32-Bit RISC Microprocessor TX19 Family

TMP1962C10BXBG

1. Features

The TX19 is a family of high-performance 32-bit microprocessors that offers the speed of a 32-bit RISC solution with the added advantage of a significantly reduced code size of a 16-bit architecture. The instruction set of the TX19 includes as a subset the 32-bit instructions of the TX39, which is based on the MIPS R3000A™ architecture. Additionally, the TX19 supports the MIPS16™ Application-Specific Extensions (ASE) for improved code density.

The TMP1962 is built on a TX19 core processor and a selection of intelligent peripherals. The TMP1962 is suitable for low-voltage, low-power applications.

Features of the TMP1962 include the following:

(1) TX19 core processor

1) Two instruction set architecture (ISA) modes: 16-bit ISA for code density and 32-bit ISA for speed

- The 16-bit ISA is object-code compatible with the code-efficient MIPS16™ ASE.
- The 32-bit ISA is object-code compatible with the high-performance TX39 family.

2) Combines high performance with low power consumption.

- High performance
  - Single clock cycle execution for most instructions
  - 3-operand computational instructions for high instruction throughput
  - 5-stage pipeline

- On-chip high-speed memory
- DSP function: Executes 32-bit x 32-bit multiplier operations with a 64-bit accumulation in a single clock cycle.

- Low power consumption
  - Optimized design using a low-power cell library
  - Programmable standby modes in which processor clocks are stopped

3) Fast interrupt response suitable for real-time control

- Distinct starting locations for each interrupt service routine
- Automatically generated vectors for each interrupt source
- Automatic updates of the interrupt mask level

(2) On-chip program memory and data memory

Product	On-Chip ROM	On-Chip RAM
TMP1962C10AXB	1 Mbyte	40 Kbyte
TMP1962F10AXB	1 Mbyte (Flash)	40 Kbyte

- ROM correction logic (8 words x 8 blocks)

(3) External memory expansion

- 16-Mbyte off-chip address space for code and data
- External data bus

  Separate bus/multiplexed bus: Dynamic bus sizing for 8-bit and 16-bit data ports

(4) 8-channel DMA controller

- Interrupt- or software-triggered
- Transfer destination: On-chip memory, on-chip peripherals, external memory, external peripherals

(5) 12-channel 8-bit timer

- 8/16/24/32-Bit Interval Timer mode
- 8-Bit PWM mode
- 8-Bit PPG mode

(6) 4-channel 16-bit timer

- 16-Bit Interval Timer mode
- 16-Bit Event Counter mode
- 16-bit PPG output
- Input capture
- 2-phase pulse input counter (2 channels)

(7) 32-bit input capture

- 32-bit input capture registers (8 channels)
- 32-bit compare registers (8 channels)
- 32-bit time base timer (1 channel)

(8) 7-channel general-purpose serial interface

- Either UART mode or Synchronous mode can be selected.

(9) 1-channel serial bus interface

- Either I^2C Bus mode or Clock-Synchronous mode can be selected.

(10) 24-channel 10-bit A/D converter (with internal sample/hold)

- External trigger supported
- Fixed-Channel or Channel Scan mode
- Single Conversion or Continuous Conversion mode
- Timer monitoring

(11) 1-channel watchdog timer

(12) 4-channel chip select/wait controller

(13) Interrupt sources

- 4 CPU interrupts: Software interrupt instruction
- 55 internal interrupts: 7 priority levels, with the exception of the watchdog timer interrupt
- 25 external interrupts: 7 priority levels, with the exception of the NMI interrupt
  The external sources include 14 KWUP sources, which are all assigned to a single interrupt vector.

(14) 202-pin input/output ports

(15) Standby modes

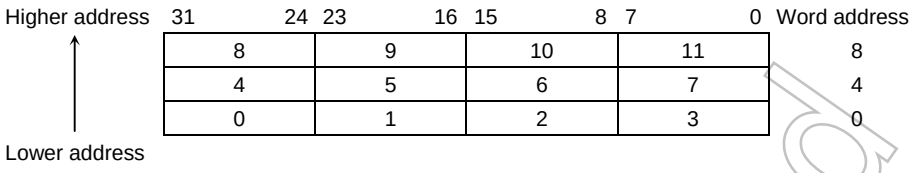
- Two standby modes: IDLE, STOP

(16) Clock generator

- On-chip PLL (x3)
- Clock gear: Divides the high-speed clock by 1/2, 1/4 or 1/8.

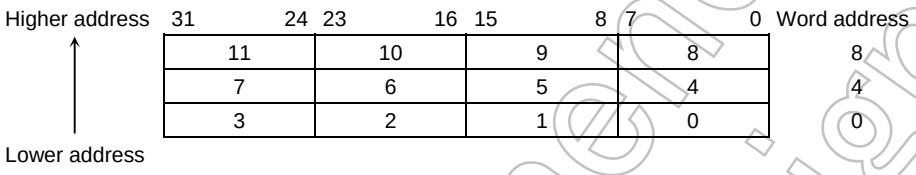
(17) Optional big-endian alignment

Big-endian



- Byte 0 is the highest-order byte (bits 31-24).
- The address of a word data item is the address of its highest-order byte (byte 0).

Little-endian



- Byte 0 is the lowest-order byte (bits 7-0).
- The address of a word data item is the address of its lowest-order byte (byte 0).

