generate INTWDT, set WDMOD<RESCR > to "0".

18.6.2. Watchdog Timer Control Register (WDCR)

Set the disable code (0xb1) to disable WDT.

Set the clear code (0x4e) when clearing the binary counter.

18.6.3. Setting Example

18.6.3.1. Disables the WDT

When the disable code (0xb1) is written to WDCR after setting WDMOD<WDTE> to "0", the watchdog timer is disabled and the binary counter is cleared.

		7	6	5	4	3	2	1	0	
WDMOD	←	0	-	-	-	-	-	-	-	Set WDMOD<WDTE> to "0".
WDCR	←	1	0	1	1	0	0	0	1	Set WDCR to the disable code (0xb1).

Note: *: Any value can be set., -: Cannot be changed.

18.6.3.2. Enables the WDT

Set "1" to WDMOD<WDTE>.

		7	6	5	4	3	2	1	0	
WDMOD	←	1	*	*	*	0	*	*	0	Set WDMOD<WDTE> to "1".

Note: *: Any value can be set., -: Cannot be changed.

18.6.3.3. Clearing the Binary Counter

When the clear code (0x4e) is set to WDCR, the binary counter is cleared and counting continues.

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Set WDCR to the clear code (0x4e).

Note: *: Any value can be set., -: Cannot be changed.

18.6.3.4. Selecting the Detection Time

Setting WDMOD<WDTP[2:0]> to "011" selects 2^{21} / fsys as the detection time.

		7	6	5	4	3	2	1	0	
WDMOD	←	1	0	1	1	0	*	*	0	Set WDMOD<WDTE> "1". The detection time is 2^{21} / fsys.

Note: *: Any value can be set., -: Cannot be changed.

19. Op-amplifiers/Analog Comparators (AMP/CMP)

TMPM370FYDFG/TMPM370FYFG has four channels of op-amplifiers and analog comparators that amplify the port-input voltages and output them to a 12-bit successive-approximation analog-to-digital converter (ADC). This is used to amplify the voltage of the shunt resistor that detects the motor current. The output of the op-amplifier is also input to the analog comparator, and the result is output to the PMD's wave generation circuit by the comparison voltage created by the external resistor. This performs abnormal current detection.

19.1. Block Diagram

Figure 19.1 shows the block diagram of the op-amplifiers and analog comparators.

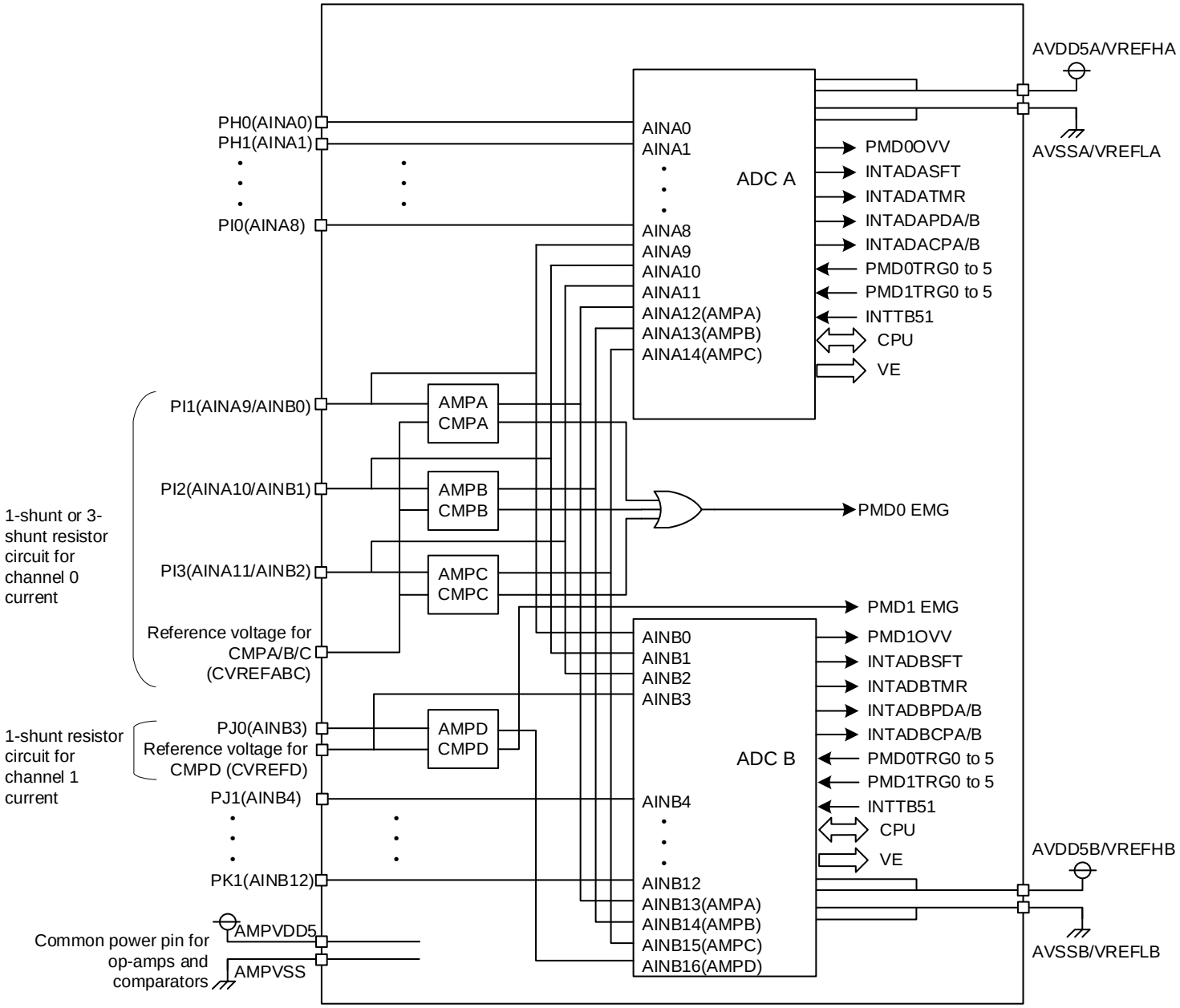


Figure 19.1 Op-amplifiers and Analog Comparators Connection Diagram

19.2. Registers

19.2.1. List of Registers

19.2.2. Op-amplifiers

The op-amplifiers are controlled by AMPCTLA, AMPCTLB, AMPCTLC, AMPCTLD. In addition to enable/disable, eight gain types can be selected.

The op-amplifier control registers and addresses are as follows.

Register name		Address (Base+)
Amplifier A control register	AMPCTLA	0x0000
Amplifier B control register	AMPCTLB	0x0008
Amplifier C control register	AMPCTLC	0x0010
Amplifier D control register	AMPCTLD	0x0018

19.2.2.1. AMPCTLA/AMPCTLB/AMPCTLC/AMPCTLD (Amplifier A to D Control Registers)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	AMPGLIN			AMPEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:4	-	R	Read as "0".
3:1	AMPGLIN[2:0]	R/W	Gain setting 000: x1.5 100: x4.0 001: x2.5 101: x6.0 010: x3.0 110: x8.0 011: x3.5 111: x10.0
0	AMPEN	R/W	AMP enable 0: Disabled 1: Enabled

Note: When AMP is enabled, it takes about 10μs for the circuit to stabilize.

19.2.3. Comparators

The analog comparators are controlled by CMPCTLA, CMPCTLB, CMPCTLC, CMPCTLD. In addition to enabling/disabling, the input port can be selected via op-amplifier or via non-route.

When an op-amplifier is connected externally and the built-in op-amplifier is not used, set to op-amplifier disable (<AMPEN> = "0") and comparator input-amp non-route (<CMPSEL> = "0").

The analog comparator control registers and addresses are as follows.

Register name		Address (Base+)
Comparator A control register	CMPCTLA	0x0000
Comparator B control register	CMPCTLB	0x0008
Comparator C control register	CMPCTLC	0x0010
Comparator D control register	CMPCTLD	0x0018

19.2.3.1. CMPCTLA/CMPCTLB/CMPCTLC/CMPCTLD (Comparator A to D Control Registers)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	-	-	-	-	-	-	CMPSEL	CMPEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31:2	-	R	Read as "0".
1	CMPSEL	R/W	Input selection 0: Via non-route op-amplifier (input port) 1: Via op-amplifier (op-amplifier output)
0	CMPEN	R/W	CMP enable 0: Disabled 1: Enabled

Note: When CMP is enabled, it takes about 10μs for the circuit to stabilize.

19.3. Operations

AMPA, B, C assumes three-shunt current sensing, and the amplified voltages are output to two AD converter units. Two of U, V, W's three-shunt voltages can be converted simultaneously.

The inputs of AMPA, B, C are connected to the direct 2 AD converter unit so that the two shunt voltages can be converted simultaneously, as described above, without using AMPs.

AMPD assumes one-shunt current sensing, and the amplified voltage is output to one AD converter unit.

Figure 19.2 shows the configuration diagram of the op-amplifier and analog comparator.

CMPA, B, C, D are connected to AMPA, B, C, D respectively, and the amplified voltages can be used as inputs. Also, when op-amplifier is not used, the input voltage of the AMP can be directly input to the CMP.

CMPA, B, C assumes three-shunt current sensing and the reference voltage (CVREFABC) is common. CMPD assumes one-shunt current sensing, and the reference voltage (CVREFD) is independent.

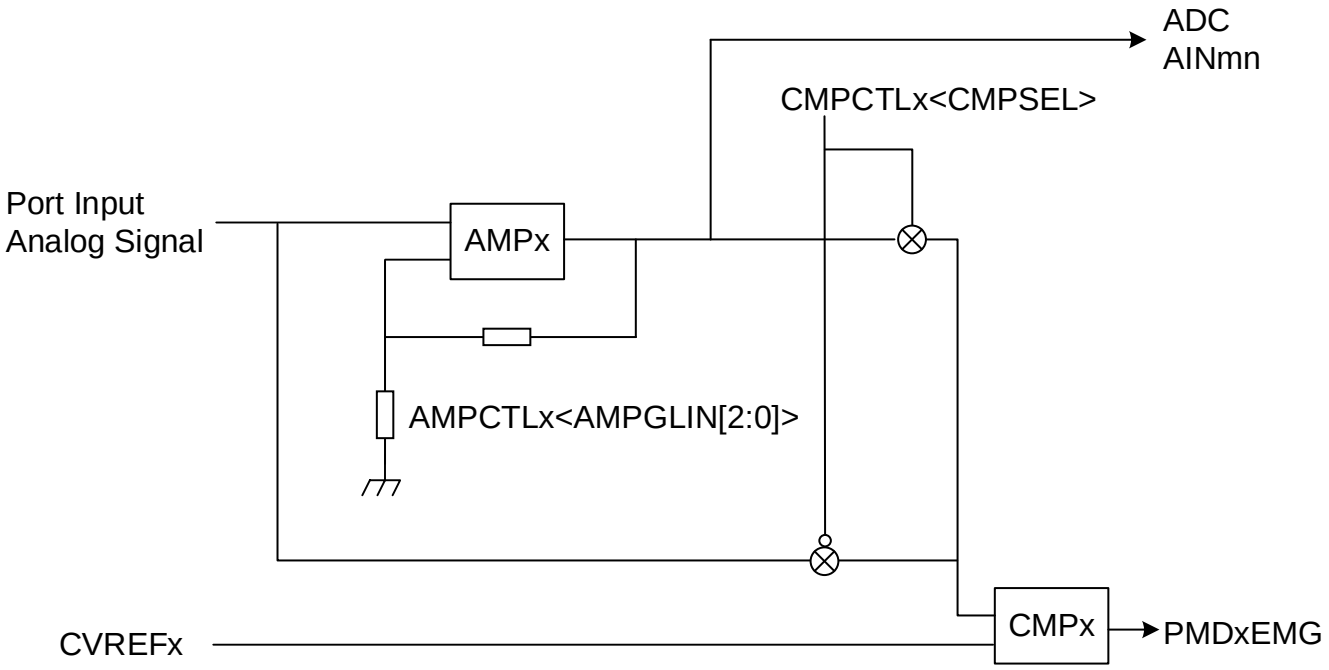


Figure 19.2 Op-amplifier and Analog Comparator Configuration

20. Flash Operations

This section describes the hardware configuration and operation of Flash function.

20.1. Flash Memory

20.1.1. Features

(1) Memory capacity

TMPM370FYDFG/TMPM370FYFG contain a flash memory. The memory capacity and configuration. are shown in the table below.

Each block can be written individually. When the internal flash memory is accessed from the CPU, the data bus width is 32 bits.

(2) Write/erase time

Writing is performed on a page-by-page basis. 1 page is 64 words.

The write time per page is 1.25ms (typ.) regardless of the number of words.

The erase time is 0.1s (typ.) per block.

Write time and erase time per chip are as follows.

Table 20.1 Write/erase time

Product	Memory capacity	Block configuration				Number of words	Write time	Erase time
		128KB	64KB	32KB	16KB			
TMPM370FYDFG TMPM370FYFG	256KB	0	3	1	2	64	1.28s	0.4s

Note: The above values represent theoretical time and do not include data transfer time, etc. The time per chip depends on the user's rewrite method.

(3) Programming methods

The onboard programming mode that can be rewritten on the user's board has the following two modes.

(a) User Boot mode

Support for user's own rewriting method

(b) Single Boot mode

Support for rewriting method in serial transfer (original)

(4) Rewrite method

The internal flash memory of TMPM370FYDFG/TMPM370FYFG are compliant with JEDEC standards except for some functions. This makes it easier to migrate to the MCU even when flash memory is used as external memory. In addition, it has an internal circuit that automatically writes and erases chips in the flash memory, eliminating the need for the user to create a complex flow related to the write and erase operations themselves in a program.

Table 20.2 Rewrite Method

JEDEC compliant functions	Functions that have been modified, added, or deleted
・Automatic programming	<Modified>
・Automatic chip erase	Block-by-block write/erase protection (only software protection is supported)
・Automatic block erase	<Deleted>
・Data polling/toggle bit	Erase resume/suspend function

(5) Protect/security function

TMPM370FYDFG/TMPM370FYFG add the security function that prohibits the reading of flash data by the writer. On the other hand, write/erase protection, which sets rewrite inhibition, cannot support a method (hardware) in which a 12V voltage is applied only by the command (software) to set it. For details on protection and security functions, refer to the "21. Protect/Security Function".

Note: When the password is erased data ("0xFF"), the password can be easily verified, making security difficult. It is recommended that unique values be placed even when Single Boot mode is not used.

20.1.2. Flash Block Diagram

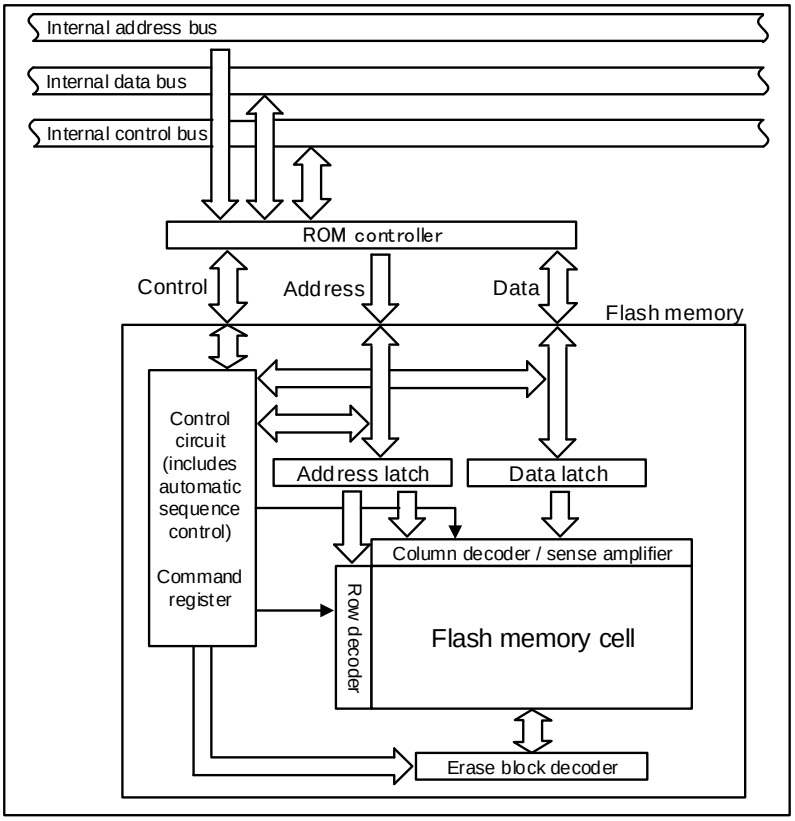


Figure 20.1 Flash Block Diagram

20.2. Operation Mode

There are three operation modes of TMPM370FYDFG/TMPM370FYFG, including when the internal flash memory is not used.

Table 20.3 Operation Modes

Operation mode		Contents of operation
Single chip mode		After the reset is released, it starts up from the internal flash memory.
	Normal mode	This operation mode is defined by dividing the mode for executing the user's application program and the mode for executing the flash memory rewriting on the user's set. The former is called "Normal mode" and the latter is called "User Boot mode".
	User Boot mode	Switching between the two can be set by the user yourself. For example, it can be freely designed as Normal mode when port A0 is "1", and User Boot mode when port A0 is "0". The user should prepare a routine to determine the switch to a part of the application program.
Single Boot mode		After the reset is released, CPU starts from the internal Boot ROM (Mask ROM). Boot ROM is programmed with algorithms that can be rewritten on the user's set through TMPM370FYDFG/TMPM370FYFG's serial port. The internal flash memory can be rewritten by connecting to an external host through the serial port and transferring data according to the specified protocol.

There are two flash memory operating modes that can be programmed in Table 20.3, User Boot mode and Single Boot mode. The modes in which the internal flash memory can be rewritten on the user's set are "User Boot mode" and "Single Boot mode", which define the two as on-board programming mode.

Each operation modes in Single chip and Single Boot are determined by setting the level of the $\overline{\text{BOOT}}$ (PF0) pin externally in the reset state.

Table 20.4 Operation Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PF0)
Single chip mode	"0" → "1"	"1"
Single Boot mode	"0" → "1"	"0"

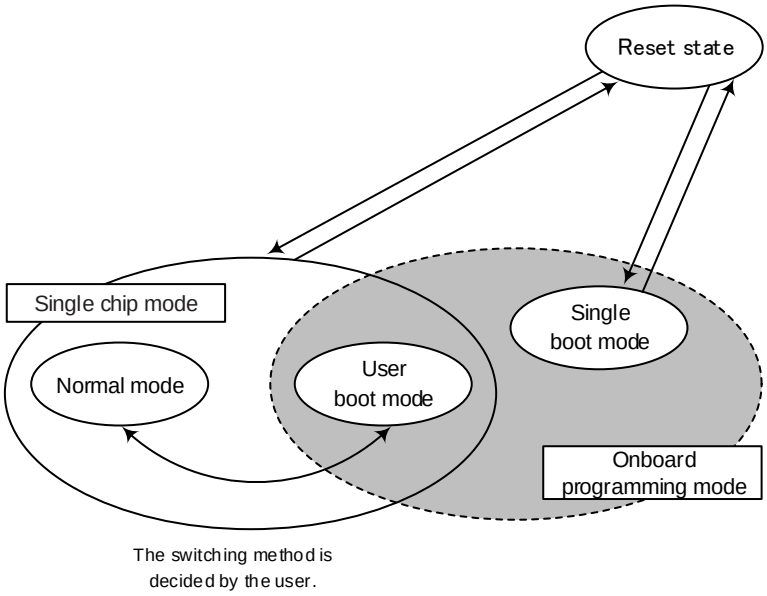


Figure 20.2 Mode Transition Diagram

20.2.1. Reset Operation

To reset TMPM370FYDFG/TMPM370FYFG, set the $\overline{\text{RESET}}$ input pin to "0" for at least duration of 12 system clocks (0.15μs at 80MHz operation (clock gear 1/1 mode after reset)) while the power supply voltage is within the operating voltage range and the oscillation of the internal oscillator is stable.

Note1: After the power is turned on, wait for at least 700μs after the power supply voltage has stabilized before releasing the reset state.

Note2: The reset period of 0.5μs or more is required to perform a hardware reset during the automatic program/erase operation of the internal flash memory regardless of the system clock. In this case, it takes about 2ms before reading becomes possible after the reset is released.

20.2.2. User Boot Mode (Single Chip Mode)

User Boot mode is a method of using user's own flash memory programming routine. This is used when the data transfer bus used in the flash memory rewrite program provided in the user application is different from the serial I/O. Operation is performed in Single chip mode. For this reason, it is necessary to shift from the Normal mode in which the normal user application program is running in Single chip mode to the User Boot mode for rewriting the flash. Therefore, incorporate the program for determining the condition into the reset processing program in the user application.

Use I/O of TMPM370FYDFG/TMPM370FYFG to set the conditions for this mode switching independently according to the user's system set conditions. In addition, the user's own flash memory programming routine to be used after entering the User Boot mode must also be incorporated in the user application beforehand, and rewritten using these routines after entering the User Boot mode. The internal flash memory cannot read data from the flash memory during the erase/write operation mode. For this reason, the programming routine must be stored and executed outside the flash memory area. In addition, it is recommended that write/erase protection be applied to the necessary blocks after the rewrite processing is completed to prevent the content of the flash from being accidentally rewritten during Single chip mode (normal operation mode). In addition, disable all interrupts, including non-maskable interrupts, during User Boot mode.

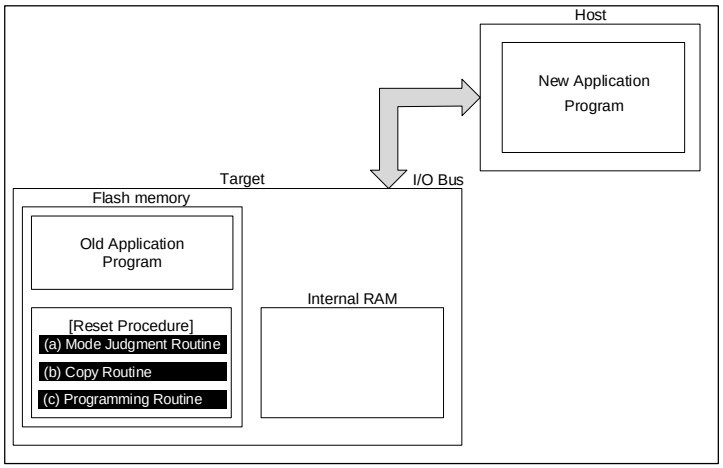
Following (1-A) and (1-B) describe the procedure in two cases, one for placing the programming routine in the internal flash memory and the other for transferring from an external device. For details on how to write/erase the flash memory, refer to "20.3. Flash Memory Write/Erase in Onboard Programming".