(18) Operating frequency

- 40.5 MHz (Vcc = 1.35 V to 1.65 V)

(19) Package

- P-FBGA281 (13 mm x 13 mm, 0.65-mm pitch)

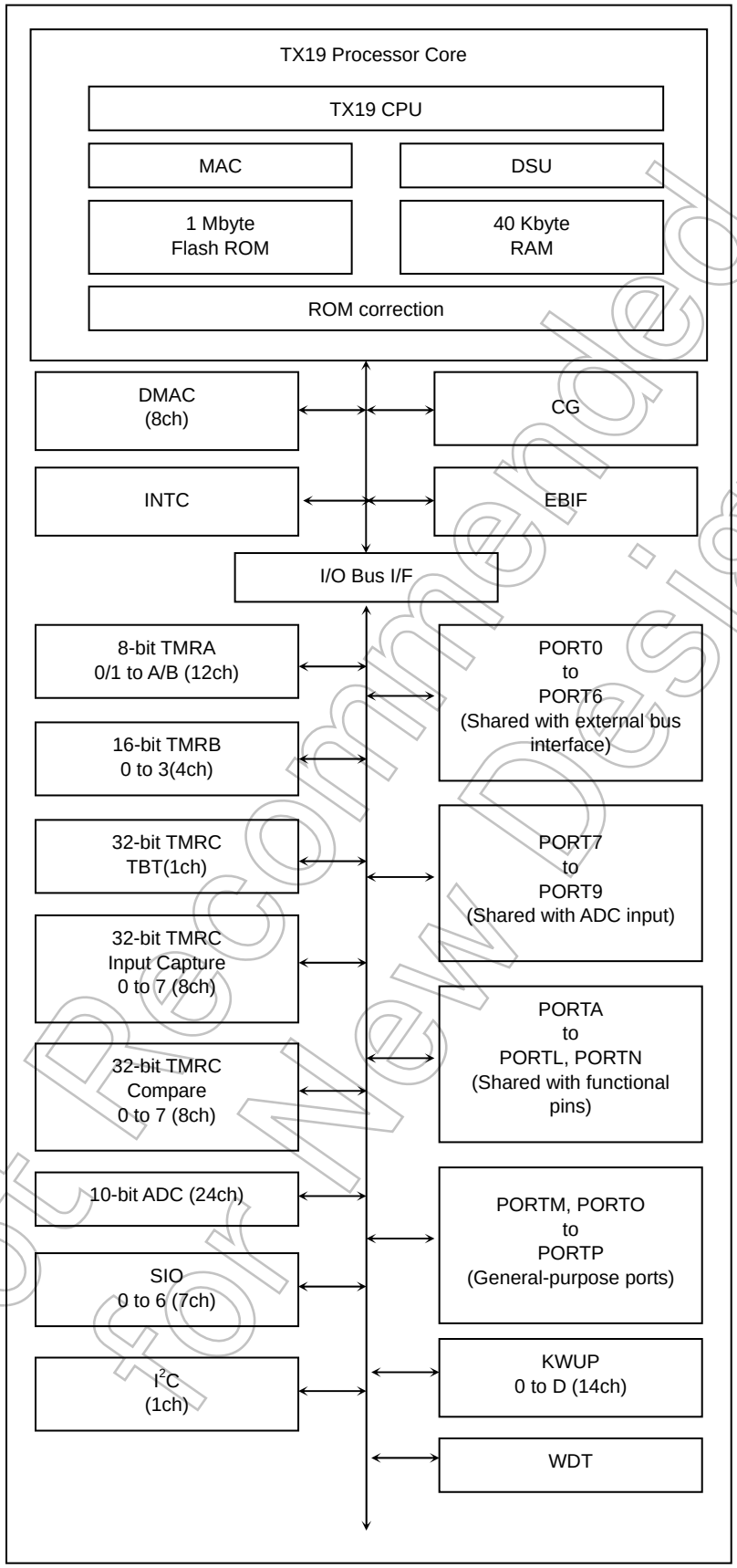


Figure 1.1 TMP1962 Block Diagram

2. Signal Descriptions

This section contains pin assignments for the TMP1962 as well as brief descriptions of the TMP1962 input and output signals.

2.1 Pin Assignment (TOP View)

The following illustrates the TMP1962 pin assignment.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18
F1	F2	F3	F4	F5		F7	F8	F9	F10	F11	F12		F14	F15	F16	F17	F18
G1	G2	G3	G4	G5	G6							G13	G14	G15	G16	G17	G18
H1	H2	H3	H4	H5	H6							H13	H14	H15	H16	H17	H18
J1	J2	J3	J4	J5	J6							J13	J14	J15	J16	J17	J18
K1	K2	K3	K4	K5	K6							K13	K14	K15	K16	K17	K18
L1	L2	L3	L4	L5	L6							L13	L14	L15	L16	L17	L18
M1	M2	M3	M4	M5	M6							M13	M14	M15	M16	M17	M18
N1	N2	N3	N4	N5		N7	N8	N9	N10	N11	N12		N14	N15	N16	N17	N18
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18
	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	

Figure 2.1 P-FBGA281 Pin Assignment

Table 2.1 shows the correspondence between the numbers and names of the TMP1962 pins.

Table 2.1 Pin Numbers and Names (1/2)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
A1	NC	A13	PK1/KEY1	B8	P75/AIN5	C2	PCST3 (DSU)	C14	PK6/KEY6
A2	VREFL	A14	PI1/INT1	B9	PL0/TA4IN	C3	P92/AIN18	C15	PI5/INT9
A3	P90/AIN16	A15	PI3/INT3	B10	PL3/TAAIN	C4	P95/AIN21	C16	TCK (JTAG)
A4	P93/AIN19	A16	PI6/INTA	B11	PM1	C5	P82/AIN10	C17	CVCC15 (CVCC2)
A5	P80/AIN8	A17	X2	B12	PM4	C6	P85/AIN13	C18	NC (XT2)
A6	P83/AIN11	B1	AVCC31	B13	PK2/KEY2	C7	P72/AIN2	D1	SDAO/TPC (DSU)
A7	P70/AIN0	B2	VREFH	B14	PI2/INT2	C8	AVSS	D2	PCST2 (DSU)
A8	P74/AIN4	B3	P91/AIN17	B15	PI4/INT4	C9	PL1/TA6IN	D3	SDI/ $\overline{\text{DINT}}$ (DSU)
A9	NC	B4	P94/AIN20	B16	PI7	C10	PL4/TB0IN0	D4	DVCC15 (DVCC22)
A10	PL2/TA8IN	B5	P81/AIN9	B17	CVSS	C11	PM2	D5	P96/AIN22
A11	PM0	B6	P84/AIN12	B18	X1	C12	PM5	D6	P86/AIN14
A12	PK0/KEY0	B7	P71/AIN1	C1	PCST0 (DSU)	C13	PK3/KEY3	D7	P73/AIN3