20.2.2.1. (1-A) Example of Procedure to Build in Programming Routine into Flash Memory

(1) Step-1

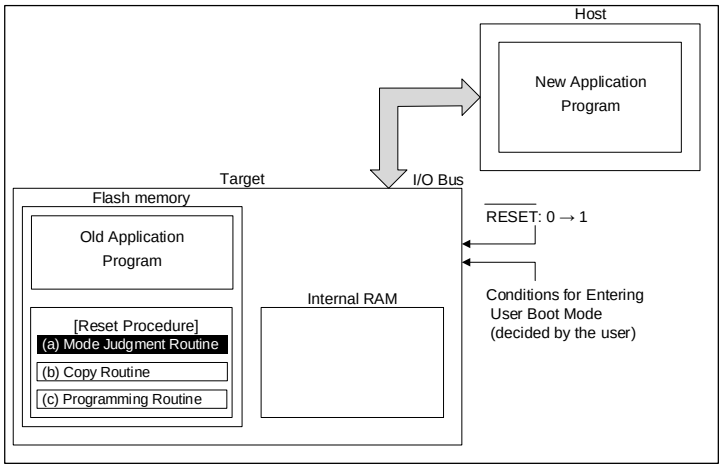
The user determines what conditions (e.g., pin state) are set in advance to shift the User Boot mode. Which I/O bus used for data transfer is determined, and designs circuits and programs that match them. Before building in TMPM370FYDFG/TMPM370FYFG into the board, the user writes the following three programs to any block in the flash memory using a writer, etc.

(a) Mode judgment routine : Program for shifting to the rewrite operation
(b) Copy routine : Program for copying the following (c) to the internal RAM or external memory
(c) Programming routine : program for transferring rewrite data from the outside and rewriting the flash memory



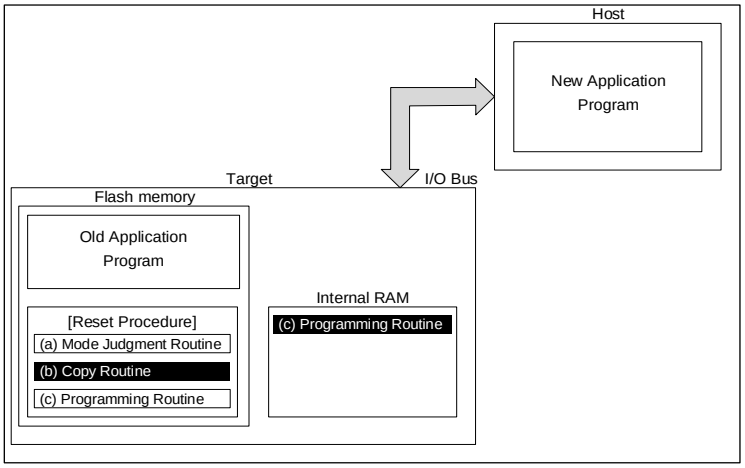
(2) Step-2

The following describes the case where these routines are incorporated in the reset processing program. First, the reset processing program after reset release determines the shifting to User Boot mode. At this time, when the transition conditions are satisfied, the program enters the User Boot mode for rewriting. (When shifting User Boot mode, do not use interrupts afterwards.



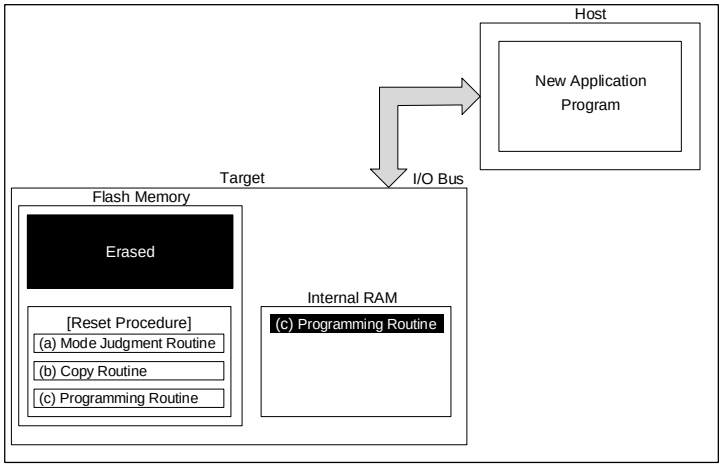
(3) Step-3

When the User Boot mode is shifted, the (b) copy routine is used to copy the (c) programming routine to the internal RAM.



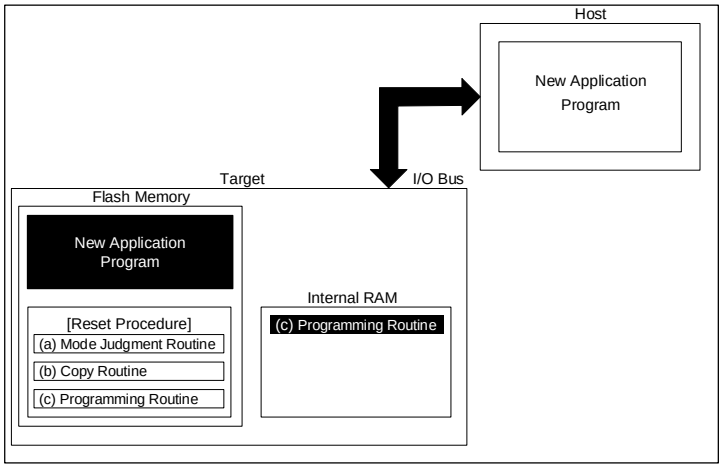
(4) Step-4

Jump to the programming routine on the RAM, releases the write/erase protection of the old user program area, and erases (in blocks).



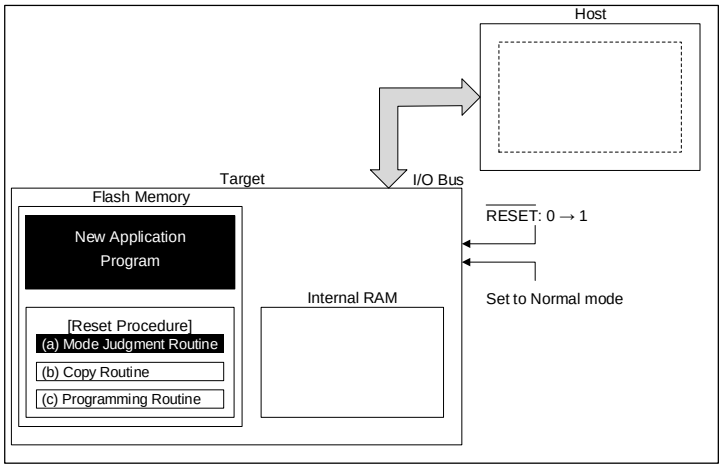
(5) Step-5

In addition, the programming routine on the RAM is executed to load the data of the new user application program from the transfer source (host) and write the data to the erased area of the flash memory. When writing is complete, turn on write/erase protection in the user program area.



(6) Step-6

$\overline{\text{RESET}}$ input pin is set to "0" to reset, and set the setting condition to Normal mode. After the reset is released, the operation starts with the new user application program.



20.2.2.2. (1-B) Example of Procedure for Transferring Programming Routine from Externally

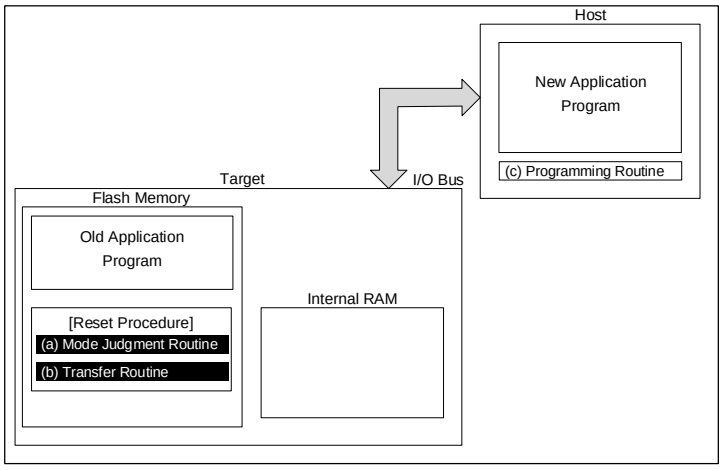
(1) Step-1

The user determines what conditions (e.g., pin state) are set in advance to shift the User Boot mode. Which I/O bus used for data transfer is determined, and designs circuits and programs that match them. Before building in TMPM370FYDFG/TMPM370FYFG into the board, the user writes the following two programs to any block in the flash memory using a writer, etc.

(a) Mode judgment routine : Program for shifting to the rewrite operation
(b) Transfer routine : Program for transferring rewrite programs from the outside

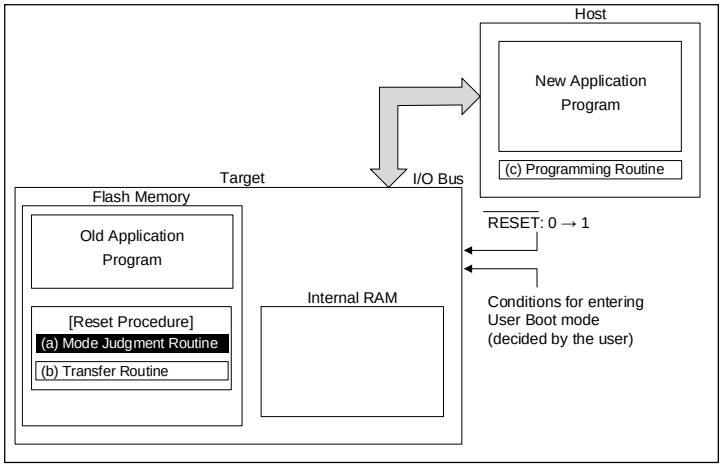
In addition, prepare the following programs on the host.

(c) Programming routine : Program for rewriting the flash memory



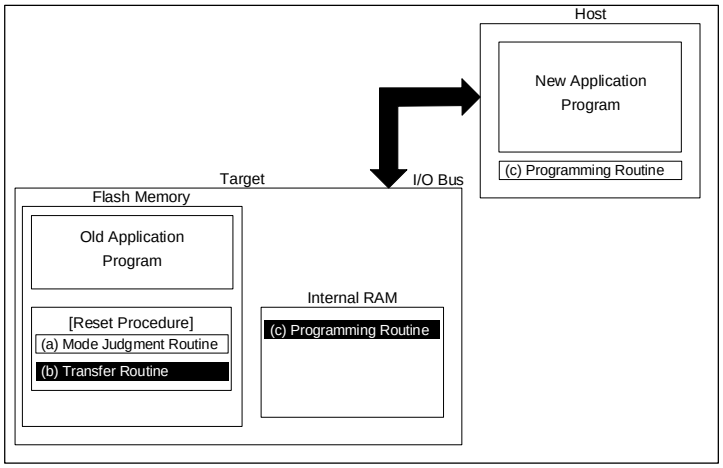
(2) Step-2

The following describes the case where these routines are included in the reset processing program. First, the reset processing program after reset release determines the transition to User Boot mode. At this time, when the transition conditions are satisfied, the program shifts the User Boot mode for rewriting. (When shifting User Boot mode, do not use interrupts thereafter.)



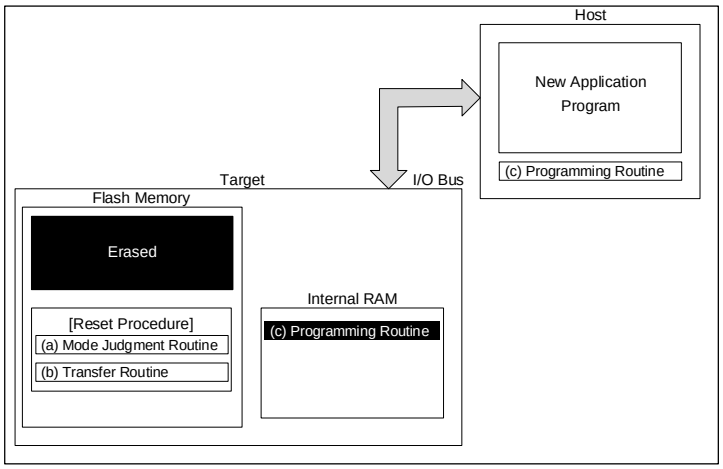
(3) Step-3

When the User Boot mode is shifted, the (b) transfer routine is used to load the (c) programming routine from the forwarding source (host) into the internal RAM.



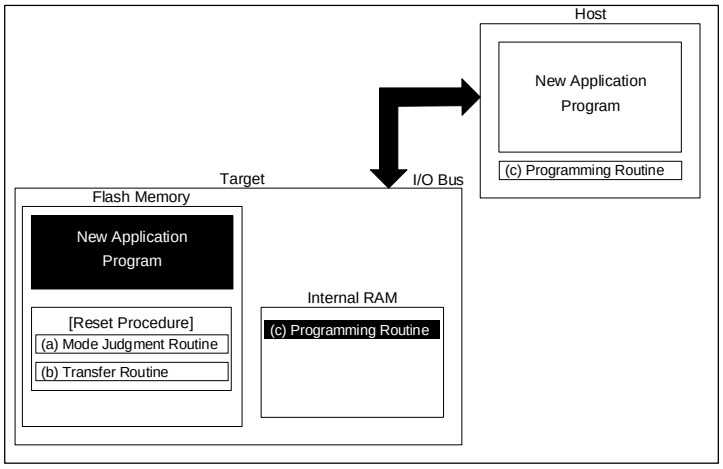
(4) Step-4

Jump to the programming routine on RAM, releases the write/erase protection of the old user program area, and erases (in blocks).



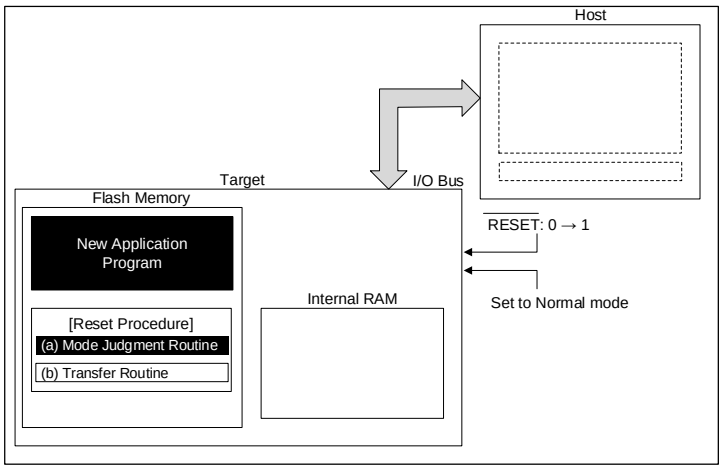
(5) Step-5

In addition, the (c) programming routine on the RAM is executed to load the data of the new user application program from the transfer source (host) and write the data to the erased area. When writing is complete, turn on write/erase protection in the user program area.



(6) Step-6

$\overline{\text{RESET}}$ input pin is set to "0" to reset, and set the setting condition to Normal mode. After the reset is released, the operation starts with the new user application program.



20.2.3. Single Boot Mode

This method starts the internal Boot ROM (mask ROM) and rewrites the flash memory using the Boot ROM program. In this mode, the internal Boot ROM is mapped to the area containing the interrupt vector table and the Boot ROM program is executed. The flash memory is mapped to a different address space than the Boot ROM area.

In Boot mode, commands and data are serially transferred and the flash memory is rewritten. The external host is connected to TMPM370FYDFG/TMPM370FYFG's SIO/UART (channel 0), the external host copies the rewrite program to TMPM370FYDFG/TMPM370FYFG's internal RAM, and the programming routine on the RAM is executed to rewrite the flash memory. The programming routine is executed by sending commands and rewrite data from the host. For details of communication with the host, follow the protocol described later.

The program transfer to RAM verifies the user password before execution in order to ensure the user's ROM data and security. When the passwords do not match, the RAM transfer itself is not executed. Even in Single Boot mode, the RAM transfer is performed in the interrupt disabled state in the same way as in User Boot mode. In Single Boot mode, the Boot ROM program operates in NORMAL mode.

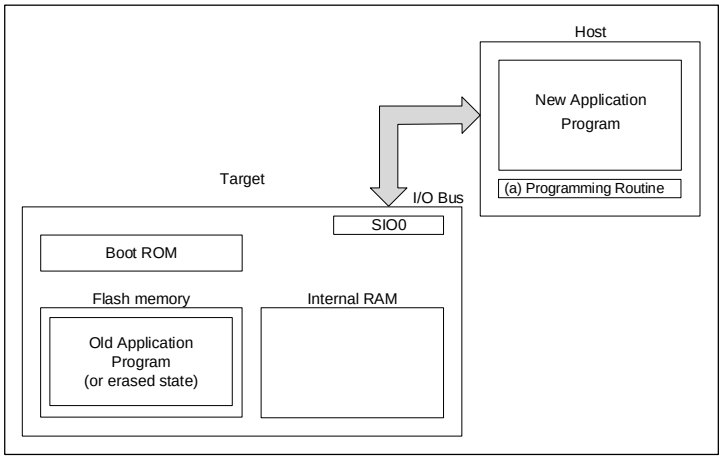
In order to prevent the content of the flash memory from being accidentally rewritten during Single chip mode (normal operation mode), it is recommended that write/erase protection be applied to the necessary blocks when the rewrite processing is completed.

20.2.3.1. (2-A) Using Rewrite Algorithm of Internal Boot ROM

(1) Step-1

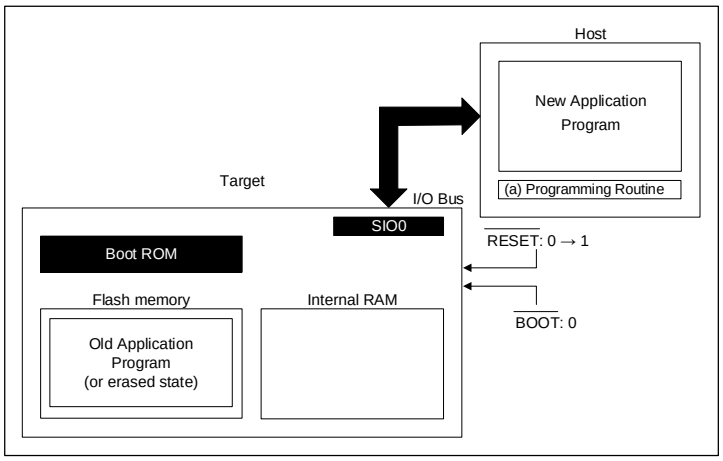
In using rewrite algorithm of internal Boot ROM, it can be supported to the content of flash memory whether the previous version of the user program is written or erased. The transfer of the programming routine, rewrite data, etc. is performed through the SIO/UART (channel 0), so the SIO0 of TMPM370FYDFG/TMPM370FYFG is connected to the external host on the board. Prepare the (a) programming routine on the host for rewriting.

(a) Programming routine : Program for transferring rewritten data from the outside and rewriting the flash memory



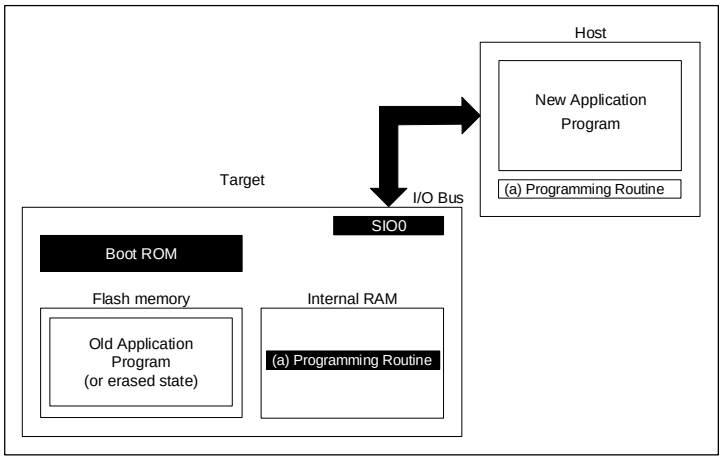
(2) Step-2

Reset is released by the pin condition setting of Boot mode, and it is started by the Boot ROM. Follow the Boot mode procedure to forward the (a) programming routine from the transfer source (host) through SIO0, but first verify the password with the password recorded in the user application program. (Even when the flash memory is erased, the erase data ("0xFF") is verified as a password.)



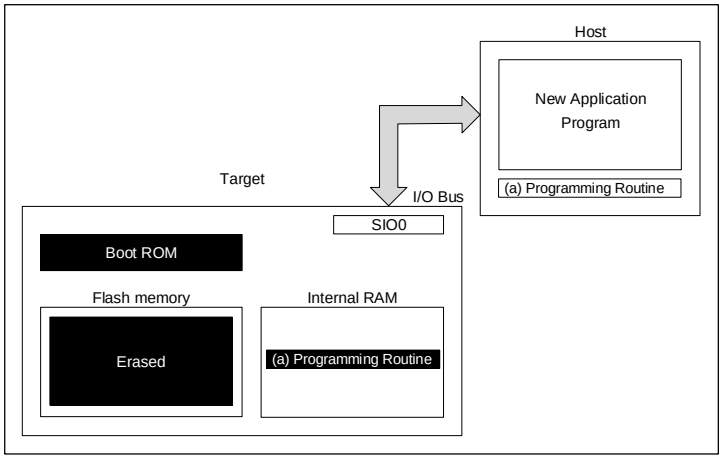
(3) Step-3

When password verification is completed, the (a) programming routine is transferred from the transfer source (host). The Boot ROM loads the routine into internal RAM. However, store the data in the range from the RAM address "0x2000_400" to the last RAM address.



(4) Step-4

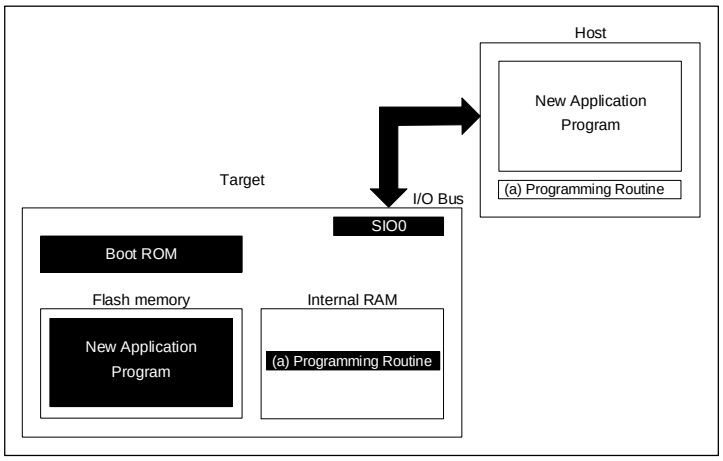
Jump to the (a) programming routine on the RAM and erase the old user application program area. (block unit or all blocks)



(5) Step-5

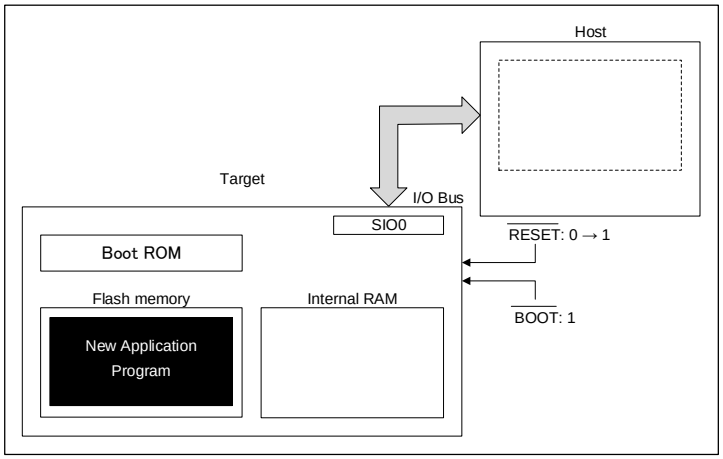
In addition, the (a) programming routine on the RAM is executed to load the data of the new user application program from the transfer source (host) and write the data to the erased area of the flash memory. When writing is complete, turn on write/erase protection in the user program area.

In the example below, the rewritten data is also transferred through the same host and SIO0 as when the programming routine is transferred. However, after the operation starts in RAM, the data bus and transfer source can be set uniquely. Assemble the circuit and programming routines according to the method.



(6) Step-6

When writing is complete, power off the board once and disconnect the cable that was connected to the host. Then, turn the power off and on again, start the Single chip mode (Normal mode), and run the new user application program.



20.2.4. Mode Setting

To perform onboard programming, start TMPM370FYDFG/TMPM370FYFG in Single Boot mode. The following settings are used to start up in Single Boot mode.

$\overline{\text{BOOT}}$ (PF0) = "0"
$\overline{\text{RESET}}$ = "0" → "1"

Set the $\overline{\text{RESET}}$ input pin to "0" and set the $\overline{\text{BOOT}}$ (PF0) pin to the above conditions in advance. After that, it will start in Single Boot mode when reset is released.

20.2.5. Memory Map

Figure 20.3 shows a comparison of memory maps in Single chip mode and Single Boot mode. As shown in the figure, in Single Boot mode, the internal flash memory is mapped from "0x3F80_0000". In addition, the Boot ROM (mask ROM) is mapped from "0x0000_0000" to "0x0000_0FFF".

The internal flash memory and RAM are mapped as follows.

Table 20.5 Mapping of Internal Flash Memory to RAM

Product	Flash sizes	RAM sizes	Flash address (Single chip mode/Single Boot mode)	RAM address
TMPM370FYDFG TMPM370FYFG	256KB	10KB	"0x0000_0000" ~ "0x0003_FFFF"/ "0x3F80_0000" ~ "0x3F83_FFFF"	"0x2000_0000" ~ "0x2000_27FF"

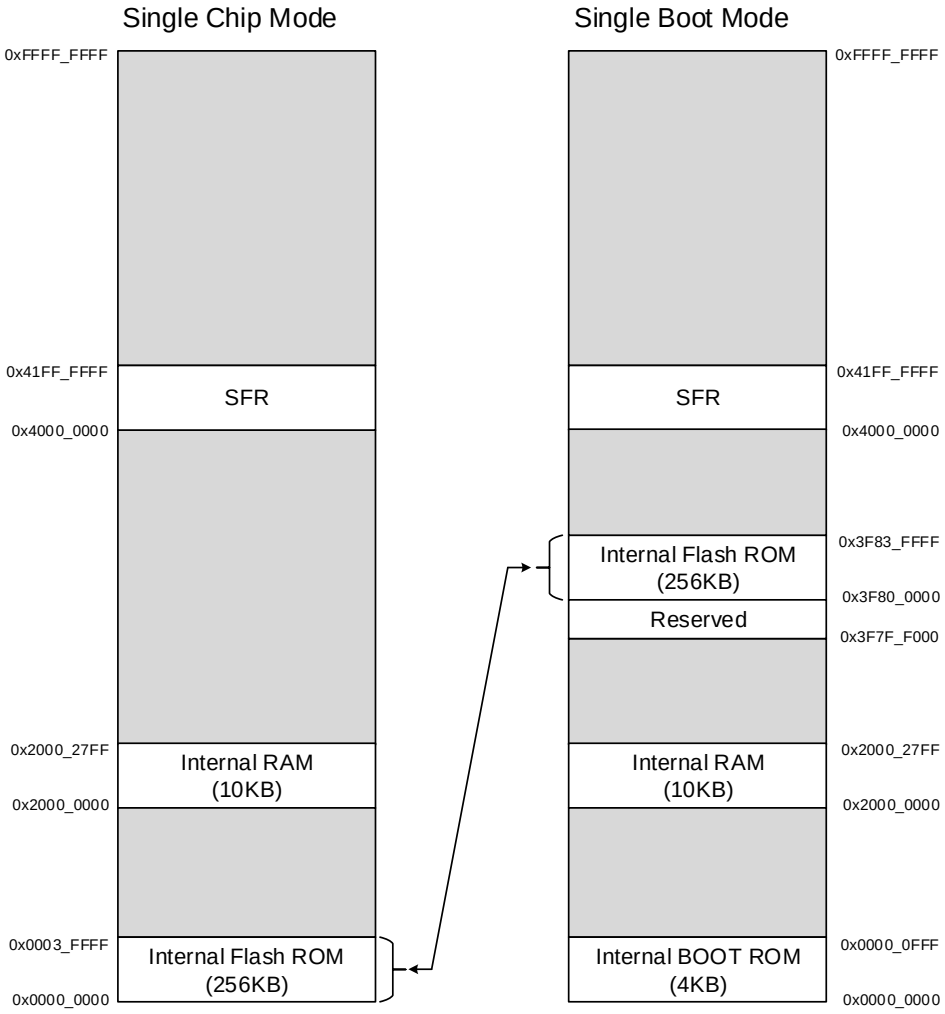


Figure 20.3 Memory Map

20.2.6. Interface Specifications

The communication format in Single Boot mode is shown below. The serial operation mode supports both UART (asynchronous communication mode) and SIO (synchronous communication mode). To perform on-board programming, the communication format of the write controller must be set in the same way.

- When communicating by UART

  Communication channel : SIO/UART0
  Serial Transfer Mode : UART (asynchronous communication mode), half-duplex communication, LSB first
  Data length : 8 bits
  Parity bit : None
  STOP bit : 1 bit
  Baud rate : Any baud rate

- When communicating by SIO

  Communication channel : SIO/UART0
  Serial Transfer Mode : SIO (synchronous communication mode), full-duplex communication, LSB first
  Synchronization (SCLK0) : Input mode
  Handshake pin : Output mode (PE4)
  Baud rate : Any baud rate

Table 20.6 Connection of Pins

Pin		Serial Transfer Mode	
		UART	SIO
Power supply pin	DVDD5	✓	✓
	DVSS	✓	✓
	AVDD5A	✓	✓
	AVSSA	✓	✓
	AVDD5B	✓	✓
	AVSSB	✓	✓
	VOUT3	✓	✓
	VOUT15	✓	✓
	RVDD5	✓	✓
Mode setting pin	$\overline{\text{BOOT}}$ (PF0)	✓	✓
Reset pin	$\overline{\text{RESET}}$	✓	✓
Communication pin	TXD0 (PE0)	✓	✓
	RXD0 (PE1)	✓	✓
	SCLK0 (PE2)	-	✓ (Input mode)
	PE4	-	✓ (Output mode)

Note: ✓: Required, -: Not required

20.2.7. Data Transfer Format

Table 20.7 and Table 20.9 to Table 20.10 show the operation commands and data transfer formats in each operation mode. Read this section in conjunction with the "20.2.10. Operation Description of Boot Program".

Table 20.7 Operation Command Data

Operation command data	Operation command
"0x10"	RAM transfer
"0x40"	Flash memory chip erase and protected bit erase

20.2.8. Constraints of Internal Memories

In Single Boot mode, internal RAM and internal Flash ROM are constrained as shown in Table 20.8.

Table 20.8 Memory Constraints for Single Boot

Memory	Constraints
Internal RAM	The addresses of "0x2000_0000" to "0x2000_03FF" are the work areas of the Boot ROM. Store the program of RAM transfer from "0x2000_400" to the last address of RAM.
Internal Flash ROM	The following addresses are storage areas for ID information and password for software, so as much as possible avoid using as a program area. "0x3F83_FFF0" to "0x3F83_FFFF"

20.2.9. Transfer Format of the Boot Program

The transfer format of the Boot program for each command are explained. Refer to also "20.2.10. Operation Description of Boot Program".

20.2.9.1. RAM Transfer

Table 20.9 Transfer Format of Boot Program [RAM Transfer]

	Number of byte	Data transferred from the controller to this device	Baud rate	Data transferred from this device to the controller
Boot ROM	1st byte	Serial operating mode and baud rate setting For UART: "0x86" For SIO: "0x30"	Specified baud rate (Note1)	-
	2nd byte	-		ACK response to serial operation mode ・For UART In case of normal (settable): "0x86" (Operation stops when it is judged that setting of baud rate is impossible.) ・For SIO In case of normal: "0x30"
	3rd byte	Operation command data ("0x10")		-
	4th byte	-		ACK response to operation command (Note2) In case of normal: "0x10" In case of abnormality: "0xX1" In case of communication error: "0xX8"
	5th byte to 16th byte	Password data (12-bytes) "0x3F83_FFF4" to "0x3F83_FFFF"		-
	17th byte	CHECKSUM value of 5th to 16th byte		-
	18th byte	-		ACK response to CHECKSUM value (Note 2) In case of normal: "0x10" In case of abnormality: "0xX1" In case of communication error: "0xX8"
	19th byte	RAM storage start address 31 to 24		- (Note3)
	20th byte	RAM storage start address 23 to 16		- (Note3)
	21st byte	RAM storage start address 15 to 8		- (Note3)
	22nd byte	RAM storage start address 7 to 0		- (Note3)
	23rd byte	Number of RAM storage byte 15 to 8		- (Note3)
	24th byte	Number of RAM storage byte 7 to 0		- (Note3)
	25th byte	CHECKSUM value of 19th to 24th byte		- (Note3)
	26th byte	-		ACK response to CHECKSUM value (Note2) In case of normal: "0x10" In case of abnormality: "0xX1" In case of communication error: "0xX8"
	27th byte to m th byte	RAM storage data		-
	(m + 1) th byte	CHECKSUM value between 27th and m th byte		-
	(m + 2) th byte	-		ACK response to CHECKSUM value (Note2) In case of normal: "0x10" In case of abnormality: "0xX1" In case of communication error: "0xX8"
RAM	(m + 3) th byte	-		Jump to RAM storage starting address

Note1: In SIO mode, set the baud rate of the 1st byte and the 2nd byte by the specified baud rate divided by 16.

Note2: After an error response, the MCU waits for an operation command (the 3rd byte). In SIO mode, "In case of communication error" does not occur.

Note3: Program the data of the 19th to 25th bytes so that they fall within the area of the last RAM address from the address "0x2000_400" on the RAM.

20.2.9.2. Flash Memory Chip Erase and Protected Bits Erase

Table 20.10 Transfer Format of Boot Program
[Flash Memory Chip Erase and Protect Bit Erase]

	Number of byte	Data transferred from the controller to this device	Baud rate	Data transferred from this device to the controller
Boot ROM	1st byte	Serial Operating Mode & Baud Rate Setting For UART: "0x86" For SIO: "0x30"	Specified baud rate (Note1)	-
	2nd byte	-		ACK response to serial operation mode ・In case of normal (settable) ・For UART: "0x86" ・For SIO: "0x30" (Operation stops when it is judged that setting of baud rate is impossible.)
	3rd byte	Operation command data ("0x40")		-
	4th byte	-		ACK response to operation command (note2) In case of normal: "0x40" In case of abnormality: "0xX1" In case of communication error: "0xX8"
	5th byte	Erase enable command data ("0x54")		-
	6th byte	-		ACK response to operation command (note2) In case of normal: "0x54" In case of abnormality: "0xX1" In case of communication error: "0xX8"
	7th byte	-		ACK response to the erase command In case of normal: "0x4F" In case of abnormality: "0x4C"
	8th byte	(Wait for next operation command data)		-

Note1: In SIO mode, set the baud rate of the 1st byte and the 2nd byte by the specified baud rate divided by 16.

Note2: After an error response, the MCU waits for an operation command (the 3rd byte). In SIO mode, " In case of Communication error" does not occur.

20.2.10. Operation Description of Boot Program

When launched in Single Boot mode, the Boot program starts and provides the following functions:

(1) RAM transfer command

RAM transfer stores data sent from the controller to the internal RAM. When the transfer ends normally, the user program starts executing. Except for the area used by the Boot program ("0x2000_0000" to "0x2000_03FF"), "0x2000_0400" or later can be used as the user program area. The execution start address is the RAM storage start address.

This RAM transfer function allows user-specific onboard programming control. To execute on-board programming in the user program, the flash memory command sequence described in "20.3. Flash Memory Write/Erase in Onboard Programming" must be used. The RAM transfer command checks the password verification results prior to execute. If the password does not match, it will not be executed.

Note: When the password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

(2) Flash memory SUM command

The flash memory SUM command calculates the SUM of the entire flash memory area and returns the result. The boot program does not support the operation command to read the data in the entire area of the flash memory. Instead, there is this flash memory SUM command. The flash memory SUM command can manage the revision of the application program by reading the SUM.

(3) Product information read command

The product information read command returns the product name and memory information of TMPM370FYDFG/TMPM370FYFG. This command returns data in some areas of flash memory. In addition to the flash memory SUM command, this data can be used to manage the revision of application programs. The target areas for each product are as follows.

Table 20.11 Target area

Product	Target area
TMPM370FYDFG TMPM370FYFG	"0x3F83_FFF0" ~ "0x3F83_FFF3"

(4) Flash memory chip erase and protected bit erase

This command erases the flash memory of all blocks. This function erases all blocks in a memory cell and erases the write/erase protection of all blocks, regardless of the write/erase protection or security status. This command does not compare passwords because it also recovers the operation of the Boot program when the password is forgotten.

20.2.10.1. RAM Transfer Command

Refer to Table 20.9 for the data transfer format.

(1) The data of the 1st byte is the data that determines the serial operation mode. For details on how to determine the serial operation mode, refer to "20.2.10.4. Serial Operation Mode Determination". When it is determined to be UART in the serial operation mode, the MCU determines whether the baud rate can be set. The data of the 1st byte is in a state where reception is prohibited (SC0MOD0<RXE> = "0").

- When communicating via UART
  To send the data from the controller to the target board, set the data to "0x86" at the desired baud rate according to UART setting. If it is judged as UART by judging the serial operation mode, it is judged whether the baud rate can be set or not. If it is determined that the setting is impossible, the operation stops, and communication cannot be performed.

- When communicating via SIO
  To send the data from the controller to the target board, set the data to "0x30" at the desired baud rate divided by 16 in the synchronous setting. Similarly, set the 2nd byte to the desired baud rate divided by 16. Transfer at the desired baud rate from the 3rd byte (operation command data).
  In the case of SIO, the CPU sees the reception pin as an input port and monitors the level change of that input port. Therefore, when the baud rate is higher or the operating frequency is lower, the CPU may not be able to determine the level change. To prevent this, for SIO, the baud rate is specified by the desired baud rate divided by 16. When it is judged as SIO, it will be in SCLK input mode. The controller should transmit at a baud rate that satisfies the AC timing. For SIO, the reception error flag is not checked. Therefore, there is no communication error ACK (bit3) ("0xX8") of the ACK response data.

(2) The transmission data of the 2nd byte becomes the ACK response data to the serial operation mode setting data of the 1st byte. When the data of the first byte is judged to be UART and the baud rate can be set, "0x86" is transmitted and when it is judged to be SIO, "0x30" is transmitted.

- When judged as UART
  Determines whether the baud rate can be set. When it is determined that the setting is possible, the value of SC0BRCR is changed, and "0x86" is transmitted. When it is determined that the setting is impossible, nothing is transmitted because the operation is stopped. The controller sets a timeout period (5 seconds) after the transmission of the 1st byte of data is completed. When data ("0x86") cannot be received normally within the time-out time, determine that communication is disabled. The timing for enabling reception (SC0MOD0<RXE> = "1") is performed prior to writing data ("0x86") to the transmit buffer.

- When judged as SIO
  Rewrite SC0MOD0, SC0CR so that SIO/UART operates in SIO mode, write "0x30" to SC0BUF, and wait for SCLK0 clocks. After the first byte of data has been transmitted, the controller should output SCLK0 clock after the idle time (several ms). The baud rate at this time is set at the desired baud rate divide by 16. When the reception data is "0x30", determine that communication is possible. Use the desired baud rate from the 3rd byte. The timing for enabling reception (SC0MOD0<RXE> = "1") is performed prior to writing data ("0x30") to the transmit buffer.

(3) The 3rd byte of receive data becomes operation command data. In this case, it will be the RAM transfer command data ("0x10").

(4) The transmission data of the 4th byte becomes the ACK response data to the operation command data of the 3rd byte. First, check whether there is a reception error in the reception data of the 3rd byte. When there is a reception error, the ACK response data (bit3) "0xX8" of the communication error is transmitted, and the next operation command (the 3rd byte) data is waited. The higher 4 bits of transmit data are undefined. (This bit is the higher 4 bits of the previous operation command data.) In the case of SIO, the reception error is not checked.
Next, if the reception data in the 3rd byte corresponds to any of the operation command data in Table 20.7, the reception data is echoed back (normal ACK response data). In this case, "0x10" is echoed back and branched to the RAM transfer processing routine. After branching to this routine, the data in the password area is checked. For details about how to check the password area data, refer to "20.2.10.5. About Password". Otherwise, the ACK response data (bit0) of the operation command error "0xX1" is transmitted, and the next operation command (the 3rd byte) data is waited. The higher 4 bits of transmit data are undefined. (This is the higher 4 bits of the previous operation command data.)

(5) The reception data of the 5th to 16th bytes is password data (12 bytes). Verify the addresses in the following table in the flash memory in order starting from the 5th byte of reception data. Set the password error flag if it does not match.

Table 20.12 Password Area

Product	Password area
TMPM370FYDFG TMPM370FYFG	"0x3F83_FFF4" ~ "0x3F83_FFFF"

(6) The receive data of the 17th byte becomes CHECKSUM data. Transmit the 2's complement of the lower 8-bit value obtained by adding unsigned 8-bit (ignoring overflow) to the transmit data of 5th to 16th byte from the controller. For details on how to calculate CHECKSUM, refer to "20.2.10.6. How to Calculate CHECKSUM" described later.

(7) The transmission data of the 18th byte becomes the ACK response data (ACK response to CHECKSUM value) for the data of the 5th to 17th bytes. First, check whether there is a reception error in the reception data of the 5th to 17th byte. When there is a reception error, the ACK response data (bit3) "0x18" of the communication error is transmitted, and the next operation command (the 3rd byte) data is waited. The higher 4 bits of transmit data are set to "1" because they are the higher 4 bits of the previous operation command data. In the case of SIO, the reception error is not checked.
Next, CHECKSUM data of the 17th byte is checked. CHECKSUM data checking method checks whether the lower 8 bits of the value obtained by adding unsigned 8 bits of reception data from the 5th to 16th byte (overflow is ignored) are "0x00". When it is not "0x00", the ACK response data (bit0) "0x11" of CHECKSUM error is sent to wait for the next operation command (3rd byte) data.
Finally, check the results of password matching. In the following cases, the ACK response data (bit 0) "0x11" of the password error is transmitted, and the next operation command (3rd byte) data is waited.

- When the 12-byte data in the password area is the same data other than "0xFF" regardless of the verification result of the password data of the 5th to 16th byte
- When the verification of the password data of the 5th to 16th bytes does not match all

After completing the above check, if everything is normal, the normal ACK response data "0x10" is transmitted.

(8) The reception data of 19th to 22nd byte indicates the start address of the RAM to be stored in the block transfer. The 19th byte corresponds to 31 to 24 bit of the address, and the 22nd byte corresponds to 7 to 0 bit. The RAM start address of the storage destination must not be an even address.

(9) The reception data of the 23rd and 24th byte represents the number of byte to be transferred to the block. The 23rd byte corresponds to 15 to 8 bit of the number of transfer bytes, and the 24th byte corresponds to 7 to 0 bit.

(10) The reception data of the 25th byte becomes CHECKSUM data. Transmit the 2's complement value of the lower 8-bit value obtained by adding the unsigned 8-bit data (ignoring overflow) from 19th to 24th byte from the controller. For details on how to calculate CHECKSUM, refer to "20.2.10.6. How to Calculate CHECKSUM" described later.

(11) The transmission data of the 26th byte is the ACK response data (ACK response to CHECKSUM value) for the data of the 19th to 25th byte. First, check whether there is a reception error in the reception data of 19th to 25th byte. When there is a reception error, the ACK response data (bit3) "0x18" of the communication error is transmitted, and the next operation command (the 3rd byte) data is waited. The higher 4 bits of the transmit data are set to "1" because they are the higher 4 bits of the previous operation command data. In the case of SIO, the reception error is not checked.
Next, the CHECKSUM data of 25th byte is checked. CHECKSUM data checking method checks whether the lower 8 bits of the value obtained by adding unsigned 8 bits of reception data from the 19th to 24th byte (overflow is ignored) are "0x00". When it is not "0x00", the ACK-response data (bit0) "0x11" of CHECKSUM error is sent to wait for the next operation command (3rd byte) data.

(12) The reception data of the 27th to m th byte becomes the data to be stored in RAM. Writes the data to be stored in RAM from the address specified by 19th to 22nd, byte and writes the data by the only specified number of bytes from 23rd to 24th byte.

(13) The reception data of (m + 1) th byte becomes CHECKSUM data. Transmit the 2's complement of the lower 8-bit value obtained by adding unsigned 8-bit data (ignoring overflow) to the transmit data in 27th to m th byte from the controller. For details on how to calculate CHECKSUM, refer to "20.2.10.6. How to Calculate CHECKSUM".

(14) The transmission data of (m + 2) th byte is ACK response data (ACK response to CHECKSUM) to the data of 27th to (m + 1) th byte. First, check whether there is a reception error in the reception data of the 27th byte to (m + 1) th byte. When there is a reception error, the ACK response data (bit3) "0x18" of the communication error is transmitted, and the next operation command (3rd byte) data is waited. The higher 4 bits of the transmit data are set to "1" because they are the higher 4 bits of the previous operation command data. In the case of SIO, the reception error is not checked. Next, CHECKSUM data of (m + 1) th byte is checked. CHECKSUM data checking method checks whether the lower 8 bits of the value obtained by adding unsigned 8 bits of reception data from the 27th to the m th byte (overflow is ignored) are "0x00". When it is not "0x00", the ACK response data (bit0) "0x11" of CHECKSUM error is sent to wait for the next operation command (3rd byte) data. If all the above checks are normal, the normal ACK response data "0x10" is transmitted.

(15) When the ACK response data of (m + 2) th byte is normal ACK response data, after sending normal ACK response data "0x10", branch to the address specified in 19th to 22nd byte.

20.2.10.2. Flash Memory Chip Erase and Protect Bit Erase Commands

Refer to Table 20.10 for the data transfer format.

(1) The transmission/reception data from the 1st to the 2nd byte is the same as that of the RAM transfer command.

(2) Controller to Device

The reception data of 3rd byte becomes operation command data. In this case, it becomes the flash memory chip erase command data ("0x40").

(3) Device to controller

The transmission data of the 4th byte becomes the ACK response data to the operation command data of the 3rd byte.

First, check whether there is a reception error in the reception data of the 3rd byte. When there is a reception error, the ACK response data (bit3) "0xX8" of the communication error is transmitted, and the next operation command (the 3rd byte) data is waited. The higher 4 bits of the transmit data become undefined (the higher 4 bits of the previous operation command data).

Next, when the reception data in the third byte corresponds to any of the operation command data in Table 20.7, the reception data is echoed back (normal ACK response data). In this case, "0x40" is echoed back transmitted. Otherwise, the ACK response data (bit0) "0xX1" of the operation command error is transmitted, and the next operation command (the 3rd byte) data is waited. The higher 4 bits of the transmit data become undefined (the higher 4 bits of the previous operation command data).

(4) Controller to Device

The reception data of 5th byte becomes the erase enable command data ("0x54").