Table 2.1 Pin Numbers and Names (2/2)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
D8	DVCC15 (DVCC22)	F18	P44/SCOUT	K14	P12/D10/AD10	N18	DVSS	T8	PD4/TXD4
D9	DVSS	G1	$\overline{\text{RESET}}$	K15	P13/D11/AD11	P1	PP0	T9	PC0/TXD0
D10	PL5/TB0IN1	G2	TEST5	K16	P14/D12/AD12	P2	PB2/TB2IN0/INT5	T10	PC3/TXD1
D11	PM3	G3	DVCC2 (FVCC2)	K17	DVCC33	P3	PB3/TB2IN1/INT6	T11	PH4/TCOUT4
D12	PM6	G4	NC (FVSS)	K18	P15/D13/AD13	P4	PB4/TB2OUT	T12	PE2/SCLK5/ $\overline{\text{CTS5}}$
D13	PK4/KEY4	G5	PJ0/INT0	L1	NC (FVCC3)	P5	PB5/TB3IN0/INT7	T13	PE5/KEYB
D14	PK7/KEY7	G6	BW0	L2	PO1	P6	PG5/TC5IN	T14	P53/A3
D15	DVCC34	G13	$\overline{\text{TRST}}$	L3	PO2	P7	PG7/TC7IN	T15	P56/A6
D16	TDI (JTAG)	G14	NC (CAP1)	L4	PO3	P8	PD6/SCLK4/ $\overline{\text{CTS4}}$	T16	P62/A10
D17	TDO (JTAG)	G15	P41/ $\overline{\text{CS1}}$	L5	PO4	P9	PC2/SCLK0/ $\overline{\text{CTS0}}$	T17	P65/A13
D18	NC (XT1)	G16	P37/ALE	L6	PO7	P10	PC5/SCLK1/ $\overline{\text{CTS1}}$	T18	P20/A16/A0
E1	DCLK (DSU)	G17	P35/ $\overline{\text{BUSAK}}$	L13	NC (TEST3)	P11	PH6/TCOUT6	U1	PA0/TA0IN
E2	PCST1 (DSU)	G18	NC (FVCC2)	L14	P06/D6/AD6	P12	NC	U2	PA3/TA3OUT
E3	$\overline{\text{DBGE}}$	H1	$\overline{\text{NMI}}$	L15	NC (FVCC2)	P13	P50/A0	U3	PA6/TA9OUT
E4	PJ3/INTLV	H2	DVCC31	L16	P07/D7/AD7	P14	P51/A1	U4	PF1/SI/SCL
E5	PJ4/ENDIAN	H3	PN7	L17	P10/D8/AD8	P15	P54/A4	U5	PF5/ $\overline{\text{DREQ3}}$
E6	P97/AIN23	H4	BW1	L18	P11/D9/AD9	P16	P23/A19/A3	U6	PG2/TC2IN
E7	P87/AIN15	H5	$\overline{\text{PLLOFF}}$	M1	PO0	P17	P24/A20/A4	U7	PD2/RXD3
E8	P76/AIN6	H6	NC (TEST1)	M2	PP5	P18	P25/A21/A5	U8	DVCC32
E9	P77/AIN7	H13	NC (TEST2)	M3	PP6	R1	PB0/TB0OUT	U9	PC7/RXD2
E10	PL6/TB1IN0	H14	P31/ $\overline{\text{WR}}$	M4	PP7	R2	PB1/TB1OUT	U10	PH1/TCOUT1
E11	PL7/TB1IN1	H15	P32/ $\overline{\text{HWR}}$	M5	PB7/TB3OUT	R3	PF3/ $\overline{\text{DREQ2}}$	U11	PH3/TCOUT3
E12	PM7	H16	P33/WAIT/RDY	M6	DVCC32	R4	PF4/ $\overline{\text{DACK2}}$	U12	PE1/RXD5
E13	PK5/KEY5	H17	P30/ $\overline{\text{RD}}$	M13	NC (TEST4)	R5	PF7/TBTIN	U13	PE4/KEYA
E14	NC	H18	P40/ $\overline{\text{CS0}}$	M14	P02/D2/AD2	R6	PG4/TC4IN	U14	DVCC32
E15	TMS (JTAG)	J1	PN2/SCLK6/ $\overline{\text{CTS6}}$	M15	NC (FVSS)	R7	PG6/TC6IN	U15	P57/A7
E16	NC (CVCCH)	J2	PN3	M16	P03/D3/AD3	R8	PD5/RXD4	U16	P63/A11
E17	NC	J3	PN4	M17	P04/D4/AD4	R9	PC1/RXD0	U17	P66/A14
E18	DVCC15 (DVCC22)	J4	PN5	M18	P05/D5/AD5	R10	PC4/RXD1	U18	DVCC33
F1	DVSS	J5	PN6	N1	PP1	R11	PH5/TCOUT5	V2	PA2/TA2IN
F2	$\overline{\text{DRESET}}$	J6	DVCC15 (DVCC22)	N2	PP2	R12	PH7/TCOUT7	V3	PA5/TA7OUT
F3	SYSRDY	J13	NC (FVSS)	N3	PP3	R13	PE6/KEYC	V4	PF0/SO/SDA
F4	PJ1/BUSMD	J14	P16/D14/AD14	N4	PP4	R14	P52/A2	V5	PG0/TC0IN
F5	PJ2/ $\overline{\text{BOOT}}$	J15	DVSS	N5	PB6/TB3IN1/INT8	R15	P55/A5	V6	PG1/TC1IN
F7	AVSS	J16	P17/D15/AD15	N7	DVSS	R16	P61/A9	V7	PD1/TXD3
F8	AVSS	J17	P36/ R/ $\overline{\text{W}}$	N8	PD7/KEY8	R17	P21/A17/A1	V8	PD0/SCLK2/ $\overline{\text{CTS2}}$
F9	AVCC32	J18	P34/ $\overline{\text{BUSRQ}}$	N9	DVCC15 (DVCC22)	R18	P22/A18/A2	V9	PC6/TXD2
F10	DVCC34	K1	PN0/TXD6	N10	DVSS	T1	PA1/TA1OUT	V10	PH0/TCOUT0
F11	PI0/ $\overline{\text{ADTRG}}$	K2	PN1/RXD6	N11	RSTPUP	T2	PA4/TA5OUT	V11	PH2/TCOUT2
F12	DVSS	K3	PO5	N12	DVSS	T3	PA7/TABOUT	V12	PE0/TXD5
F14	NC (CAP2)	K4	PO6	N14	P26/A22/A6	T4	PF2/SCK	V13	PE3/KEY9
F15	P42/ $\overline{\text{CS2}}$	K5	NC (FVSS)	N15	P27/A23/A7	T5	PF6/ $\overline{\text{DACK3}}$	V14	PE7/KEYD
F16	P43/ $\overline{\text{CS3}}$	K6	DVSS	N16	P00/D0/AD0	T6	PG3/TC3IN	V15	P60/A8
F17	DVCC33	K13	NC (TEST0)	N17	P01/D1/AD1	T7	PD3/SCLK3/ $\overline{\text{CTS3}}$	V16	P64/A12
								V17	P67/A15