(5) Device to controller

The transmission data of the 6th byte becomes the ACK response data to the erase enable command data of the 5th byte.

First, check whether there is a reception error in the reception data of the 5th byte. When there is a reception error, the ACK response data (bit3) "0xX8" of the communication error is transmitted, and the next operation command (3rd byte) data is waited. The higher 4 bits of the transmission data become undefined (the higher 4 bits of the previous operation command data).

Next, if the reception data of the 5th byte corresponds to the erase enable command data, the reception data is echoed back (normal ACK response data). In this case, "0x54" is echoed back and branched to the flash memory chip erase processing routine. Otherwise, the ACK response data (bit0) "0xX1" of the operation command error is transmitted, and the next operation command (3rd byte) data is waited. The higher 4 bits of the transmit data become undefined (the higher 4 bits of the previous operation command data).

(6) Device to controller

This byte indicates whether or not the transmission data of the 7th byte has completed normally.

When the operation complete normally, the end code ("0x4F") is returned.

If an erase error occurs, an error code ("0x4C") is returned.

(7) The reception data of the 8th byte becomes the next operation command data.

20.2.10.3. ACK Response Data

The Boot program transmits the processing status to the controller by various codes. Table 20.13 to Table 20.15 show the ACK response data for each reception data. The higher 4 bits of the ACK response data are the higher 4 bits of the operation command data. The 3rd bit indicates a receive error, and the 0th bit indicates an operation command error, a CHECKSUM error, or a password error. The 1st and 2nd bits are always "0". In the case of SIO, the reception error is not checked.

Table 20.13 ACK Response Data for Serial Operation Determination Data

Transmission data	Meaning of Transmission Data
"0x86"	It was judged that the communication in UART was possible. (Note)
"0x30"	It was judged that communication by SIO was possible.

Note: In the case of UART, when the baud rate setting is determined to be impossible, the operation is stopped without sending anything.

Table 20.14 ACK Response Data for Operation Command Data

Transmission data	Meaning of Transmission Data
"0x?8" (Note)	The reception error occurred in the operation command data.
"0x?1" (Note)	Undefined operation command data was received normally.
"0x10"	It was judged as RAM transfer command.
"0x40"	It was judged as flash memory chip erase command.

Note: The higher 4 bits are the higher 4 bits of the previous operation command data.

Table 20.15 ACK Response Data for CHECKSUM Data

Transmission data	Meaning of Transmission Data
"0xN8" (Note)	The reception error has occurred.
"0xN1" (Note)	The CHECKSUM error has occurred. Alternatively, the password error occurred.
"0xN0" (Note)	CHECKSUM value was judged to be normal.

Note: The higher 4 bits are the higher 4 bits of the operation command data. For example, when a password error occurs, this bit is set to "1" (N = RAM transfer command data[7:4]).

Table 20.16 ACK Response Data for Flash Memory Chip Erase and Protect Bit Erase Operation

Transmission data	Meaning of Transmission Data
"0x54"	It was judged as the erase enable command.
"0x4F"	The erase command was completed.
"0x4C"	The erase command was completed abnormally.

20.2.10.4. Serial Operation Mode Determination

The controller sets the 1st byte to "0x86" at the desired baud rate when communicating with UART, and when communicating with the SIO, set the 1st byte to "0x30" at the desired baud rate divided by 16. Figure 20.4 shows the waveforms in each case.

Note: The points between A, B, C, and D of Figure 20.4 are represented as t_{AB}, t_{AC}, t_{AD}, t_{CD}.

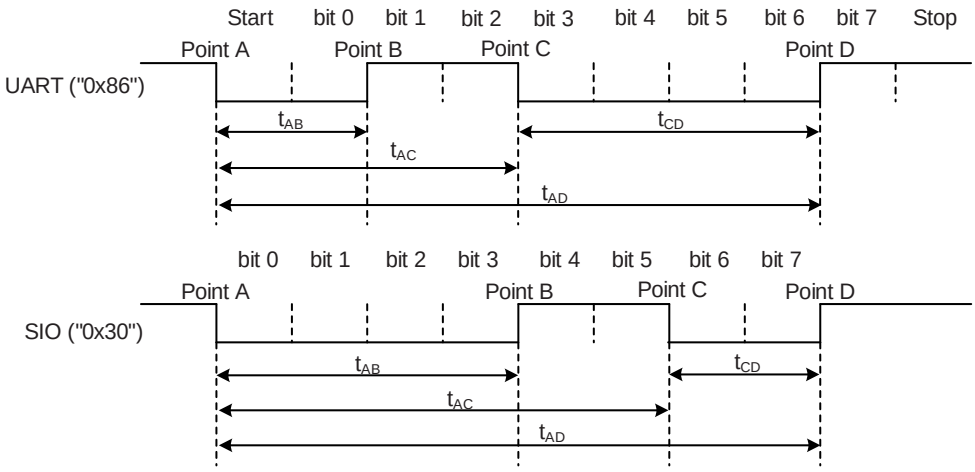


Figure 20.4 Serial Operation Mode Determination Data

The Boot program prohibits reception of the serial operation mode determination data ("0x86" and "0x30") of the 1st byte after reset release, and obtains t_{AB}, t_{AC} and t_{AD} times shown in Figure 20.4 in the flowchart shown in Figure 20.5. As shown in the flowchart in Figure 20.5, when the CPU monitors the level of the reception pin. When there is a change in the level, the timer value at that time is captured. Therefore, there will be an error between the timer values of t_{AB}, t_{AC} and t_{AD}. Also note that when the baud rate is high, the CPU may not be able to determine changes in the level of the reception pin. In particular, SIO has higher baud rates than UART, so this is more likely to occur. To avoid this, for SIO, set the baud rate of the controller to the desired baud rate divided by 16 before transmitting.

As shown in the flowchart in Figure 20.5, the determination of the serial operation mode is determined by the large and small relationship between the width of the time when the reception pin is at the "Low" level. When it is $t_{AB} \leq t_{CD}$, it is judged as UART, and whether the baud rate can be set automatically is judged from t_{AD} time. When it is $t_{AB} > t_{CD}$, it is judged as SIO. Note that, as mentioned above, there is an error in the timer value of t_{AB}, t_{AC}, t_{AD}, so if the baud rate is high and the operating frequency is low, the timer value may decrease and an unintended determination may be made. (Set UART in the rewrite routine again.)Figure 20.5 Reception flowchart

For example, even though the controller wants to communicate with UART, it may be judged as SIO. For this reason, when the controller wants to communicate with UART, after it must transmit the data of 1st byte, and then determine that communication is impossible if it cannot receive data "0x86" normally within the timeout period. To communicate with SIO, after send the data of the 1st byte, output SCLK clock after the idle time, receive the data, and judge that communication is not possible if the reception data is not "0x30".

When the controller want to communicate with SIO, as described above, if it is $t_{AB} > t_{CD}$, the data of the 1st byte may not be "0x30". To be able to determine the falling edge of points A and C, and the rising edge of points B and D, transmit "0x91", "0xA1" or "0xB1" as the 1st byte data. When $t_{AB} > t_{CD}$ is satisfied and SIO is selected in the operation mode determination result, (even if the transmission data of the 1st byte is not "0x30") the data of the 2nd byte will be "0x30", (Below, the data of the 1st byte for SIO judgment is "0x30".)

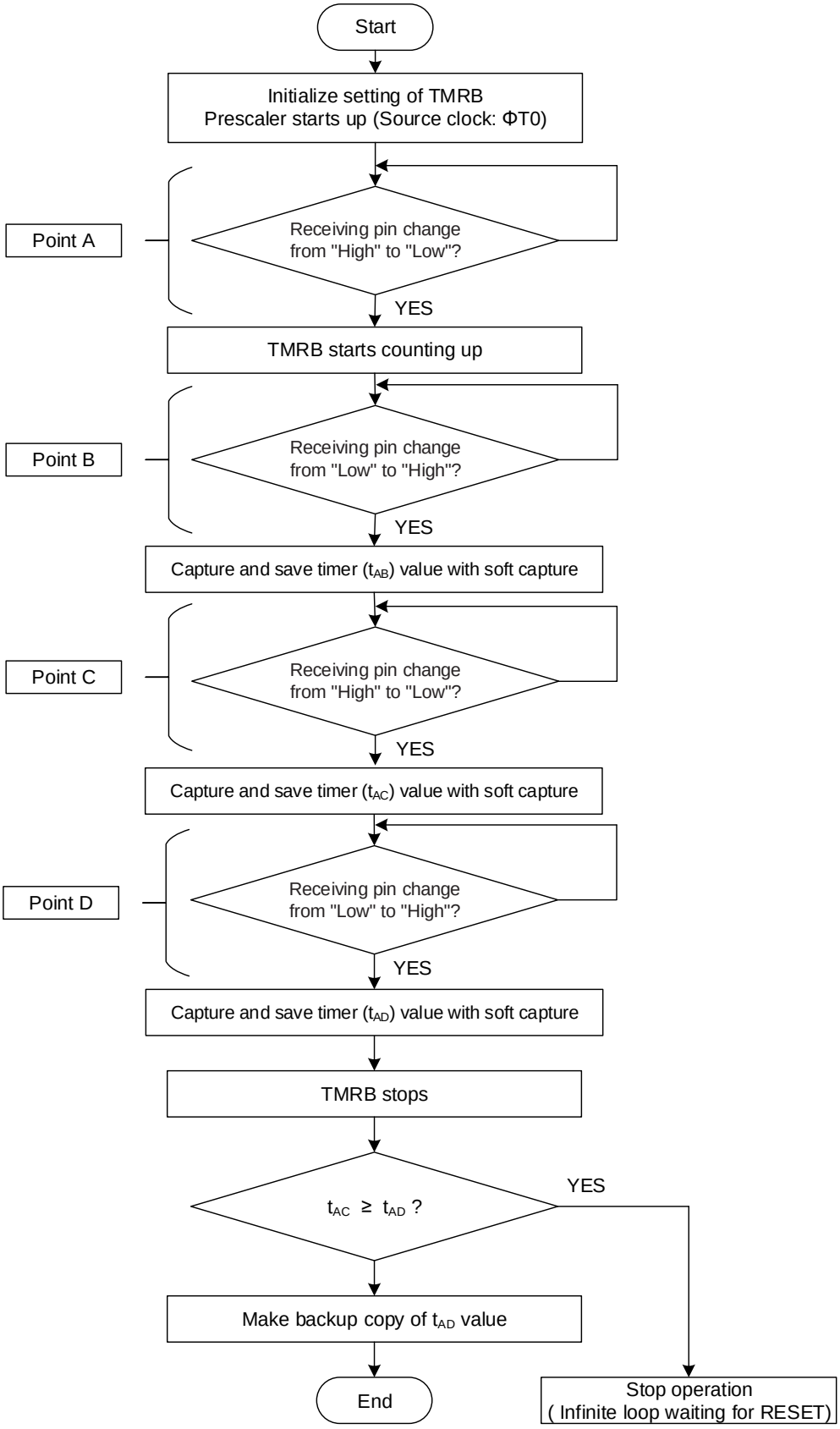


Figure 20.5 Reception flowchart of Serial Operation Mode

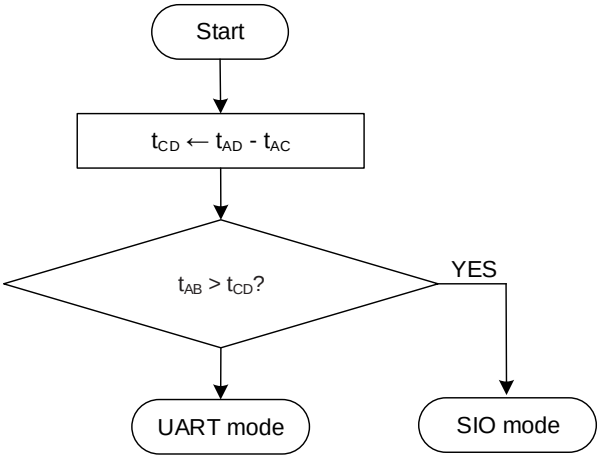


Figure 20.6 Flowchart of Serial Operation Mode Determination

20.2.10.5. About Password

The password confirmation method differs depending on the operation command. The password area is common regardless of the command and is as follows. The password is referenced even when the security function is enabled.

Table 20.17 Password Area

Product	Password area
TMPM370FYDFG TMPM370FYFG	"0x3F83_FFF4" ~ "0x3F83_FFFF"

Note: When the password is erased data ("0xFF"), the password can be easily verified, ensuring security is difficult. It is recommended that unique values be placed even when Single Boot mode is not used.

When the data in the password area is the same data other than "0xFF" as shown in Figure 20.7, it is judged as a password area error. When a password area error is determined, the ACK response to the CHECKSUM value of 17th byte is "0x11" regardless of the verification result of the password data.

Next, the reception data (password data) of the 5th to 16th byte is verified. When all 12 bytes do not match, a password error occurs. When a password error is determined, the ACK response to the CHECKSUM value of 17th byte is the password error.

The password is referenced even when the security function is enabled.

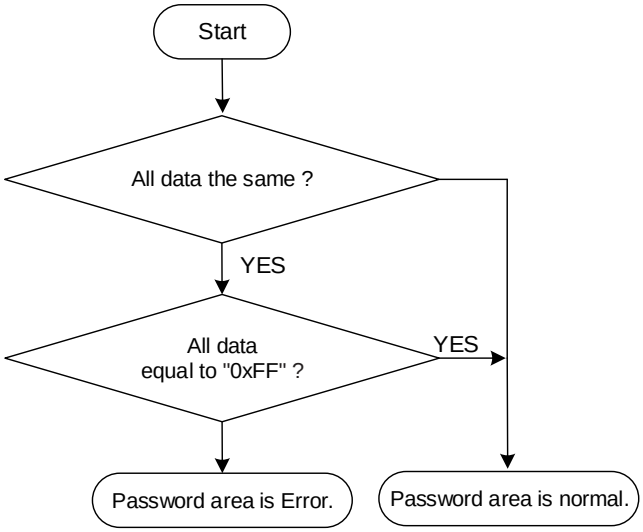


Figure 20.7 Flowchart of Password Area Check

20.2.10.6. How to Calculate CHECKSUM

To calculate CHECKSUM, the 2's complement of the lower 8-bit value obtained by adding unsigned 8-bit of transmission data (ignoring overflow) is calculated. When transmitting CHECKSUM, the controller should use this method.

e.g., CHECKSUM calculation

Obtain CHECKSUM of 2-byte data "0xE5" and "0xF6". First, unsigned 8-bit addition is performed.

"0xE5" + "0xF6" = "0x1DB"

The two's complement of the lower 8 bits of this value is as follows, and this value is CHECKSUM value. Therefore, "0x25" is transmitted to the controller.

"0" - "0xDB" = "0x25"

20.2.11. Flowchart of Entire Boot Program

The flowchart of the entire Boot program is shown below.

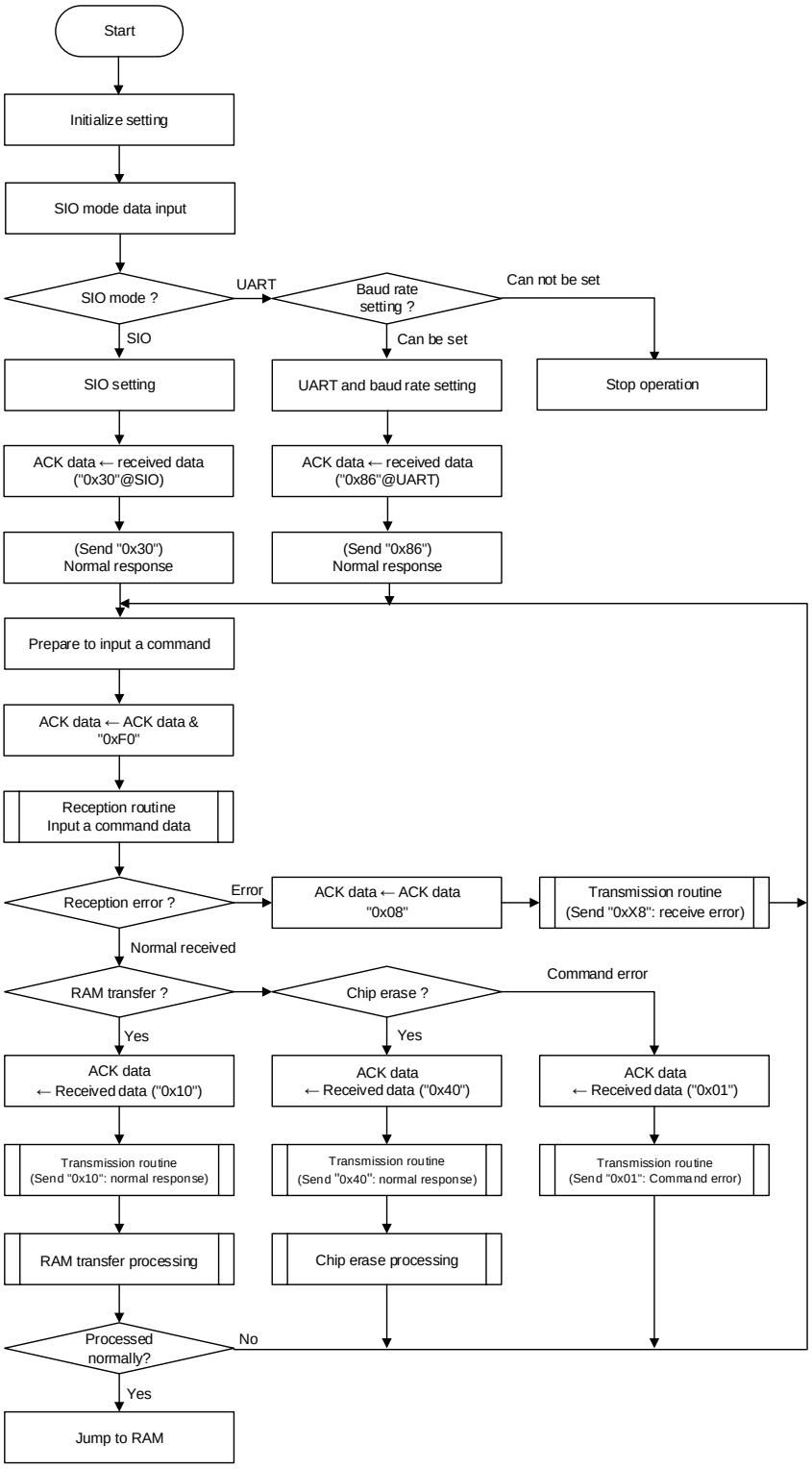


Figure 20.8 Flowchart of Entire Boot program

20.3. Flash Memory Write/Erase in Onboard Programming

In on-board programming, the flash can execute write/erase by executing a command in software by the CPU. This write/erase control program is prepared by the user beforehand. Because the flash memory itself cannot be read while the flash memory is being written or erased, execute the write/erase control program on the internal RAM after shifting the User Boot mode.

20.3.1. Flash Memory

Except for some functions, writing and erasing flash memory conforms to JEDEC standards. Due to the interface with the CPU, the addressing of operation commands differs from that of standard commands.

To write or erase data, enter a command to the flash memory using a 32-bit (word) data transfer instruction. After entering the command, writing and erasing are performed internally automatically.

Table 20.18 Flash Memory Function

Main function	Description
Automatic page program	Data writing is performed automatically.
Automatic chip erase	All areas of the flash memory are erased automatically at once.
Automatic block erase	Erase is performed automatically in block units.
Write/Erase protection	Writing and erasing can be prohibited for each block.

20.3.1.1. Block Configuration

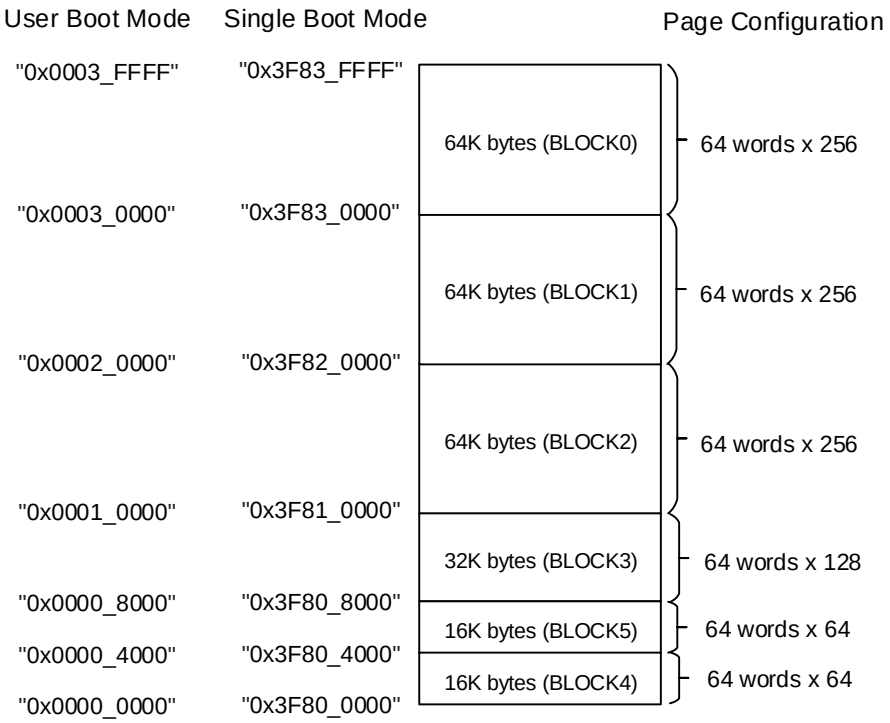


Figure 20.9 Block Configuration

20.3.1.2. Basic Operation

This flash memory can be broadly divided into the following two operation modes:

- Mode to read memory data (read mode)
- Mode to automatically erase or rewrite memory data (automatic operation)

Automatic operation can be started by executing the command sequence in read mode. Flash memory data cannot be read and instructions on the Flash memory cannot be executed during automatic operation. Except for hardware reset, if an exception occurs during automatic operation, the mode does not change to read mode. During automatic operation, do not occur all exceptions except for debug exceptions and reset when connecting to the debug port. Except for a hardware reset, if an exception occurs, the MCU does not shift read mode.

(1) Read

When reading data, set the flash memory to read mode. The flash memory enters read mode immediately after the power is turned on, after the CPU reset is released, or after normal termination of automatic operation. To terminate the automatic operation abnormally or to return to read mode from another mode, use bellow Read/ reset command (software reset) or hardware reset. The read mode must also be used to execute instructions written to the flash memory.

- Read/ reset command and Read command (software reset)
  When ID-Read command is executed, the flash memory does not automatically return to Read mode. Instead, the flash memory is stopped in that state. To return from such a condition to read mode, use Read/ reset command. The Read/Reset command is also used to cancel the command entered halfway. Read command is a command to return to the read mode by executing a 32-bit (word) data transfer command with "0x0000_00F0" data at any address in the flash memory. Read/reset command enters read mode after the end of the 3rd bus write cycle.

(2) Command Write

This flash memory uses a command control method. The command is executed by executing a command sequence to the flash memory. The flash memory executes each automatic operation command according to the input address and data combination (refer to "20.3.1.6. Command Sequence List").

When the command write in the middle of the command sequence is canceled or the wrong command sequence is entered, execute Read/ reset command. The flash memory stops command execution and shifts read mode.

A 32-bit (word) data transfer instruction to the flash memory is called a "bus write cycle". Each command consists of several bus cycles. Each bus write cycle has order, and the flash memory performs automatic operation when the bus write cycle address and data are written to the command in the specified order. When there is a bus write cycle that has not been written to the command in the specified order, the flash memory stops executing the command and enters the read mode.

Note1: Each command sequence is executed from an area outside the flash memory.

Note2: Perform each bus write cycle consecutively using a 32-bit (word) data transfer instruction. Do not access the flash memory while executing each command sequence. In addition, do not generate all interrupts (except for debug exceptions when the debug port is connected).
Unexpected read access to the Flash memory may occur during each bus write cycle and each command sequence, and the command sequencer may not recognize the command correctly. Each command sequence may not terminate normally, and at the same time, it may be recognized as an incorrect command write.

Note3: In order for the command sequencer to recognize the command, the state before the command start must be in read mode. Confirm that FCFLCS <RDY_BSY> = "1" before the 1st bus write cycle of the command sequence. It is recommend to execute Read command.

Note4: In issuing command, when an incorrect address or data is written, be sure to issue a software reset to return to read mode.

20.3.1.3. Reset (Hardware Reset)

Hardware reset is used to forcibly stop execution of automatic program/erase operation or to cancel the operation mode set by command write when automatic operation terminates abnormally.

This flash memory has a reset input as a memory block, and this input is connected to the reset signal of the CPU. Therefore, when the $\overline{\text{RESET}}$ input pin of TMPM370FYDFG/TMPM370FYFG becomes "Low" or the CPU is reset due to an overflow of the watchdog timer, etc., the flash memory stops its operation and returns to read mode even if automatic operation is being executed. Note that when a hardware reset is entered during automatic operation, data cannot be rewritten normally. Rewrite the data again.

For details of the CPU reset operation, refer to "20.2.1. Reset Operation". After a given reset input, the CPU reads the reset vector data from the flash memory and starts the operation after the reset is released.

20.3.1.4. Command Description

(1) Automatic page program

Writing to the flash memory is to change "1" data cell to "0" data cell. "0" data cell cannot be change to "1" data cell. To change the "0" data cell to "1" data cell, an erase operation must be performed.

Automatic page program of TMPM370FYDFG/TMPM370FYFG is writing per page. 1 page is 64 words. In the case of 64 words per page, the address [31:8] is the same, and the group of the leading address [7:0] = "0x00" and the last address [7:0] = "0xFF". The units of page programs are referred to as pages in the following.

Writing to data cell is done automatically by the internal sequencer and does not require external control by the CPU. The status of the automatic page program (whether writing is in progress) can be checked with FCFLCS <RDY_BSY>.

Also, a new command sequence is not accepted during an automatic page program. To cancel the automatic page program operation, use a hardware reset. If this cancels the operation, the automatic page program must be executed again after the erase operation because data is not written correctly to the corresponding page.

The automatic page program can be executed only once for the page after erasing. Even if it is a "1" data cell or a "0" data cell, it cannot be executed more than once for the page. When writing to a page that has been written again, the automatic page program must be executed again after executing the automatic block erase or automatic chip erase command. Executing the page program twice or more on the same page without erasing may damage the device.

Automatic verification operation is not performed inside TMPM370FYDFG/TMPM370FYFG. Check whether the data has been written normally or read after executing the command.

The automatic page program starts from the end of the 3rd bus write cycle of the command cycle. After the 5th bus write cycle, data is written sequentially starting from the next address (in the 4th bus write cycle, the start address of the page is written as a command) specified in the 4th bus write cycle (Data input is performed in 32-bit units.). Be sure to use a 32-bit (word) data transfer instruction to write commands after the 4th bus cycle. At this time, do not execute the 32-bit (word) data transfer instruction to a position crossing the word boundary. After the 5th bus write cycle, the data is performed command write for the same page area. In addition, even if a part of a page is written, automatic page programming must be executed on a page-by-page basis. In this case, the address input of the 4th bus write cycle must also be the start address of the page. At this time, set the input data to "1" and write the command to the position where the data cell is changed to "0". For example, When the first address of a page is not written, the data input of the 4th bus write cycle is performed command write as "0xFFFFFFFF".

When the 3rd bus write cycle is executed, the automatic program is running. This can be verified by monitoring FCFLCS <RDY_BSY>. A new command sequence is not accepted during automatic program operation. To cancel the operation, use a hardware reset. Note that data cannot be written normally when operation is aborted. After writing one page of data to the command, FCFLCS <RDY_BSY> = "1" is set when the automatic page writing is completed normally, and the CPU returns to the read mode.

When writing data to multiple pages, the page program command must be executed for each page (the size that can be written with one automatic page program command is 1 page). Automatic page programming for data input across pages is not possible.

Writing to a write/erase protected block is not possible. When the automatic program finishes normally, it automatically returns to read mode. This is verified by monitoring FCFLCS<RDY_BSY>. When the automatic program operation becomes defective, the flash memory remains locked in this mode and does not return to read mode. To return to the read state, the device must be reset by a hardware reset. In this case, writing to this address is bad, so it is recommend that the device or blocks that contain this address are not used later.

Note: Software reset is disabled in the bus write cycle after the automatic page program 4th bus write cycle.

(2) Automatic chip erase

Automatic chip erase operation starts from the end of the sixth bus write cycle of the command cycle.

The automatic chip erase operation can be checked by monitoring FCFLCS <RDY_BSY>. TMPM370FYDFG/TMPM370FYFG do not perform automatic verification. Therefore, check that the operations are erased normally by reading the data after execution. A new command sequence is not accepted during automatic chip erase operation. To cancel the operation, use a hardware reset. When the operation is aborted, data cannot be erased normally. Therefore, automatic chip erase must be performed again.

Also, when there is a block that is write/erase protected, the block will not be erased. When all the blocks are write/erase protected, automatic chip erase is not executed, and the CPU enters the read mode after completion of the sixth bus write cycle of the command sequence. When the automatic chip erase finishes normally, it automatically returns to the read mode. If the automatic chip erase operation becomes defective, the flash memory remains in this mode and is locked and does not return to read mode.
To return to read mode, the device must be reset by a hardware reset. In this case, the block where the failure occurred cannot be detected. It is recommended to stop the use of the device or use the block erase function to identify the defective block and not use the defective block thereafter.

(3) Automatic block erase (per block)

Automatic block erase starts from the end of the sixth bus write cycle in the command cycle.

The state of the auto block erase operation can be checked by monitoring FCFLCS <RDY_BSY>. TMPM370FYDFG/TMPM370FYFG do not perform automatic verification. Therefore, check that the operations are erased normally by reading the data after execution. A new command sequence is not accepted during automatic block erase. To cancel the operation, use a hardware reset. In this case, data cannot be erased normally, so automatic block erase must be performed again.

Also, when there is a block that is write/erase protected, the block will not be erased. When the automatic block erase operation becomes defective, the flash memory remains locked in this mode and does not return to read mode. Use a hardware reset to reset the device.

(4) Automatic protect bit program (per block)

TMPM370FYDFG/TMPM370FYFG have a built-in protect bit. It can be set in block units. The relation between block and protect bit is shown in Table 20.23. The automatic protection bit program is executed in 1-bit units. The bit is specified by the PBA of the seventh bus write cycle. The automatic-protect bit program disables (protects) write and erase operations for each block. The protection status of block can be checked in FCFLCS<BLPRO >. The state of the automatic-protect bit program operation can be checked by monitoring FCFLCS <RDY_BSY>. A new command sequence is not accepted during the automatic protection bit program operation. To cancel the operation, use a hardware reset. At this time, the protection may not be set correctly, so block protection operation is restarted. After all protect bits have been programmed, all <BLPRO> in FCFLCS are set to "1". After this, all block cannot be written or erased.

Note: In the 7th bus write cycle of the automatic-protect bit program, the software reset is disabled. FCFLCS <RDY_BSY> become "0" after the 7th bus write cycle is inputted.

(5) Automatic protection bit erase

The execution result of the auto protect bit erase command differs depending on the status of the security bit and protect bit. When FCSECBIT<SECBIT >= "1", the operation is determined by whether all <BLPRO> of FCFLCS are "1" or other values. Before executing the automatic protect bit erase command, be sure to check FCFLCS<BLPRO> setting. For details on security, refer to "21. Protect/Security Function".

- IN case of FCFLCS<BLPRO> = all "1" (all protect bits are programmed)

  When the auto protect bit erase command is written, the flash memory is automatically initialized in TMPM370FYDFG/TMPM370FYFG. After the 7th bus write cycle is completed, the data cells in all areas of the flash memory are erased, and the protected bits are subsequently erased. This operation can be checked by monitoring FCFLCS <RDY_BSY>. When the auto protect bit erase operation is finished normally, FCFLCS = "0x00000001". TMPM370FYDFG/TMPM370FYFG do not perform automatic verification. Therefore, check that the operations are erased normally by reading the data after execution. To return to read mode during automatic operation after the 7th bus cycle, the device must be reset by a hardware reset. In this case, after returning to read mode, it is necessary to check the state of the protect bit in FCFLCS<BLPRO> and execute automatic protect bit erase or auto chip erase or auto block erase again as required.

- IN case of FCFLCS<BLPRO > ≠ all "1" (all protect bits are not programmed)

  The protection status can be released by erasing the protection bit. In TMPM370FYDFG/TMPM370FYFG, protect bits are programmed in Block units as shown in Table 20.23, whereas 4 bits are erased together. The protect bit to be erased is specified in the 7th bus write cycle. The status of the protection bits of each block can be checked with FCFLCS<BLPRO>. The state of the automatic protect bit program operation can be checked by monitoring FCFLCS <RDY_BSY>. When the auto protect bit erase operation is completed normally, the erased and selected protect bit in FCFLCS<BLPRO> is set to "0".

In either case, a new command sequence is not accepted during the automatic protection bit erase operation. To cancel the operation, use a hardware reset. When the automatic protect bit erase operation is completed normally, the mode returns to read mode.

Note: FCFLCS <RDY_BSY> is set to "0" during automatic operation and to "1" after the automatic operation is completed.

(6) ID-Read

ID-Read command can be founded out information such as the types of flash memory that are built in TMPM370FYDFG/TMPM370FYFG. The data loaded depends on the value of address [15:14] after the fourth bus write cycle (data input value is "0x00" recommended). When any flash memory area is read after the 5th bus write cycle, the ID value is read. After the 4th bus write cycle of ID-Read command, the CPU does not automatically return to read mode. Reading of the value of the fourth bus write cycle and ID can be executed repeatedly. To return to read mode, use Read/ reset command or hardware reset.

20.3.1.5. Flash Control/Status Register

For the control register and address, refer to "21. Protect/Security Function".

20.3.1.6. Command Sequence List

The address and data for each command are shown in Table 20.19.

The bus cycles except 2nd bus cycle of Read command, the 4th bus cycle of Read/reset command, and the 5th bus cycle of ID-Read command are all "bus write cycles". The bus write cycle is performed by a 32-bit (word) data transfer instruction. (Only the lower 8 bits of data are shown in the table.)

For addresses details, refer to Table 20.20. The following is used for Addr[15:8] marked "Command".

Note: Always set "0" to all bus cycles and address bits [1: 0].

Table 20.19 Flash Memory Access by Internal CPU

Sequence / Command	First bus Cycle	Second bus Cycle	Third bus Cycle	Fourth bus Cycle	Fifth bus Cycle	Sixth bus Cycle	Seventh bus Cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	-	-	-	-	-	-
	0xF0	-	-	-	-	-	-
Read/ Reset	0x54XX	0xAAXX	0x54XX	RA	-	-	-
	0xAA	0x55	0xF0	RD	-	-	-
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	-	-
	0xAA	0x55	0x90	0x00	ID	-	-
Automatic page program	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	-
	0xAA	0x55	0x80	0xAA	0x55	0x10	-
Automatic block erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	-
	0xAA	0x55	0x80	0xAA	0x55	0x30	-
Automatic protection bit program	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address
- PD: Program data (32-bit data)

  After the 4th bus cycle, enter data for one page in order of address
- BA: Block address
- PBA: Protect bit address

20.3.2. Address Bit Configuration during Bus Write Cycle

Table 20.20 is used in conjunction with "Table 20.19 Flash Memory Access by Internal CPU".

Set the address according to "Normal bus write cycle address setting" from the 1st bus cycle. ""0" is recommended" can be changed as appropriate.

Table 20.20 Address Bit Configuration during Bus Write cycle

[Normal Command]

Address	Adr [31:19]	Adr [18]	Adr [17]	Adr [16]	Adr [15]	Adr [14]	Adr [13:11]	Adr [10]	Adr [9]	Adr [8]	Adr [7:0]
Normal command	Bus write cycle address setting for normal command										
	Flash area	"0" recommended			Command						Adr[1:0] = "0" fixed, other bits are "0" recommended.

[ID-Read]

Address	Adr [31:19]	Adr [18]	Adr [17]	Adr [16]	Adr [15]	Adr [14]	Adr [13:11]	Adr [10]	Adr [9]	Adr [8]	Adr [7:0]
ID-Read	IA: ID address (4th bus write cycle address setting of ID-READ)										
	Flash area	"0" recommended			ID address		Adr[1:0] = "0" fixed, other bits are "0" recommended				

[Block erase]

Address	Adr [31:19]	Adr [18]	Adr [17]	Adr [16]	Adr [15]	Adr [14]	Adr [13:11]	Adr [10]	Adr [9]	Adr [8]	Adr [7:0]
Block erase	BA: Block address (6th bus write cycle address setting of block erase)										
	Block address (Table 20.19)						Adr[1:0] = "0" fixed, other bits are "0" recommended				

[Automatic page program]

Address	Adr [31:19]	Adr [18]	Adr [17]	Adr [16]	Adr [15]	Adr [14]	Adr [13:11]	Adr [10]	Adr [9]	Adr [8]	Adr [7:0]
Automatic page program	PA: Program page address (4th bus write cycle address setting of page program)										
	Page address										Adr[1:0] = "0" fixed, other bits are "0" recommended.

[Protect bit program]

Address	Adr [31:19]	Adr [18]	Adr [17]	Adr [16]	Adr [15]	Adr [14]	Adr [13:11]	Adr [10]	Adr [9]	Adr [8]	Adr [7:0]
Protect bit program	PBA: Protect bit address (7th bus write cycle address setting of protect bit program)										
	Flash area	Protect bit selection (Table 20.22)			Fixed to "0"				Protect bit selection (Table 20.22)		Adr[1:0] = "0" fixed, other bits are "0" recommended.

[Protect Bit Erase]

Address	Adr [31:19]	Adr [18]	Adr [17]	Adr [16]	Adr [15]	Adr [14]	Adr [13:11]	Adr [10]	Adr [9]	Adr [8]	Adr [7:0]
Protect bit erase	PBA: Protect bit address (7th bus write cycle address setting of protect bit erase)										
	Flash area	Protect bit selection (Table 20.23)	Fixed to "0"								Adr[1:0] = "0" fixed, other bits are "0" recommended.

The block address can be specified any address contained in the block to be erased.

For the block configuration, refer to "20.3.1.1. Block Configuration".

Table 20.21 Block Address

Block	Address (User Boot mode)	Address (Single Boot mode)	Size (Kbyte)
4	"0x0000_0000" to "0x0000_3FFF"	"0x3F80_0000" to "0x3F80_3FFF"	16
5	"0x0000_4000" to "0x0000_7FFF"	"0x3F80_4000" to "0x3F80_7FFF"	16
3	"0x0000_8000" to "0x0000_FFFF"	"0x3F80_8000" to "0x3F80_FFFF"	32
2	"0x0001_0000" to "0x0001_FFFF"	"0x3F81_0000" to "0x3F81_FFFF"	64
1	"0x0002_0000" to "0x0002_FFFF"	"0x3F82_0000" to "0x3F82_FFFF"	64
0	"0x0003_0000" to "0x0003_FFFF"	"0x3F83_0000" to "0x3F83_FFFF"	64

Note: As for the upper addresses from the 1st bus cycles to the 5th bus cycles, specify the addresses of the blocks to be erased.

Table 20.22 Protect Bit Program Address

Block	Protection bit	Address of the 7th bus write cycle						
		Address [18]	Address [17]	Address [16]	Address [15:11]	Address [10]	Address [9]	Address [8]
Block0	<BLPRO[0]>	0	0	Fixed to "0"			0	0
Block1	<BLPRO[1]>	0	0				0	1
Block2	<BLPRO[2]>	0	0				1	0
Block3	<BLPRO[3]>	0	0				1	1
Block4	<BLPRO[4]>	0	1				0	0
Block5	<BLPRO[5]>	0	1				0	1

Table 20.23 Protect Bit Erase Address

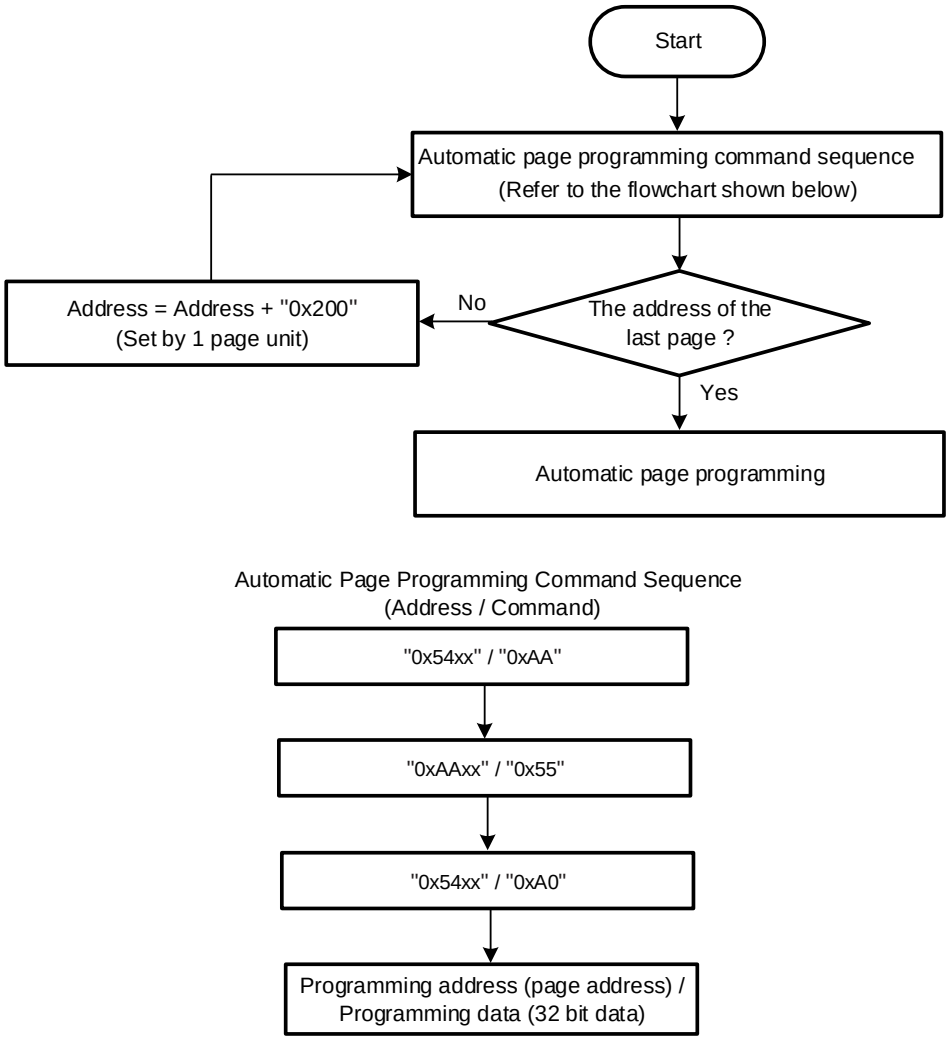
Block	Protection bit	Addresses of the 7th bus write cycle [18:17]	
		Address [18]	Address [17]
Block0 to 3	<BLPRO[3:0]>	0	0
Block4 to 5	<BLPRO[5:4]>	0	1

Note: The protect bit erase command cannot be erased in units of protected bits.

Table 20.24 ID Addresses (IA) of 4th Bus Write Cycle of ID-Read Command and Data (ID) that can be Read by Subsequent 32-bit Transfer Instruction

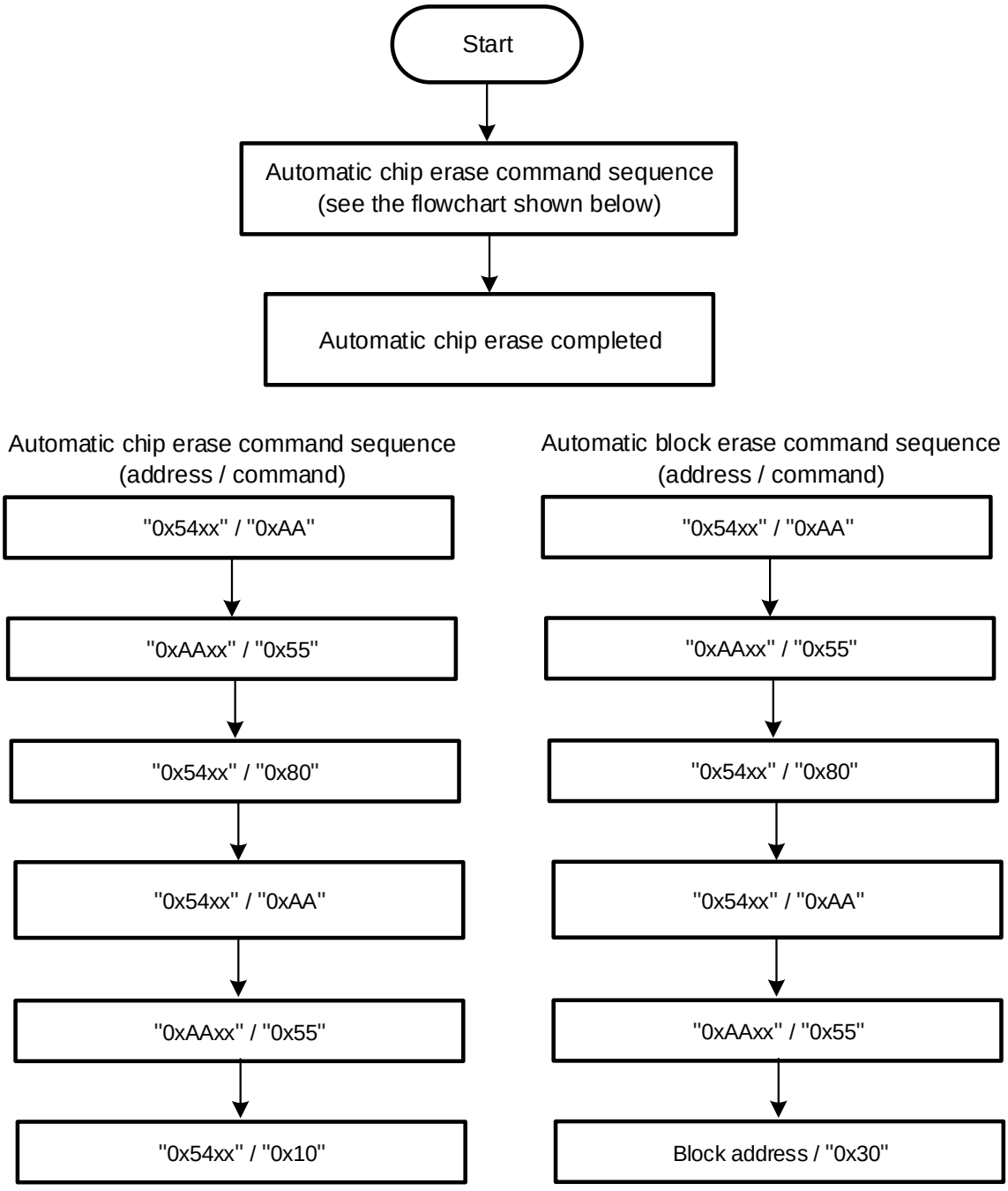
IA[15:14]	ID[7:0]	Code
00	"0x98"	Manufacturer code
01	"0x5A"	Device code
10	Reserved	-
11	"0x11"	Macro code

20.3.2.1. Flowchart



Note: "0x54xx" executes the command sequence even at "0x55xx".