Note : Parentheses indicate the pin name on TMP1962F10AXBG with on-chip flash memory. (Except "DSU" and "JTAG". The same pin names are used for the on-chip mask ROM type and on-chip flash memory type.)

2.2 Pin Usage Information

Table 2.2 lists the input and output pins of the TMP1962, including alternate pin names and functions for multi-function pins.

Table 2.2 Pin Names and Functions (1/6)

Pin Name	Number of Pins	Type	Function
P00 - P07 D0 - D7 AD0 - D7	8	Input/Output Input/Output Input/Output	Port 0: Individually programmable as input or output Data (Lower): Bits 0 to 7 of the data bus (Separate Bus mode) Address/Data (Lower): Bits 0 to 7 of the address/data bus (Multiplexed Bus mode)
P10 - P17 D8 - D15 AD8 - AD15 A8 - A15	8	Input/Output Input/Output Input/Output Output	Port 1: Individually programmable as input or output Data (Upper): Bits 8 to 15 of the data bus (Separate Bus mode) Address/Data (Upper): Bits 8 to 15 of the address/data bus (Multiplexed Bus mode) Address: Bits 8 to 15 of the address bus (Multiplexed Bus mode)
P20 - P27 A16 - A23 A0 - A7 A16 - A23	8	Input/Output Output Output Output	Port 2: Individually programmable as input or output Address: Bits 16 to 23 of the address bus (Separate Bus mode) Address: Bits 0 to 7 of the address bus (Multiplexed Bus mode) Address: Bits 16 to 23 of the address bus (Multiplexed Bus mode)
P30 $\overline{RD}$	1	Output Output	Port 30: Output-only Read Strobe: Asserted during a read operation from an external memory device
P31 $\overline{WR}$	1	Output Output	Port 31: Output-only Write Strobe: Asserted during a write operation on D0 to D7
P32 $\overline{HWR}$	1	Input/Output Output	Port 32: Programmable as input or output (with internal pull-up resistor) Higher Write Strobe: Asserted during a write operation on D8 to D15
P33 $\overline{WAIT}$ $\overline{RDY}$	1	Input/Output Input Input	Port 33: Programmable as input or output (with internal pull-up resistor) Wait: Causes the CPU to suspend external bus activity Ready: Notifies the CPU that the bus is ready
P34 $\overline{BUSRQ}$	1	Input/Output Input	Port 34: Programmable as input or output (with internal pull-up resistor) Bus Request: Asserted by an external bus master to request bus mastership
P35 $\overline{BUSAK}$	1	Input/Output Output	Port 35: Programmable as input or output (with internal pull-up resistor) Bus Acknowledge: Indicates that the CPU has relinquished the bus in response to $\overline{BUSRQ}$
P36 R/$\overline{W}$	1	Input/Output Output	Port 36: Programmable as input or output (with internal pull-up resistor) Read/Write: Indicates the direction of data transfer on the bus: 1 = read or dummy cycle, 0 = write cycle
P37 ALE	1	Input/Output Output	Port 37: Programmable as input or output Address Latch Enable (enabled only when an external memory device is accessed)
P40 $\overline{CS0}$	1	Input/Output Output	Port 40: Programmable as input or output (with internal pull-up resistor) Chip Select 0: Asserted low to enable external devices at programmed addresses
P41 $\overline{CS1}$	1	Input/Output Output	Port 41: Programmable as input or output (with internal pull-up resistor) Chip Select 1: Asserted low to enable external devices at programmed addresses
P42 $\overline{CS2}$	1	Input/Output Output	Port 42: Programmable as input or output (with internal pull-up resistor) Chip Select 2: Asserted low to enable external devices at programmed addresses
P43 $\overline{CS3}$	1	Input/Output Output	Port 43: Programmable as input or output (with internal pull-up resistor) Chip Select 3: Asserted low to enable external devices at programmed addresses
P44 SCOUT	1	Input/Output Output	Port 44: Programmable as input or output System Clock Output: Drives out a clock signal at the same frequency as the CPU clock (high-speed or low-speed) or half the high-speed clock frequency
P50 - P57 A0 - A7	8	Input/Output Output	Port 5: Individually programmable as input or output Address: Bits 0 to 7 of the address bus (Separate Bus mode)
P60 - P67 A8 - A15	8	Input/Output Output	Port 6: Individually programmable as input or output Address: Bits 8 to 15 of the address bus (Separate Bus mode)