Figure 20.10 Automatic Program



Note: "0x54xx" executes the command sequence even at "0x55xx".

Figure 20.11 Automatic Erase

21. Protect/Security Function

21.1. Outline

TMPM370FYDFG/TMPM370FYFG incorporate the writing/erasing protect function for the internal ROM (flash) and the security function which prevents reading the internal ROM (flash) area by the writer. The security function also limits the use of the debug function. The protect/security function has the following two functions:

- Writing/erasing protect for the internal ROM (flash)
- Security function

21.2. Features

21.2.1. Writing/erasing protect of internal ROM (Flash)

The internal flash can prohibit the operation of writing and erasing on a block-by-block basis. This function is called writing/erasing protect.

To activate the function, write "1" to the corresponding bits to a block to protect. Writing "0" to the bits cancels the protect. (Refer to "20. Flash Operations" for programming details.)

The protect bits can be monitored by FCFLCS<BLPRO[5:0]>.

21.2.2. Security function

The security function restricts readout and debugging of internal flash. This function is called a security function. This function is available under the conditions shown below.

(1) FCSECBIT<SECBIT> is set to"1".

(2) All the protect bits (FCFLCS<BLPRO>) used for the writing/erasing protect function are set to "1".

Note: FCSECBIT<SECBIT> is set to "1" at cold reset.

Table 21.1 shows details of the restrictions by the security function.

Table 21.1 Restrictions by the security function

Item	Details
ROM data readout	Data can be read from CPU.
Debug port	Communication of JTAG/SW and trace are prohibited.
Executing commands for flash	Writing a command to the flash is prohibited. When trying to erase the protect bit for writing/erasing protect, chip erase is executed and all protect bits are also erased.

21.3. Registers

21.3.1. Registers List

The control registers and their addresses are as follows.

Register name		Address (Base+)
Reserved	-	0x0000,0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024 to 0x0FFF

Note: Access to the "Reserved" area is prohibited.

21.3.2. FCSECBIT (Security bit register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-		-	-	-	SECBIT
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31:1	-	R	Read as "0."
0	SECBIT	R/W	Security bit 0: Security function is disabled 1: Security function is enabled

Note: This register is initialized at cold reset.

21.3.3. FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	-	-		-	-	-	RDY_BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31:22	-	R	Read as "0."
21:16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 (Note1) 0: Not protected 1: Protected The protect bit value represents the protect status of the corresponding each of block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15:1	-	R	Read as "0."
0	RDY_BSY	R	Ready/Busy flag (Note2) 0: Under auto operating 1: Auto operation terminated The RDY_BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Note1: This bit can read the value corresponding to the protected state.

Note2: This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from this condition, execute system reset. When performing a hardware reset, a reset period of 0.5 μs or more is required regardless of the system clock. In this case, it takes about 2ms to be able to read after the reset is released.

21.4. Setting/Releasing Method

21.4.1. Writing/erasing protect of internal ROM (Flash)

Writing or erasing the protect bit is executed using the command sequence. Writing to the protect bit is performed on each one block. Erasing the protect bit is performed on the units of Block0 to 3 and Block4 to 5.

Note that when all protect bits of all blocks are set to "1" and FCSECBIT<SECBIT> is "1", the security function is enabled. When the protect bit is erased in this state, all the protect bits will be erased after executing chip erase. Therefore, it is necessary to set FCSECBIT<SECBIT> to "0" first and then erase the protect bit.

Refer to "20. Flash Operations" for details on the command sequence.

21.4.2. Security bit

FCSECBIT<SECBIT> that activates the security function is set to "1" at a Power-on-Reset immediately after power-on. This bit is rewritten by the following procedure.

(1) Write the specific code "0xa74a9d23" to FCSECBIT.

(2) Write data within 16 clocks from writing of the above (1).

Note: The above procedure must be executed by using 32-bit data transfer command.

22. Debug Interface

22.1. Outline

TMPM370FYDFG/TMPM370FYFG contain the Serial Wire JTAG Debug Port (SWJ-DP) unit as debug interface for connecting with the Debug tool and the Embedded Trace Macrocell (ETM) unit for trace output of an internal program. The trace data is output to the pin for debug (TRACEDATA0 to 1, SWV) via the Trace Port Interface Unit (TPIU) in the TMPM370FYDFG/TMPM370FYFG.

For detailed information of SWJ-DP, ETM, and TPIU, refer to "Arm documentation set for the Arm Cortex-M3 processor".

22.1.1. SWJ-DP

TMPM370FYDFG/TMPM370FYFG support the Serial Wire Debug Port (SWDCK, SWDIO) and JTAG Debug Port (TDI, TDO, TMS, TCK, $\overline{\text{TRST}}$).

22.1.2. ETM

TMPM370FYDFG/TMPM370FYFG support two data signal pins (TRACEDATA 0 to 1), one clock signal pin (TRACECLK), and the SWV trace output from one Serial Wire Viewer signal (SWV).

22.2. Pin Functions

Debug interface pin shares with a general-purpose port.

The relation between debug interface pins and general-purpose ports is shown in Table 22.1.

Table 22.1 Relation Between Debug Interface Pins and General-purpose Ports

SWJ-DP/ETM pin name	General-purpose port name	JTAG Debug function		SW Debug function	
		Input/Output	Function	Input/Output	Function
TMS/SWDIO	PB3	Input	JTAG Test Mode Selection	Input/Output	Serial Wire Data Input/Output
TCK/SWCLK	PB4	Input	JTAG Test Check	Input	Serial Wire Clock
TDO/SWV	PB5	Output	JTAG Test Data Output	Output (Note)	Serial Wire Viewer Output
TDI	PB6	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PB7	Input	JTAG Test Reset	-	-
TRACECLK	PB0	Output	TRACE Clock Output		
TRACEDATA0	PB1	Output	TRACE DATA Output 0		
TRACEDATA1	PB2	Output	TRACE DATA Output 1		

Note: When SWV is used

PB3 to PB7 are assigned to debug interface pins after releasing RESET. Other debug interface pins are used as a general-purpose port. When they are used as debug interface pins, Port control registers should be set for using as debug interface. Debug interface pins which are not used can be used as a general-purpose port.

The value of Port ; which is assigned as the debug interface pin; control registers after releasing RESET is shown in Table 22.2。

Table 22.2 Value of Port; Which is Assigned as Debug Interface Pin; Control Registers After Releasing RESET

Assignment after releasing RESET	General-purpose port name	SWJ-DP/ETM pin name	Setting value of port control register					
			Function register (PBFR1)	Input control register (PBIE)	Output control register (PBCR)	Open-drain control register (PBOD)	Pull-up control register (PBPUP)	Pull-down control register (PBPDN)
Debug interface pin	PB3	TMS/SWDIO	1	1	1	0	1	0
Debug interface pin	PB4	TCK/SWCLK	1	1	0	0	0	1
Debug interface pin	PB5	TDO/SWV	1	0	1	0	0	0
Debug interface pin	PB6	TDI	1	1	0	0	1	0
Debug interface pin	PB7	$\overline{\text{TRST}}$	1	1	0	0	1	0
General-purpose port	PB0	TRACECLK	0	0	0	0	0	0
General-purpose port	PB1	TRACEDATA0	0	0	0	0	0	0
General-purpose port	PB2	TRACEDATA1	0	0	0	0	0	0

When using a low power consumption mode, take note of the followings.

- When PB3 and PB5 are used as a debug interface pin, they continuously output even in STOP mode regardless of setting CGSTBYCR<DRVE>.
- When PB4 is assigned to a debug interface pin, a power consumption of TMPM370FYDFG/TMPM370FYFG is not reduced enough. When PB4 is not used as debug interface pin, do not assign PB4 as a debug interface pin.

22.3. Connection With Debug Tool

22.3.1. Connection

Refer to the method which is recommended by each debug tool manufacture for connecting with a debug tool. Debug pins have an internal pull-up register or pull-down register. When an external pull-up register or pull-down register is connected to a debug interface pin, consider them.

22.3.2. Notice When Debug Interface Pins are Used As General-purpose Port

After releasing RESET, when a user program changes a debug interface pin to a general-purpose port, a debug tool cannot control TMPM370FYDFG/TMPM370FYFG, from then on. Therefore, the debugging by a debug tool cannot be done.

The necessary debug interface pin for using debug interface must be not used as a general-purpose port.

Table 22.3 Debug Interface And Using Debug Interface pins

Using debug interface	Using debug interface pins for each debug interface							
	PB3	PB4	PB5	PB6	PB7	PB0	PB1	PB2
	TDO/SWV	TCK/SWCLK	TMS/SWDIO	TDI	$\overline{\text{TRST}}$	TRACECLK	TRACEDATA0	TRACEDATA1
JTAG+SW (Default, After releasing RESET)	✓	✓	✓	✓	✓	-	-	-
JTAG+SW (without $\overline{\text{TRST}}$ pin)	✓	✓	✓	✓	- (Note2)	-	-	-
JTAG+TRACE	✓	✓	✓	✓	✓	✓	✓	✓
SW	-	✓	✓	-	-	-	-	-
SW+SWV	✓	✓	✓	-	-	-	-	-
No using debug function	-	-	-	-	-	-	-	-

Note1: ✓: required, -: not required (debug interface pin can be used as a general-purpose port)

Note2: For the treatment of the pin of which the $\overline{\text{TRST}}$ function is assigned, select the $\overline{\text{TRST}}$ function with the function register and set the pin to OPEN or "High level".

22.3.3. Peripheral Functions Operation During HALT Mode (Stop Execution of Program In Debugging)

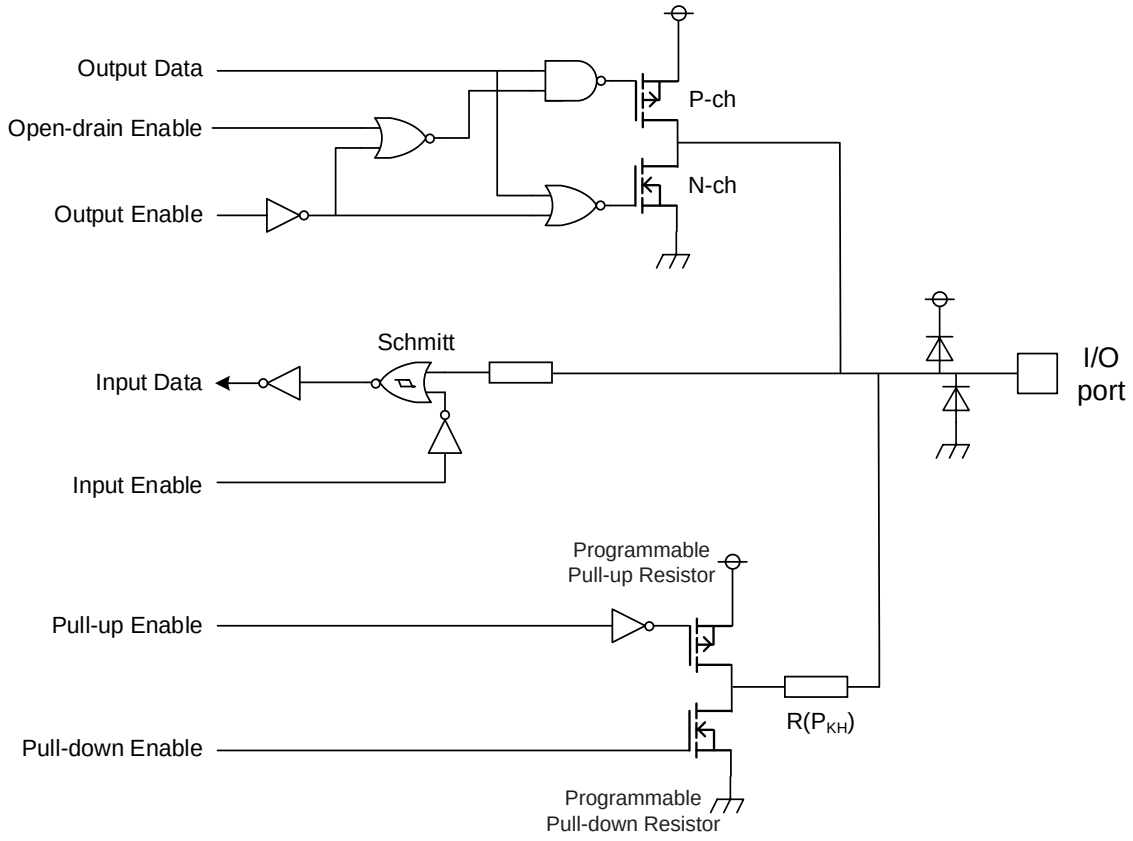
In debugging, the WDT stops counting automatically when the Coretex-M3 core enters in HALT mode. Other peripheral functions operate continuously.

23. Port Section Equivalent Circuit Schematic

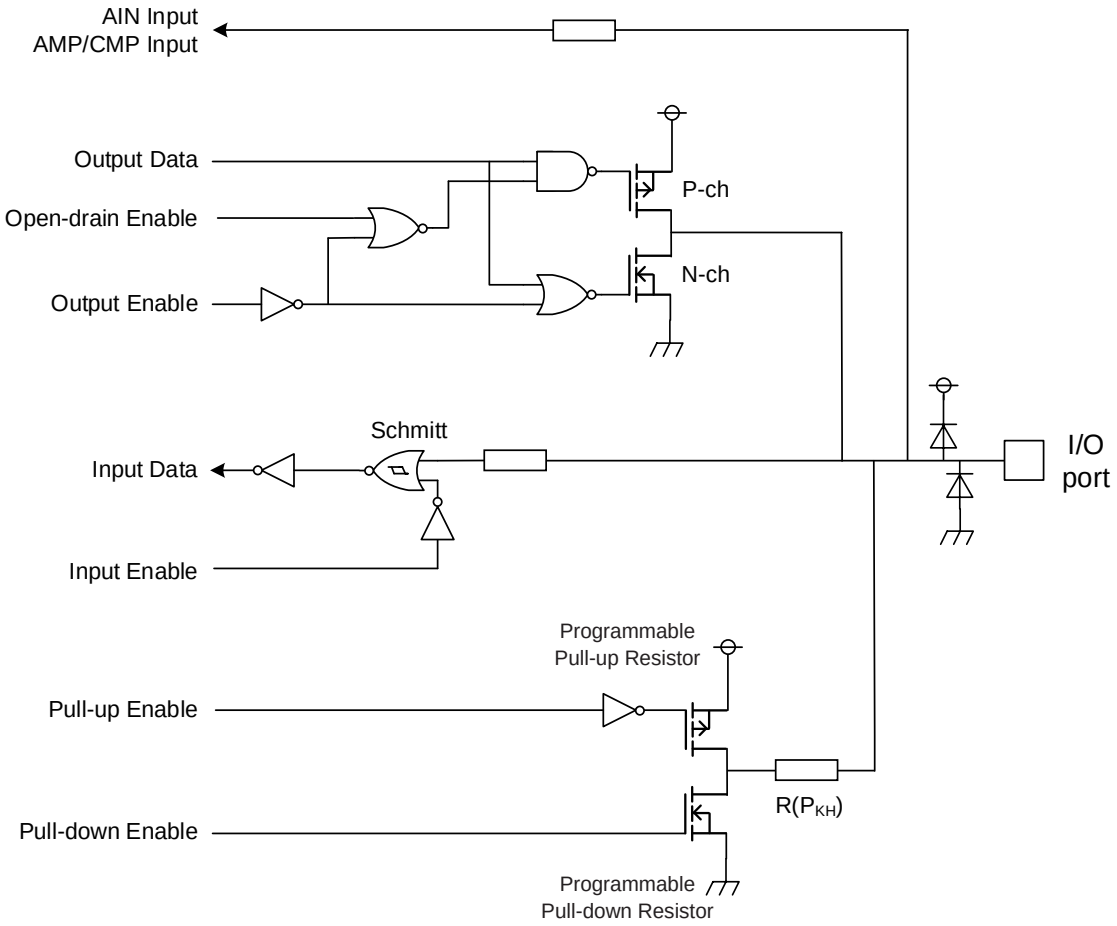
Basically, the gate symbols are the same as those used for the standard CMOS logic IC “74HCxx” series.
The input protection resistor range is from several tens of Ω to several hundreds of Ω.

Note: For the damping resistor value of X2 pin, the typ. Value is shown in the figure. The resistor without the statement of the numerical value in the figure shows input protection resistor.

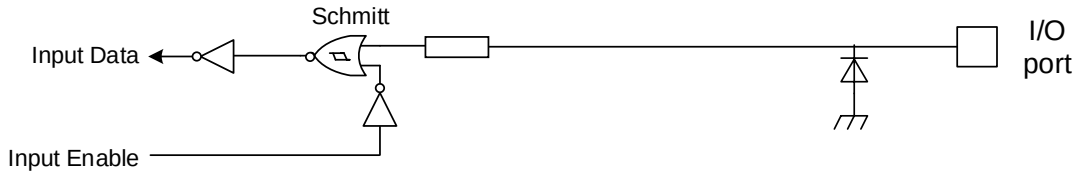
23.1. PA0 to 7, PB0 to 7, PC0 to 7, PD0 to 6, PE0 to 7, PF0 to 4, PG0 to 7



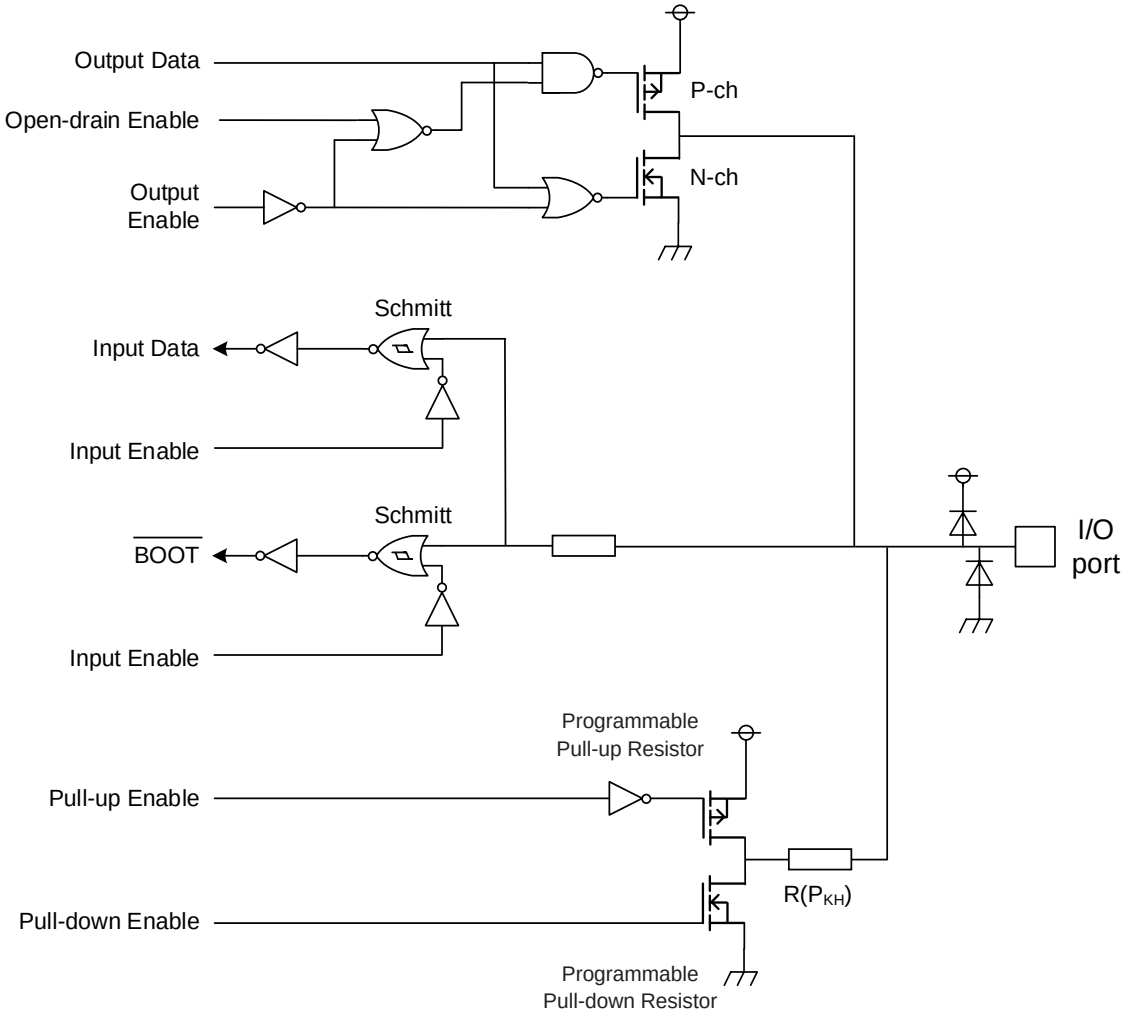
23.2. PH0 to 7, PI0 to 3, PJ0 to 7, PK0, PK1



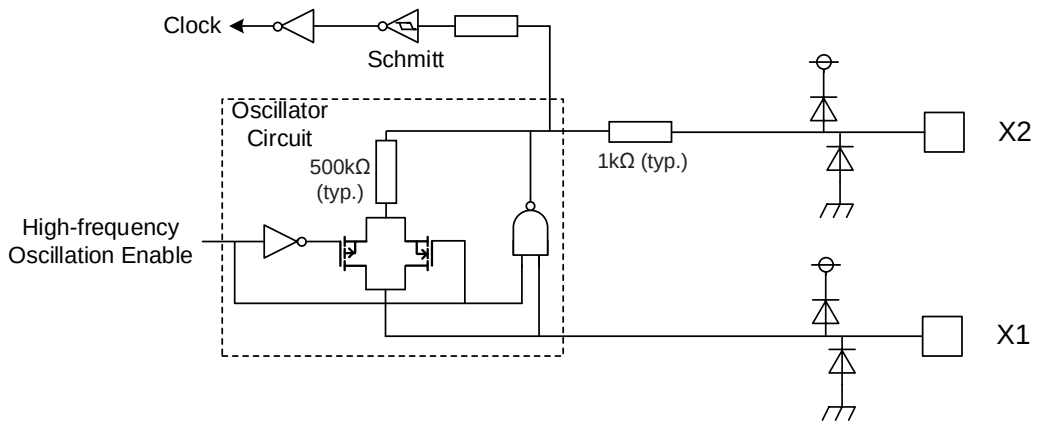
23.3. PL0, PL1



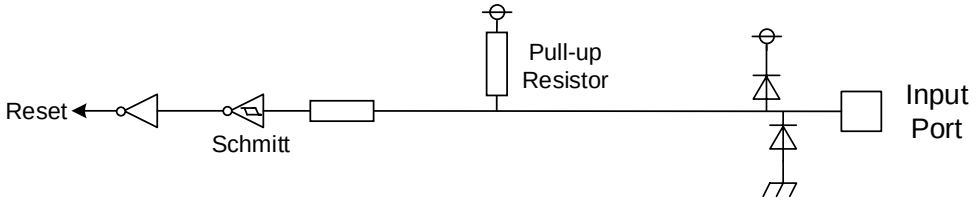
23.4. PF0



23.5. X1, X2



23.6. $\overline{\text{RESET}}$

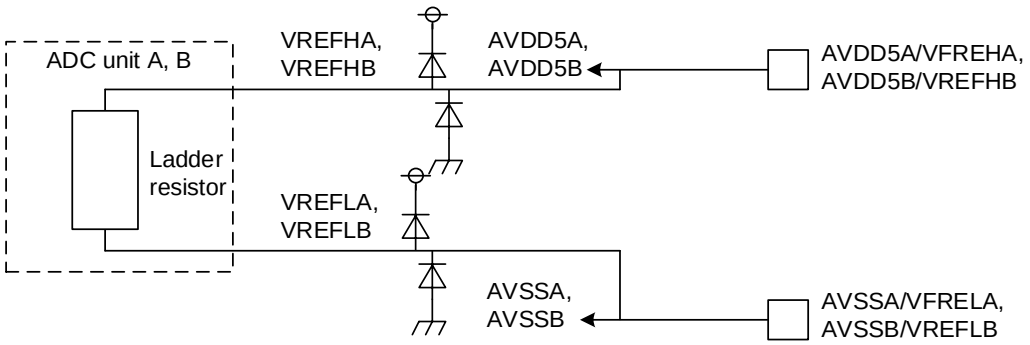


23.7. MODE

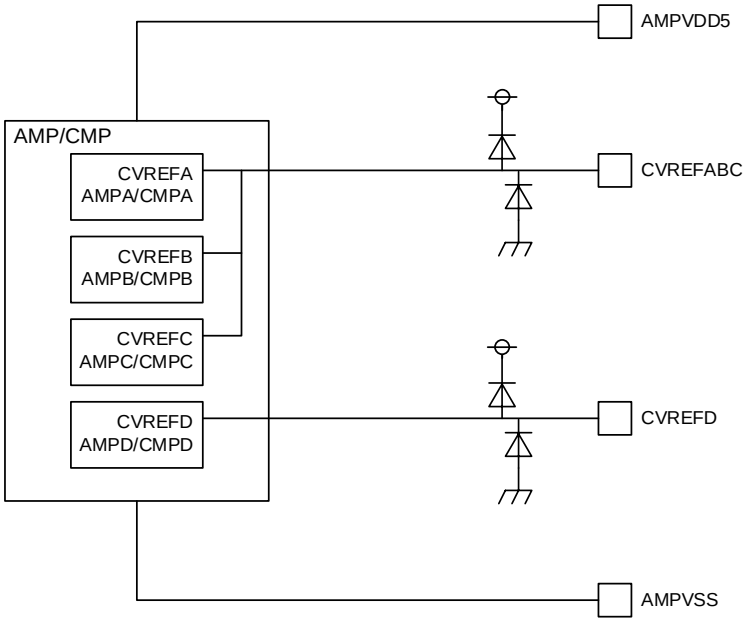


Note: MODE pin must be connected to GND.