Table 2.2 Pin Names and Functions (2/6)

Pin Name	Number of Pins	Type	Function
P70 - P77 AN0 - AN7	8	Input Input	Port 7: Input-only Analog Input: Input to the on-chip A/D converter
P80 - P87 AN8 - AN15	8	Input Input	Port 8: Input-only Analog Input: Input to the on-chip A/D converter
P90 - P97 AN16 - AN23	8	Input Input	Port 9: Input-only Analog Input: Input to the on-chip A/D converter
PI0 $\overline{\text{ADTRG}}$	1	Input/Output Input	Port I0: Programmable as input or output A/D Trigger: Starts an A/D conversion Schmitt-triggered input
PI1 INT1	1	Input/Output Input	Port I1: Programmable as input or output Interrupt Request 1: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input
PI2 INT2	1	Input/Output Input	Port I2: Programmable as input or output Interrupt Request 2: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input
PI3 INT3	1	Input/Output Input	Port I3: Programmable as input or output Interrupt Request 3: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input
PI4 INT4	1	Input/Output Input	Port I4: Programmable as input or output Interrupt Request 4: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input
PI5 INT9	1	Input/Output Input	Port I5: Programmable as input or output Interrupt Request 9: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input
PI6 INTA	1	Input/Output Input	Port I6: Programmable as input or output Interrupt Request A: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input
PI7	1	Input/Output	Port I7: Programmable as input or output
PA0 TA0IN	1	Input/Output Input	Port A0: Programmable as input or output 8-Bit Timer 0 Input: Input to 8-bit timer 0
PA1 TA1OUT	1	Input/Output Output	Port A1: Programmable as input or output 8-Bit Timer 01 Output: Output from either 8-bit timer 0 or 1
PA2 TA2IN	1	Input/Output Input	Port A2: Programmable as input or output 8-Bit Timer 2 Input: Input to 8-bit timer 2
PA3 TA3OUT	1	Input/Output Output	Port A3: Programmable as input or output 8-Bit Timer 23 Output: Output from either 8-bit timer 2 or 3
PA4 TA5OUT	1	Input/Output Output	Port A4: Programmable as input or output 8-Bit Timer 45 Output: Output from either 8-bit timer 4 or 5
PA5 TA7OUT	1	Input/Output Output	Port A5: Programmable as input or output 8-Bit Timer 67 Output: Output from either 8-bit timer 6 or 7
PA6 TA9OUT	1	Input/Output Input	Port A6: Programmable as input or output 8-Bit Timer 89 Output: Output from either 8-bit timer 8 or 9
PA7 TABOUT	1	Input/Output Output	Port A7: Programmable as input or output 8-Bit Timer AB Output: Output from either 8-bit timer A or B
PB0 TB0OUT	1	Input/Output Output	Port B0: Programmable as input or output 16-Bit Timer 0 Output: Output from 16-bit timer 0
PB1 TB1OUT	1	Input/Output Output	Port B1: Programmable as input or output 16-Bit Timer 1 Output: Output from 16-bit timer 1

Table 2.2 Pin Names and Functions (3/6)