23.8. VREFHA, VREFHB, VREFLA, VREFLB



23.9. AMPVDD5, AMPVSS, CREFABC, CREFD



23.10. VOUT15, VOUT3



24. Electrical Characteristics

24.1. Absolute Maximum Ratings

Table 24.1 Absolute Maximum Ratings

Parameter			Symbol	Rating	Unit
Power supply voltage			DVDD5	-0.3 to 6.0	V
			RVDD5	-0.3 to 6.0	
			AVDD5A AVDD5B	-0.3 to 6.0	
			AMPVDD5	-0.3 to 6.0	
Capacitor pin voltage for voltage maintenance			VOUT15	-0.3 to 3.0	V
			VOUT3	-0.3 to 3.9	
Input voltage			V_{IN}	-0.3 to VDD + 0.3 (Note2)	V
Low level output current	Per pin		I_{OL}	5	mA
	Total of all pins		ΣI_{OL}	50	
High level output current	Per pin		I_{OH}	-5	
	Total of all pins		ΣI_{OH}	-50	
Power consumption			PD	600 (Ta = 85°C)	mW
Soldering temperature (10s)			T_{SOLDER}	260	°C
Storage temperature			T_{STG}	-55 to 125	°C
Operating temperature	Except during Flash writing/ erasing	fsys = 1 to 80MHz	T_{OPR}	-40 to 85	°C
	During Flash writing/erasing			0 to 70	

Note1: Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning. Therefore, be sure to design the applied equipment so that the absolute maximum rating is not exceeded.

Note2: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5. Apply the same voltage to DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

24.2. DC Electrical Characteristics (1/2)

DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS =0V

Ta = -40 to 85°C

Parameter		Symbol	Conditions	Min	Typ. (Note2)	Max	Unit
Power supply voltage (Note3) (Note4)	DVDD5、RVDD5、AVDD5A、AVDD5B、AMPVDD5	VDD	f_{OSC} = 8 to 10MHz fsys = 1 to 80MHz	4.5	-	5.5	V
Power Supply Voltage (Flash writing/erasing) (Note3) (Note4)	DVDD5、RVDD5、AVDD5A、AVDD5B、AMPVDD5	VDD	f_{OSC} = 8 to 10MHz fsys = 1 to 80MHz Ta = 0 to 70°C	4.5	-	5.5	V
Power supply voltage (power supply rises/falls) (Note3) (Note4)	DVDD5、RVDD5、AVDD5A、AVDD5B、AMPVDD5	VDD	f_{OSC} = 8 to 10MHz fsys = 1 to 80MHz	3.9	-	5.5	V
Low level input voltage	Schmitt input	V_{IL1}	VDD = 4.5 to 5.5V (Note3)	-0.3	-	VDD × 0.25	V
High level input voltage	Schmitt input	V_{IH1}	VDD = 4.5 to 5.5V (Note3)	VDD × 0.75	-	VDD	V
Capacitance for VOUT15 and VOUT3 (Note5)		C_{OUT}	RVDD5 = 4.5 to 5.5V VOUT15、VOUT3 pin	3.3	-	4.7	μF
Low level output voltage		V_{OL}	VDD ≥ 4.5V (Note3) I_{OL} = 1.6mA	-	-	0.4	V
High level output voltage		V_{OH}	VDD ≥ 4.5V (Note3) I_{OH} = -1.6mA	4.1	-	-	V
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ VDD (Note3)	-5	0.02	5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ VDD – 0.2 (Note3)	-10	0.05	10	
Reset pull-up resistor		R_{RST}	VDD = 4.5 to 5.5V (Note3)	-	50	150	kΩ
Programmable pull-up/pull-down resistor		P_{KH}	VDD = 4.5 to 5.5V (Note3)	-	50	150	kΩ
Schmitt input width		VTH	VDD = 4.5 to 5.5V (Note3)	0.3	0.6	-	V
Pin Capacitance (excluding power supply pins)		C_{IO}	fc = 1MHz	-	-	10	pF

Note1: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Note2: Typ. Value is in Ta = 25°C and DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5 = 5.0V, unless otherwise specified.

Note3: Apply the same voltage to DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Note4: These values is the voltage range when the power supply voltage rises (when the power is turned on) and when the power supply voltage falls with the voltage detection circuit (VLTD) enabled. For details, refer to “Figure 24.10 Power-on sequence (without external reset)”.Figure 24.10 Power-on sequence (without external reset)

Note5: Connect VOUT15 pin and VOUT3 pin to GND via a voltage-holding capacitor of the same value. Power supply cannot be supplied to the external MCU from VOUT15 and VOUT3 pins.

24.3. DC Electrical Characteristics (2/2) (Current Consumption)

VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5 = 4.5 ~ 5.5V

DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS = 0V

Ta = -40 to 85°C

Parameter	Symbol	Conditions	Min	Typ. (Note2)	Max	Unit
NORMAL (Note4) (Note6)	I_{DD}	fsys = 80MHz Gear ratio 1:1	-	70	80	mA
IDLE (Note5) (Note6)			-	21	30	
STOP		-	-	7	11	mA

Note1: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Note2: Typ. Value is in Ta = 25°C and DVDD5 = RVDD5 = VDD5A = AVDD5B = AMPVDD5 = 5.0V, unless otherwise specified.

Note3: Apply the same voltage to DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Note4: Measurement conditions of I_{DD} (NORMAL): All internal peripheral functions except the ADC/AMP/CMP operation.

Note5: Measurement conditions of I_{DD} (IDLE): All internal peripheral functions are stopped.

Note6: The analog reference voltage supply current of the ADC always flows.

24.4. 12-bit analog-to-digital Converter Characteristics

DVDD5 = RVDD5 = AVDD5A/VREFHA = AVDD5B/VREFHB = AMPVDD5 = 4.5 to 5.5V
DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS = 0V
Ta = -40 to 85°C

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Analog reference voltage (+)	VREFHA VREFHB	-	-	AVDD5A AVDD5B	-	V
Analog input voltage	VAIN	-	AVSSA AVSSB	-	AVDD5A AVDD5B	V
Analog reference voltage supply current (Note1) (Note2)	IREF		-	3.5	4.5	mA
Current consumption during AD conversion (Note1)	-	Excluding IREF	-	-	6.0	mA
Integral non-linearity error (INL)	-	AIN load resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 μF Conversion time ≥ 2μs	-6.0	-	6.0	LSB
Differential non-linearity error (DNL)			-5.0	-	5.0	
Offset error			-5.0	-	5.0	
Full scale error			-5.0	-	5.0	
Total errors			-10	-	5.0	

Note1: This is a value for 1 unit.

Note2: The analog reference voltage supply current of the ADC always flows.

Note3: 1LSB = (AVDD5A/VREFHA – AVSSA/VREFLAB) / 4096[V]
1LSB = (AVDD5B/VREFHB – AVSSB/VREFLB) / 4096[V]

Note4: Characteristics when the AD converter is operated independently.

Note5: For details of the setting, refer to “11. 12-Bit Analog-to-Digital Converter (ADC)”.

24.5. Op-amplifiers Electrical Characteristics

DVDD5 = RVDD5 = AVDD5A/VREFHA = AVDD5B/VREFHB = AMPVDD5 = 4.5 to 5.5V
DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS = 0V
Ta = -40 to 85°C

Parameter	Symbol	Condition		Min	Typ.	Max	Unit
Gain	VGAIN	-		1.5	-	10	times
Op-amplifier input voltage range	VAMPIN	-		(AVDD × 0.1) / VGAIN	-	(AVDD × 0.9) / VGAIN	V
Offset voltage	VOFF1	VGAIN≧3.5		-6 × VGAIN	-	+6 × VGAIN	mV
		VGAIN≦3.5	Ta≧70°C				
	VOFF2		Ta<70°C	-20		20	
Gain error	GERR	-		-	±1	±3	%
Slew rate (Note2)	Vthr	CL=5pF、VGAIN=2.5 倍		2	-	-	V/μs
Supply current	-	When AMP is used. Supply current for 1 channel.		-	-	6	mA

Note1: AVDD is a generic name for AVDD5A/VREFHA and AVDD5B/VREFHB. AVSS is a generic name for AVSSA/VREFLA and AVSSB/VREFLB.

Note2: Slew rate means the slant until a output voltage of an op-amplifier reaches to AVDD – AVDD × 0.001 V.

24.6. Analog Comparators Electrical Characteristics

DVDD5 = RVDD5 = AVDD5A/VREFHA = AVDD5B/VREFHB = AMPVDD5 = 4.5 to 5.5V
DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS = 0V
Ta = -40 to 85°C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Offset voltage	VOFF	-	-	±4	-	mV
Analog comparator input pin voltage range	VIN	-	AVSS		AVDD	V
Reference voltage range	VREF	-	0.9	-	AVDD – 0.2	
Response time (Note2)	TP	1.0V≦VREF≦AVDD-0.2	-	-	1	μs
Supply current	-	When CMP is used. Supply current for 1 channel.	-	-	0.75	mA

Note1: AVDD is a generic name for AVDD5A/VREFHA and AVDD5B/VREFHB. AVSS is a generic name for AVSSA/VREFLA and AVSSB/VREFLB.

Note2: TP is the value when VIN changes from VREF – 0.1 V to VREF + 0.1 V or from VREF + 0.1 V to VREF – 0.1 V.

24.7. AC Electrical Characteristics

24.7.1. AC Conditions for measurement

The AC characteristics described in this chapter are under the following conditions, unless otherwise specified.

- VDD = 4.5 to 5.5V
- Ta =-40 to 85°C, fsys = 1 to 80MHz
- Output level: High = VDD × 0.8, Low = VDD × 0.2
- Input level: refer to “Low level input voltage”/”High level input voltage” in DC Electrical Characteristics.
- Load capacitance: C_L = 30pF

Note1: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Note2: Apply the same voltage to DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

24.7.2. Serial Channel (SIO/UART)

24.7.2.1. SIO mode

"x" in the tables indicates the cycle of SIO/UART operating clock. SIO/UART operating clock is the system clock fsys. Fsys depends on the clock gear setting.

(1) When the transfer clock is input to SCLKx pin

(a) Data input

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
SCLKx input clock "High" level width	t_{SCH}	3x	-	37.5	-	ns
SCLKx input clock "Low" level width	t_{SCL}	3x	-	37.5	-	
SCLKx input clock cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	-	75	-	
Valid data input → SCLKx input clock rising (SCxMOD0<SCLKS>= "0")	t_{SRD}	30	-	30	-	
SCLKx input clock rising → input data hold (SCxMOD0<SCLKS>= "0")	t_{HSR}	x + 30	-	42.5	-	
Valid data input → SCLKx input clock falling (SCxMOD0<SCLKS>= "1")	t_{SRD}	30	-	30	-	
SCLKx input clock falling → input data hold (SCxMOD0<SCLKS>= "1")	t_{HSR}	x + 30	-	42.5	-	

(b) Data output

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
SCLKx input clock "High" level width	t_{SCH}	3x	-	37.5 (Note2)	-	ns
SCLKx input clock "Low" level width	t_{SCL}	3x	-	37.5 (Note2)	-	
SCLKx input clock cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	-	75	-	
Data output → SCLKx input clock rising (SCxMOD0<SCLKS>= "0")	t_{OSS}	t_{SCY} / 2 – 4x – 45	-	0 (Note1)	-	
SCLKx input clock rising → output data hold (SCxMOD0<SCLKS>= "0")	t_{OHS}	t_{SCY} /2	-	37.5	-	
Data output → SCLKx input clock falling (SCxMOD0<SCLKS>= "1")	t_{OSS}	t_{SCY} / 2 – 4x – 45	-	0 (Note2)	-	
SCLKx input clock falling → output data hold (SCxMOD0<SCLKS>= "1")	t_{OHS}	t_{SCY} /2	-	37.5	-	

Note1: Use SCLKx clock cycle at which the calculated result becomes "0" or more.

Note2: This value indicates the minimum value when t_{OSS} is greater than "0" or more.

- When SCxMOD0<SCLKS> = "0"
- When SCxMOD0<SCLKS> = "1"

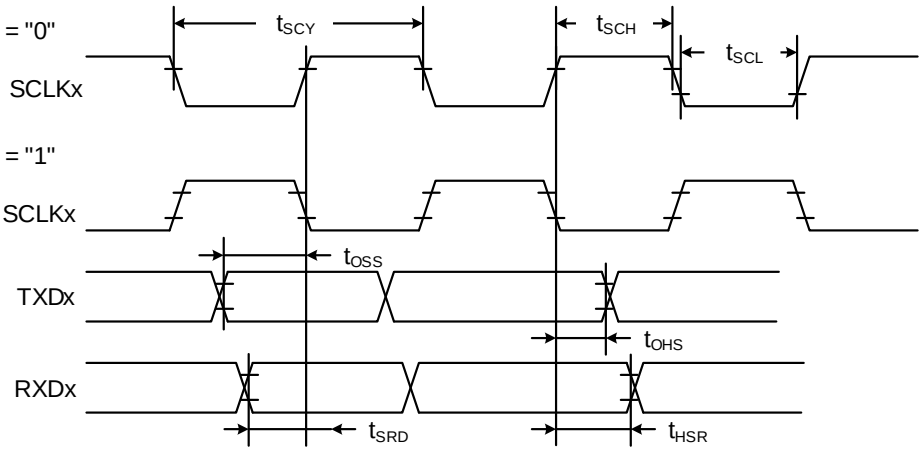


Figure 24.1 Timing Chart When Transfer Clock is Input on SCLKx Pin

(2) When the transfer clock is SCLKx pin output

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
SCLKx output clock cycle	t_{SCY}	4x	-	60	-	ns
Data output → SCLKx output clock rising	t_{OSS}	t_{SCY}/2-30 (Note1)	-	0 (Note2)	-	
SCLKx output clock rising → output data hold	t_{OHS}	t_{SCY}/2-30 (Note1)	-	0 (Note2)	-	
Valid data input → SCLKx output clock rising	t_{SRD}	45		45		
SCLKx output clock rising → input data hold	t_{HSR}	0	-	0	-	

Note1: Use SCLKx output clock cycle at which the calculation result becomes “0” or more.

Note2: This value indicates the minimum value when t_{OSS} is “0” or more.

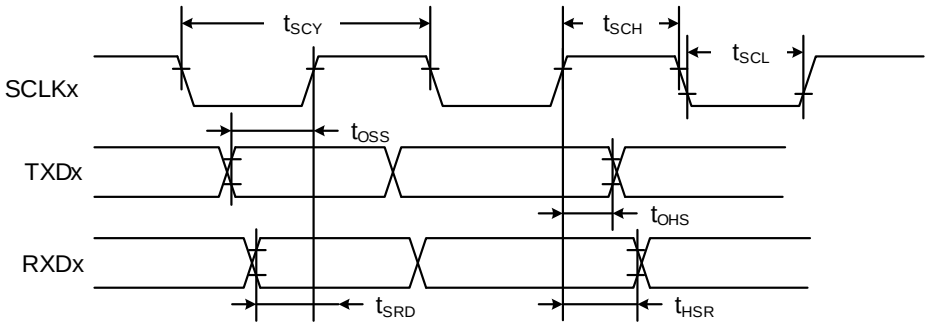


Figure 24.2 Timing Chart When Transfer Clock Is Output on SCLKx Pin

24.7.3. 16-bit Timer/Event Counter (TMRB)

"x" in the table indicates the cycle of TMRB operating clocks. TMRB operating clock is the system clock fsys. Fsys depends on the clock gear setting.

24.7.3.1. Event counter TbxIN Pin Input

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
"Low" level width	t_{VCKL}	2x + 100	-	125	-	ns
"High" level width	t_{VCKH}	2x + 100	-	125	-	

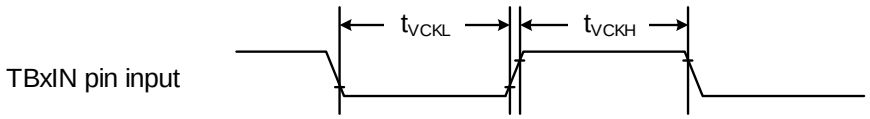


Figure 24.3 Timing Chart of Event Counter

24.7.3.2. Capture TbxIN Pin Input

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
"Low" level width	t_{CPL}	2x + 100	-	125	-	ns
"High" level width	t_{CPH}	2x + 100	-	125	-	

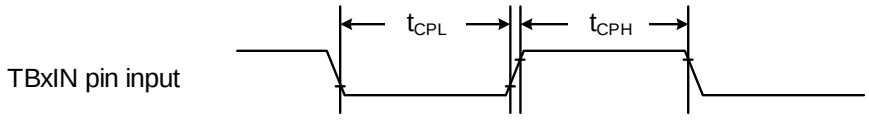


Figure 24.4 Timing Chart of Capture

24.7.4. External Interrupt Pin

"x" in the table indicates the cycle of the operation clock of the external interrupt pin. The operation clock of the external interrupt pin is the system clock fsys. Fsys depends on the clock gear setting.

24.7.4.1. When using an external interrupt other than the following

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
"Low" level width	t_{INTAL}	x + 100	-	112.5	-	Ns
"High" level width	t_{INTAH}	x + 100	-	112.5	-	

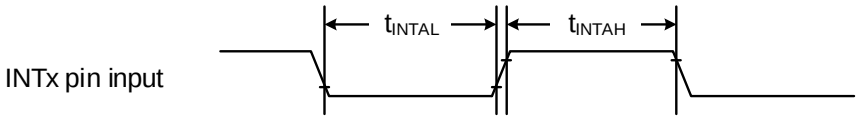


Figure 24.5 Timing Chart When Using External Interrupts Other than the following

24.7.4.2. When using an external interrupt pin for releasing STOP mode

Parameter	Symbol	Calculation formula		fsys = 80MHz		Unit
		Min	Max	Min	Max	
"Low" level width	t_{INTBL}	100	-	100	-	Ns
"High" level width	t_{INTBH}	100	-	100	-	



Figure 24.6 Timing chart When Releasing STOP Mode by External Interrupt Pin

24.7.5. Debug communication

24.7.5.1. AC Conditions for measurement

- Output level: High = VDD × 0.7, Low = VDD × 0.3
- Load capacitance: TRACECLK pin: C_L = 25pF, TRACEDATAn pin: C_L = 20pF

Note1: VDD is a generic name for DVDD5, RVDD5, AVDD5B.
Note2: Apply the same voltage to DVDD5, RVDD5 and AVDD5B.