Pin Name	Number of Pins	Type	Function
PB2 TB2IN0 INT5	1	Input/Output Input Input	Port B2: Programmable as input or output 16-Bit Timer 2 Input 0: Count/capture trigger input to 16-bit timer 2 Interrupt Request 5: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB3 TB2IN1 INT6	1	Input/Output Input Input	Port B3: Programmable as input or output 16-Bit Timer 2 Input 1: Capture trigger input to 16-bit timer 2 Interrupt Request 6: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB4 TB2OUT	1	Output Output	Port B4: Programmable as input or output 16-Bit Timer 2 Output: Output from 16-bit timer 2
PB5 TB3IN0 INT7	1	Output Input Input	Port B5: Programmable as input or output 16-Bit Timer 3 Input 0: Count/capture trigger input to 16-bit timer 3 Interrupt Request 7: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB6 TB3IN1 INT8	1	Output Input Input	Port B6: Programmable as input or output 16-Bit Timer 3 Input 1: Capture trigger input to 16-bit timer 3 Interrupt Request 8: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PB7 TB3OUT	1	Output Output	Port B7: Programmable as input or output 16-Bit Timer 3 Output: Output from 16-bit timer 3
PC0 TXD0	1	Input/Output Output	Port C0: Programmable as input or output Serial Transmit Data 0: Programmable as a push-pull or open-drain output
PC1 RXD0	1	Input/Output Input	Port C1: Programmable as input or output Serial Receive Data 0
PC2 SCLK0 $\overline{\text{CTS0}}$	1	Input/Output Input Input	Port C2: Programmable as input or output Serial Clock Input/Output 0 Serial Clear-to-Send 0: Programmable as a push-pull or open-drain output
PC3 TXD1	1	Input/Output Output	Port C3: Programmable as input or output Serial Transmit Data 1: Programmable as a push-pull or open-drain output
PC4 RXD1	1	Input/Output Input	Port C4: Programmable as input or output Serial Receive Data 1
PC5 SCLK1 $\overline{\text{CTS1}}$	1	Input/Output Input Input	Port C5: Programmable as input or output Serial Clock Input/Output 1 Serial Clear-to-Send 1: Programmable as a push-pull or open-drain output
PC6 TXD2	1	Input/Output Output	Port C6: Programmable as input or output Serial Transmit Data 2: Programmable as a push-pull or open-drain output
PC7 RXD2	1	Input/Output Input	Port C7: Programmable as input or output Serial Receive Data 2
PD0 SCLK2 $\overline{\text{CTS2}}$	1	Input/Output Input Input	Port D0: Programmable as input or output Serial Clock Input/Output 2 Serial Clear-to-Send 2: Programmable as a push-pull or open-drain output
PD1 TXD3	1	Input/Output Output	Port D1: Programmable as input or output Serial Transmit Data 3: Programmable as a push-pull or open-drain output
PD2 RXD3	1	Input/Output Input	Port D2: Programmable as input or output Serial Receive Data 3
PD3 SCLK3 $\overline{\text{CTS3}}$	1	Input/Output Input Input	Port D3: Programmable as input or output Serial Clock Input/Output 3 Serial Clear-to-Send 3: Programmable as a push-pull or open-drain output
PD4 TXD4	1	Input/Output Output	Port D4: Programmable as input or output Serial Transmit Data 4: Programmable as a push-pull or open-drain output
PD5 RXD4	1	Input/Output Input	Port D5: Programmable as input or output Serial Receive Data 4
PD6 SCLK4 $\overline{\text{CTS4}}$	1	Input/Output Input Input	Port D6: Programmable as input or output Serial Clock Input/Output 4 Serial Clear-to-Send 4: Programmable as a push-pull or open-drain output
PD7 KEY8	1	Input/Output Input	Port D7: Programmable as input or output Key-Pressed Wake-up Input 8 (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input

Table 2.2 Pin Names and Functions (4/6)

Pin Name	Number of Pins	Type	Function
PE0 TXD5	1	Input/Output Output	Port E0: Programmable as input or output Serial Transmit Data 5: Programmable as a push-pull or open-drain output
PE1 RXD5	1	Input/Output Input	Port E1: Programmable as input or output Serial Receive Data 5
PE2 SCLK5 $\overline{CTS5}$	1	Input/Output Input Input	Port E2: Programmable as input or output Serial Clock Input/Output 5 Serial Clear-to-Send 5: Programmable as a push-pull or open-drain output
PE3 KEY9	1	Input/Output Input	Port E3: Programmable as input or output Key-Pressed Wake-up Input 9 (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input
PE4 KEYA	1	Input/Output Input	Port E4: Programmable as input or output Key-Pressed Wake-up Input A (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input
PE5 KEYB	1	Input/Output Input	Port E5: Programmable as input or output Key-Pressed Wake-up Input B (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input
PE6 KEYC	1	Input/Output Input	Port E6: Programmable as input or output Key-Pressed Wake-up Input C (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input
PE7 KEYD	1	Input/Output Input	Port E7: Programmable as input or output Key-Pressed Wake-up Input D (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input
PF0 SO SDA	1	Input/Output Output Input/Output	Port F0: Programmable as input or output Data transmit pin when the Serial Bus Interface is in SIO mode Data transmit/receive pin when the Serial Bus Interface is in I2C mode Programmable as a push-pull or open-drain output Schmitt-triggered input
PF1 SI SCL	1	Input/Output Input Input/Output	Port F1: Programmable as input or output Data receive pin when the Serial Bus Interface is in SIO mode Clock input/output pin when the Serial Bus Interface is in I2C mode Programmable as a push-pull or open-drain output Schmitt-triggered input
PF2 SCK	1	Input/Output Input/Output	Port F2: Programmable as input or output Clock input/output pin when the Serial Bus Interface is in SIO mode
PF3 $\overline{DREQ2}$	1	Input/Output Input	Port F3: Programmable as input or output DMA Request 2: Asserted by an external input/output device to request DMA transfer with DMAC2
PF4 $\overline{DACK2}$	1	Input/Output Output	Port F4: Programmable as input or output DMA Acknowledge 2: Indicates acknowledgement for a DMA transfer request made with DREQ2
PF5 $\overline{DREQ3}$	1	Input/Output Input	Port F5: Programmable as input or output DMA Request 3: Asserted by an external input/output device to request DMA transfer with DMAC3
PF6 $\overline{DACK3}$	1	Input/Output Output	Port F6: Programmable as input or output DMA Acknowledge 3: Indicates acknowledgement for a DMA transfer request made with DREQ3
PF7 TBTIN	1	Input/Output Input	Port F7: Programmable as input or output 32-Bit Time Base Timer Input: Count input to the 32-bit time base timer
PG0 - PG7 TC0IN - TC7IN	8	Input/Output Input	Port G: Individually programmable as input or output 32-Bit Timer Capture Trigger Input
PH0 - PH7 TCOUT0 - TCOUT7	8	Input/Output Output	Port H: Individually programmable as input or output 32-Bit Timer Compare Match Output
PJ0 INT0	1	Input/Output Input	Port J0: Programmable as input or output Interrupt Request 0: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive Schmitt-triggered input

Table 2.2 Pin Names and Functions (5/6)