24.7.5.2. SWD interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	t_{dck}	100	-	ns
CLK rising → output data hold	t_{d1}	4	-	
CLK rising → output data valid	t_{d2}	-	37	
Input data valid → CLK rising	t_{ds}	20	-	
CLK rising → input data hold	t_{dh}	15	-	

24.7.5.3. JTAG interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	t_{dck}	100	-	ns
CLK falling → output data hold	t_{d3}	4	-	
CLK falling → output data valid	t_{d4}	-	37	
Input data valid → CLK rising	t_{ds}	20	-	
CLK rising → input data hold	t_{dh}	15	-	

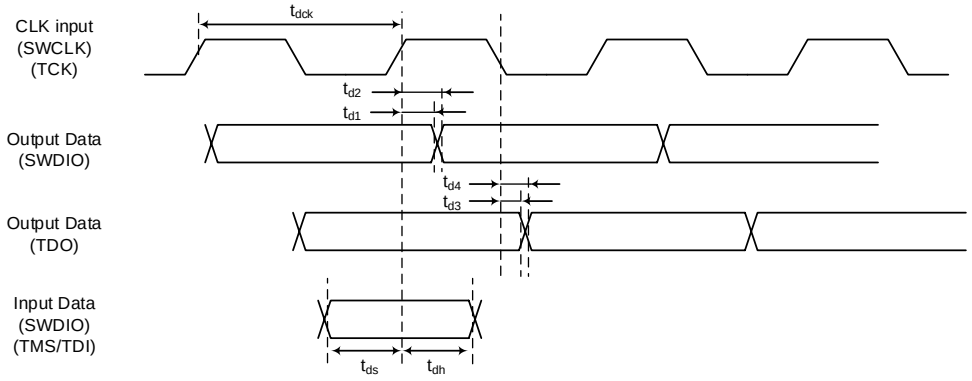


Figure 24.7 Timing Chart of JTAG/SWD

24.7.5.4. ETM trace

Parameter	Symbol	Min	Max	Unit
TRACECLK cycle	t_{tclk}	25	-	ns
TRACEDATAn valid → TRACECLK rising	t_{setupr}	2	-	
TRACECLK rising → TRACEDATAn hold	t_{holdr}	1	-	
TRACEDATAn valid → TRACECLK falling	t_{setupf}	2	-	
TRACECLK falling → TRACEDATAn hold	t_{holdf}	1	-	

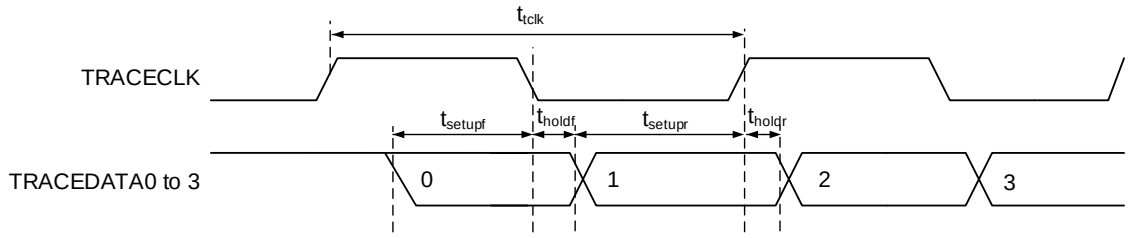


Figure 24.8 Timing Chart of ETM trace

24.8. Flash Characteristic

VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5 = 4.5 ~ 5.5V

DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS = 0V

Ta = 0 to 70°C

Parameter	Conditions	Min	Typ.	Max	Unit
Flash erasing/writing count	-	-	-	100	times

24.9. Sample Oscillation Circuit

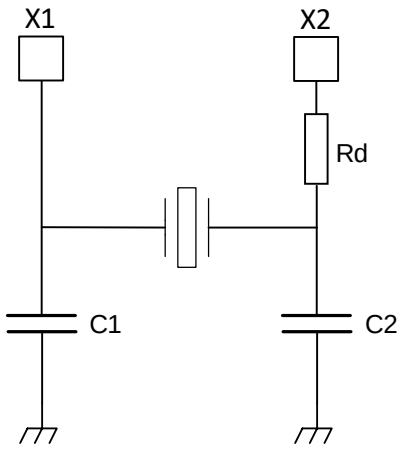


Figure 24.9 Example of high-frequency oscillation circuit

To stabilize the oscillation, the position of the oscillator and the load capacitance must be appropriate. These are greatly affected by the board pattern. In order to obtain stable oscillation, evaluate on the board to be used.

24.10. Ceramic resonator

TMPM372FWUG/TMPM373FWDUG/TMPM374FWUG have been evaluated by the ceramic resonator by Murata Manufacturing Co., Ltd.

Please refer to the Murata Website for details.

24.11. Notice When Power ON

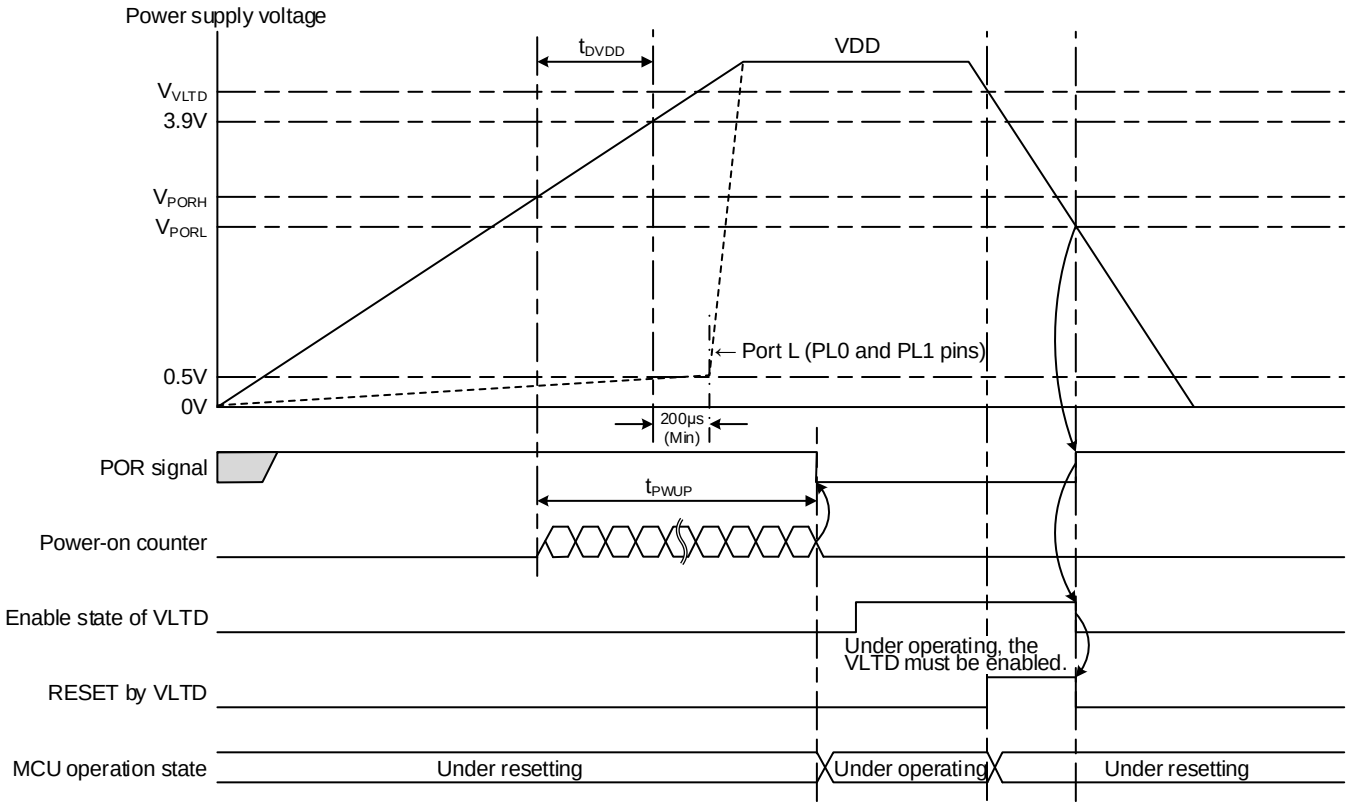
24.11.1. Port L (PL0 and PL1) State When Supply Voltage Rises

When the supply voltage rises, until power supply voltage reaches the operating voltage range (4.5V to 5.5V) and 200μs elapses, Port L (PL0 and PL1) must be opened or input voltage to Port L must be 0.5 V or less.

Same notice is required when supply voltage rises again after the supply voltage falls and a power on reset signal is generated by POR.

24.11.2. When the external reset is not used

Parameter	Symbol	Min	Typ.	Max	Unit
Warming-up time after releasing reset	t_{PWUP}	-	-	3.7	ms
Power supply rising time	t_{DVDD}	-	-	3	
VLTD detection voltage (VDCR<VDLVL>= “01”)	V_{VLTD}	3.9	4.1	4.3	V
POR release voltage	V_{PORH}	2.8	3	3.2	
POR detection voltage	V_{PORL}	2.6	2.8	3	



Note1: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Note2: When POR is used in turning power on, VDD must reach the operating voltage range (4.5 to 5.5V) within 3ms.

Note3: VLTD is enabled during MCU operation.

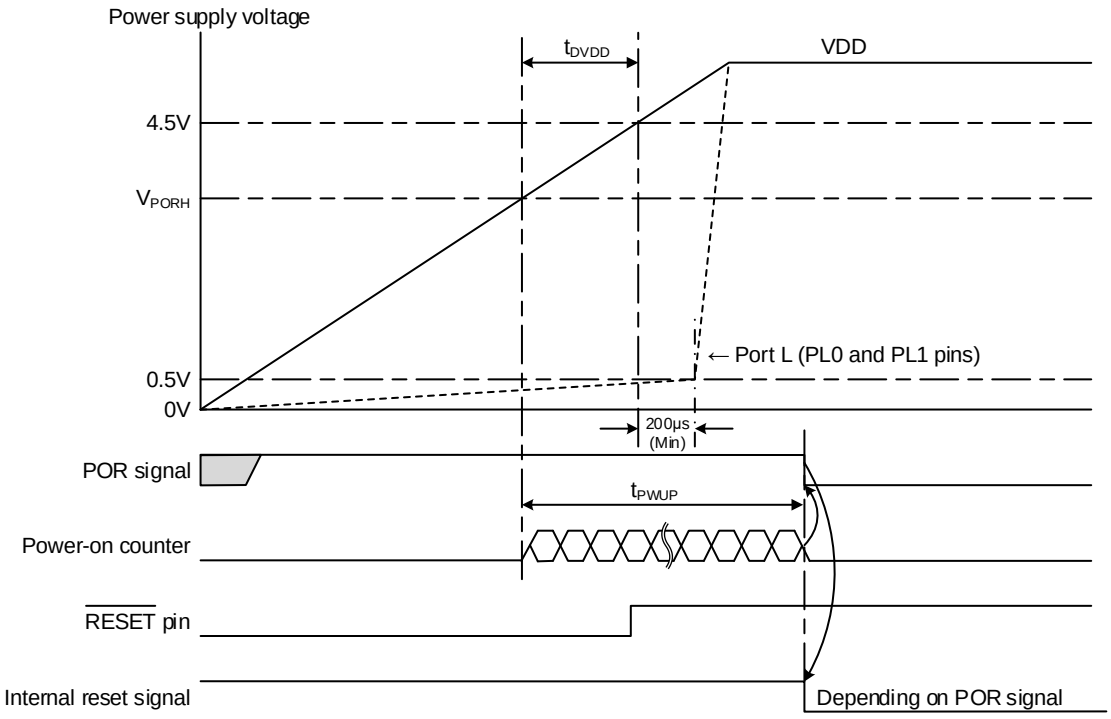
Note4: Because the POR release voltage (V_{PORH}) and the detected voltage (V_{PORL}) change relatively, the respective voltage does not reverse.

Figure 24.10 Power-on sequence (without external reset)

24.11.3. When using an external reset

24.11.3.1. When releasing external reset is earlier than the releasing POR reset

The release of the internal reset depends on the POR reset.

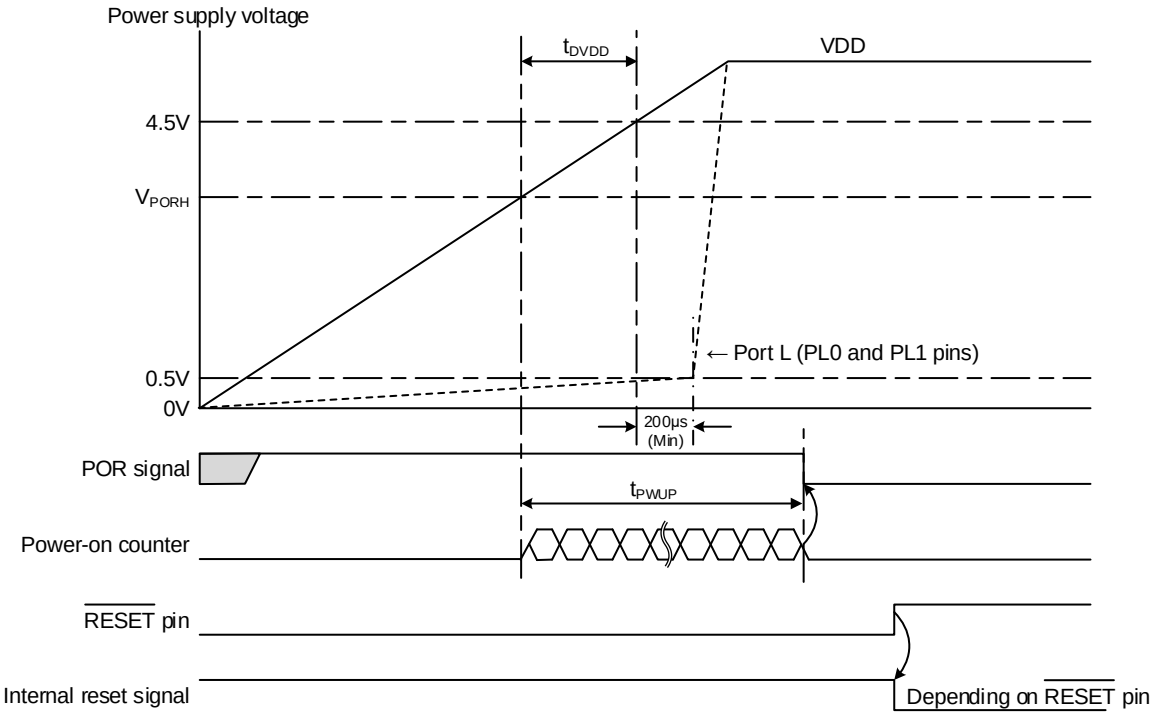


Note: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Figure 24.11 Timing chart when releasing external reset is earlier than releasing POR reset

24.11.3.2. When releasing external reset is slower than releasing POR reset

The release of the internal reset depends on the $\overline{\text{RESET}}$ pin.

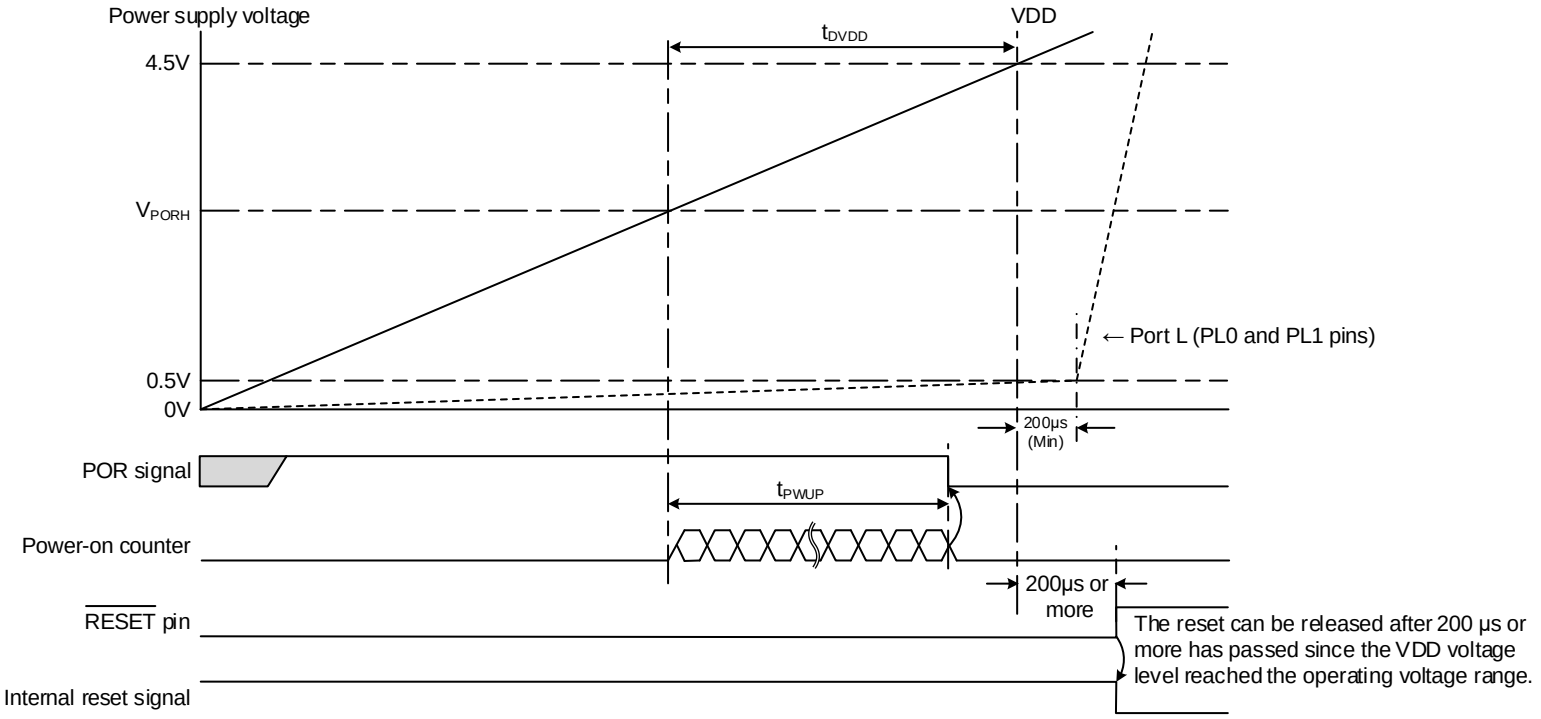


Note: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Figure 24.12 Timing chart when releasing external reset is slower than releasing POR reset

24.11.3.3. When t_{DVDD} is longer than t_{PWUP}

Turn the $\overline{\text{RESET}}$ pin to the "High" level after 200µs or more has elapsed when the VDD voltage exceeded the operating voltage.



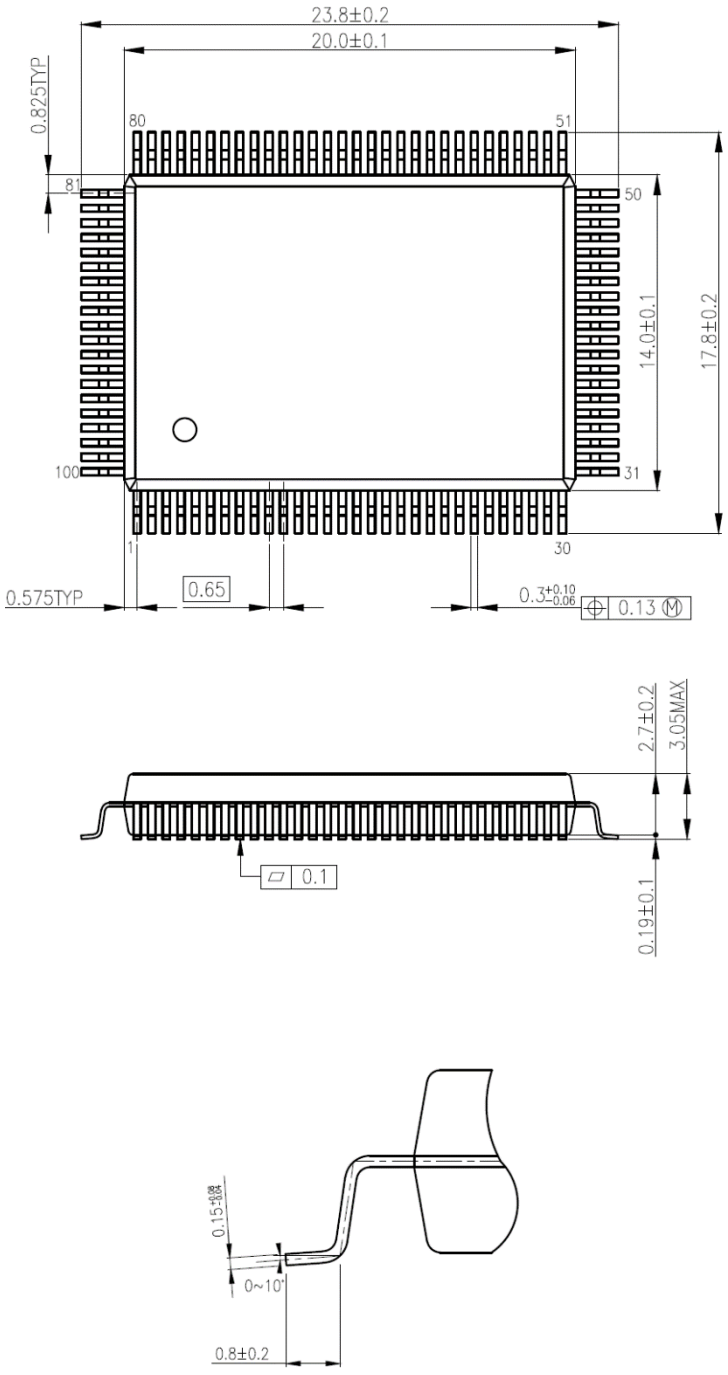
Note: VDD is a generic name for DVDD5, RVDD5, AVDD5A, AVDD5B and AMPVDD5.

Figure 24.13 Timing chart when t_{DVDD} is longer than t_{PWUP}

25. Package Dimensions

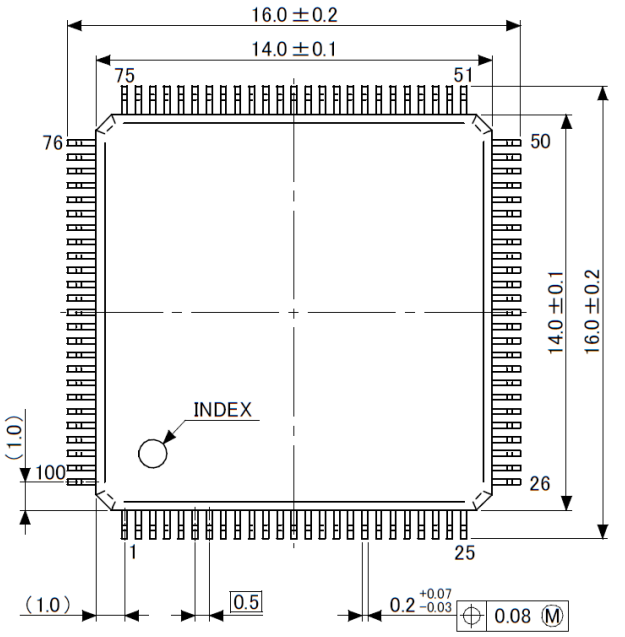
25.1. Type: QFP100-P-1420-0.65Q

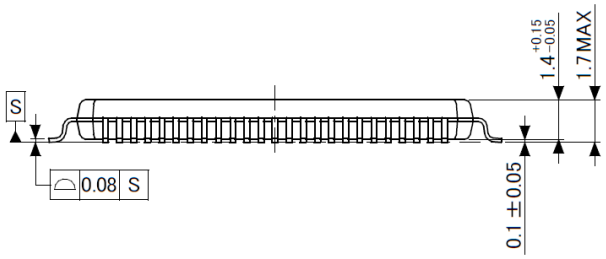
Unit: mm

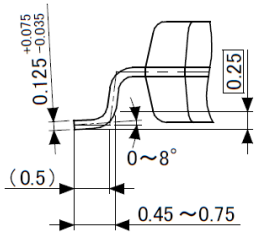


25.2. Type: LQFP100-P-1414-0.50H

Unit: mm







26. Revision History

Table 26.1 Revision History

Revision	Date	Description
1.0	2022-08-01	First release
1.1	2023-07-21	- Figure 5.1 Changed figure - 10.13.1.2. When FIFO is Enabled Changed text and Table 10.14 - 10.13.2.2. When FIFO is Enabled Changed text and Table 10.16 - 11.5.2. Starting AD Conversion Changed error - Table 11.6 Changed title - Figure 11.7 Changed figure - Figure 11.8 Changed figure - 13.3.4.3. VEFMODEx (Flow Control Register) Added <CRCEN> to bit8 - Table 13.15 Added trigger correction enable - Table 22.3 Added (Note2)

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.

Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, Class 3 medical devices, equipment used for automobiles, and military vehicles and munitions. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**