Pin Name	Number of Pins	Type	Function
PJ1 BUSMD	1	Input/Output Input	Port J1: Programmable as input or output External Bus Mode: Multiplexed Bus mode is selected if this signal is sampled high on the rising edge of the reset signal. Separate Bus mode is selected if this signal is sampled low on the rising edge of the reset signal. The BUSMD pin should be pulled up or down upon a reset according to the bus mode to be used.
PJ2	1	Input/Output Input	Port J2: Programmable as input or output Single Boot Mode: Single Boot mode is selected if this signal is sampled low on the rising edge of the reset signal. Single Boot mode is used to rewrite the contents of on-chip flash memory. Normal operation is selected if the signal is sampled high on the rising edge of the reset signal. When performing normal operation, the BOOT pin should not be pulled down upon a reset.
PJ3	1	Input/Output Input	Port J3: Programmable as input or output Interleave Mode: Interleave mode is selected if this signal is sampled high on the rising edge of the reset signal. The INTLV pin should be pulled up when using Interleave mode. Otherwise, it should be pulled down.
PJ4 ENDIAN	1	Input/Output Input	Port J4: Programmable as input or output Endian Mode: Big-Endian mode is selected if this signal is sampled high on the rising edge of the reset signal. Little-Endian mode is selected if this signal is sampled low on the rising edge of the reset signal.
PK0 - PK7 KEY0 - KEY7	8	Input/Output Input	Port K: Individually programmable as input or output Key-Pressed Wake-up Input 0 to 7 (with internal pull-up resistor): Dynamic pull-up selectable Schmitt-triggered input
PL0 TA4IN	1	Input/Output Input	Port L0: Programmable as input or output 8-Bit Timer 4 Input: Input to 8-bit timer 4
PL1 TA6IN	1	Input/Output Input	Port L1: Programmable as input or output 8-Bit Timer 6 Input: Input to 8-bit timer 6
PL2 TA8IN	1	Input/Output Input	Port L2: Programmable as input or output 8-Bit Timer 8 Input: Input to 8-bit timer 8
PL3 TAAIN	1	Input/Output Input	Port L3: Programmable as input or output 8-Bit Timer A Input: Input to 8-bit timer A
PL4 TB0IN0	1	Input/Output Input	Port L4: Programmable as input or output 16-Bit Timer 0 Input 0: Count/capture trigger input to 16-bit timer 0
PL5 TB0IN1	1	Input/Output Input	Port L5: Programmable as input or output 16-Bit Timer 0 Input 1: Capture trigger input to 16-bit timer 0
PL6 TB1IN0	1	Input/Output Input	Port L6: Programmable as input or output 16-Bit Timer 1 Input 0: Count/capture trigger input to 16-bit timer 1
PL7 TB1IN1	1	Input/Output Input	Port L7: Programmable as input or output 16-Bit Timer 1 Input 1: Capture trigger input to 16-bit timer 1
PM0 - PM7	8	Input/Output	Port M: Individually programmable as input or output
PN0 TXD6	1	Input/Output Output	Port N0: Programmable as input or output Serial Transmit Data 6: Programmable as a push-pull or open-drain output
PN1 RXD6	1	Input/Output Input	Port N1: Programmable as input or output Serial Receive Data 6
PN2 SCLK6 $\overline{\text{CTS6}}$	1	Input/Output Input Input	Port N2: Programmable as input or output Serial Clock Input/Output 6 Serial Clear-to-Send 6: Programmable as a push-pull or open-drain output
PN3 - PN7	5	Input/Output	Port N3 to N7: Individually programmable as input or output
PO0 - PO7	8	Input/Output	Port O: Individually programmable as input or output
PP0 - PP7	8	Input/Output	Port P: Individually programmable as input or output

Table 2.2 Pin Names and Functions (6/6)

Pin Name	Number of Pins	Type	Function
$\overline{\text{NMI}}$	1	Input	Nonmaskable Interrupt Request: Causes an NMI interrupt on the falling edge Schmitt-triggered input
$\overline{\text{PLLOFF}}$	1	Input	This pin should be tied to logic 1 when the frequency multiplied clock from the PLL is used; otherwise, it should be tied to logic 0 (Schmitt-triggered input).
RSTPUP	1	Input	Pull-up resistors for Ports 3 and 4 are enabled if this signal is sampled high upon a reset; otherwise, the pull-up resistors are disabled. Schmitt-triggered input
$\overline{\text{RESET}}$	1	Input	Reset (with internal pull-up resistor): Initializes the whole TMP1962. Schmitt-triggered input
X1/X2	2	Input/Output	Connection pins for a high-speed resonator
$\overline{\text{DRESET}}$	1	Input	Debug Reset: Signal for a DSU-ICE (Schmitt-triggered input with internal pull-up resistor)
DCLK	1	Output	Debug Clock: Signal for a DSU-ICE
$\overline{\text{DBGE}}$	1	Input	Debug Enable: Signal for a DSU-ICE (Schmitt-triggered input with internal pull-up resistor)
PCST3 - 0	4	Output	PC Trace Status: Signals for a DSU-ICE
SDI/ $\overline{\text{DINT}}$	1	Input	Serial Data Input/Debug Interrupt: Signal for a DSU-ICE (Schmitt-triggered input with internal pull-up resistor)
SDAO/TPC	1	Output	Serial Data Address Output/Target PC: Signal for a DSU-ICE
TCK	1	Input	Test Clock Input: JTAG test signal (Schmitt-triggered input with internal pull-up resistor)
TMS	1	Input	Test Mode Select Input: JTAG test signal (Schmitt-triggered input with internal pull-up resistor)
TDI	1	Input	Test Data Input: JTAG test signal (Schmitt-triggered input with internal pull-up resistor)
TDO	1	Output	Test Data Output: JTAG test signal
$\overline{\text{TRST}}$	1	Input	Test Reset Input: JTAG test signal (Schmitt-triggered input with internal pull-down resistor)
BW0 - 1	2	Input	Both BW0 and BW1 should be tied to logic 1 (Schmitt-triggered input).
VREFH	1	Input	Input pin for high reference voltage for the A/D Converter. This pin should be connected to the AVCC pin when the A/D Converter is not used.
VREFL	1	Input	Input pin for low reference voltage for the A/D Converter. This pin should be connected to the AVSS pin when the A/D Converter is not used.
AVCC31 - 32	2	—	Power supply pins for the A/D Converter. These pins should always be connected to power supply even when the A/D Converter is not used.
AVSS	3	—	Ground pin for the A/D Converter. This pin should always be connected to ground even when the A/D Converter is not used.
TEST5	1	Input	Test pin: This pin should be tied to ground.
SYSRDY	1	Output	Flash memory access enable
CVCC15	1	—	1.5-V power supply pin for the oscillator
CVSS	1	—	Ground pin (0 V) for the oscillator
DVCC15	1	—	1.5-V power supply pin
DVCC2	5	—	2-V power supply pin
DVCC31 - 34	9	—	3-V power supply pins
DVSS	9	—	Ground pin (0 V)

Note: PJ1, PJ2, PJ3 and PJ4 should be held at the prescribed logic states for one system clock cycle before and after the rising edge of RESET, with the RESET signal being stable in either logic state.

Table 2.3 shows the correspondence between pins and power supply pins.

Table 2.3 Pins and Corresponding Power Supply Pins

Pin	Power Supply		Pin	Power Supply	
	Mask Type	Flash Type		Mask Type	Flash Type
P0	DVCC33	DVCC33	PP	DVCC31	DVCC31
P1	DVCC33	DVCC33	X1	CVCC15	CVCC2
P2	DVCC33	DVCC33	X2	CVCC15	CVCC2
P3	DVCC33	DVCC33	$\overline{\text{RESET}}$	DVCC2	DVCC21
P4	DVCC33	DVCC33	$\overline{\text{NMI}}$	DVCC2	DVCC21
P5	DVCC33	DVCC33	$\overline{\text{PLLOFF}}$	DVCC2	DVCC21
P6	DVCC33	DVCC33	$\overline{\text{DRESET}}$	DVCC2	DVCC21
P7	AVCC32	AVCC32	DCLK	DVCC2	DVCC21
P8	AVCC32	AVCC32	$\overline{\text{DBGE}}$	DVCC2	DVCC21
P9	AVCC31	AVCC31	PCST3 - 0	DVCC2	DVCC21
PA	DVCC32	DVCC32	SDI/ $\overline{\text{DINT}}$	DVCC2	DVCC21
PB	DVCC32	DVCC32	SDAO/TPC	DVCC2	DVCC21
PC	DVCC32	DVCC32	TCK	DVCC34	DVCC34
PD	DVCC32	DVCC32	TMS	DVCC34	DVCC34
PE	DVCC32	DVCC32	TDI	DVCC34	DVCC34
PF	DVCC32	DVCC32	TDO	DVCC34	DVCC34
PG	DVCC32	DVCC32	$\overline{\text{TRST}}$	DVCC34	DVCC34
PH	DVCC32	DVCC32	BW1 - 0	DVCC2	DVCC21
PI	DVCC34	DVCC34	RSTPUP	DVCC32	DVCC32
PJ	DVCC2	DVCC21	G3	DVCC2	FVCC2
PK	DVCC34	DVCC34	G18	NC	FVCC2
PL	DVCC34	DVCC34	K5	NC	FVSS
PM	DVCC34	DVCC34	L1	NC	FVCC
PN	DVCC31	DVCC31	L15	NC	FVCC2
PO	DVCC31	DVCC31	M15	NC	FVSS

Table 2.4 shows the supply voltage for power supply pins.

Table 2.4 Supply Voltage for Power Supply Pins

Power Supply Pin	Supply Voltage	Applied for
DVCC15	1.35 V - 1.65 V	Mask Type
CVCC15	1.35 V - 1.65 V	
DVCC2	2.3 V - 3.3 V	
DVCC21	2.2 V - 2.7 V	Flash Type
DVCC22	2.2 V - 2.7 V	
CVCC2	2.2 V - 2.7 V	
FVCC2	2.2 V - 2.7 V	
FVCC3	2.9 V - 3.6 V	
DVCC31 - 34	1.65 V - 3.3 V	Mask/Flash Type
AVCC31 - 32	2.7 V - 3.3 V	

Note 1: AVCC32 ≤ AVCC31

- **When P7 to P9 are used as A/D converter inputs:**
  2.7 V < AVCC3*
- **When P9 (powered by AVCC31) is used as an A/D converter input while P7 and P8 (powered by AVCC32) are used as ports:**
  2.7 V ≤ AVCC31 ≤ 3.3 V
  1.65 V ≤ AVCC32 ≤ AVCC31
- **When P7 (powered by AVCC32) is used as an A/D converter input while P8 (powered by AVCC32) and P9 (powered by AVCC31) are used as ports:**
  2.7 V ≤ AVCC32 ≤ AVCC31 ≤ 3.3 V

Note 2: With power supplies for CPU and internal logic (mask type: DVCC15/DVCC2/CVCC15, and flash type: DVCC21/DVCC22/CVCC2/FVCC2/FVCC3) being applied, power supplies for other I/O ports can be interrupted on TMP1962. However, when AVCC31 for analog power supply is interrupted, overlap current is generated on the TMP1962F10AXBG with on-chip flash memory during the transition to be stable in 0 V. Overlap current can be suppressed by AD conversion of the conversion result 0 V before interrupting AVCC31 power supply, but please suppress it on devices.

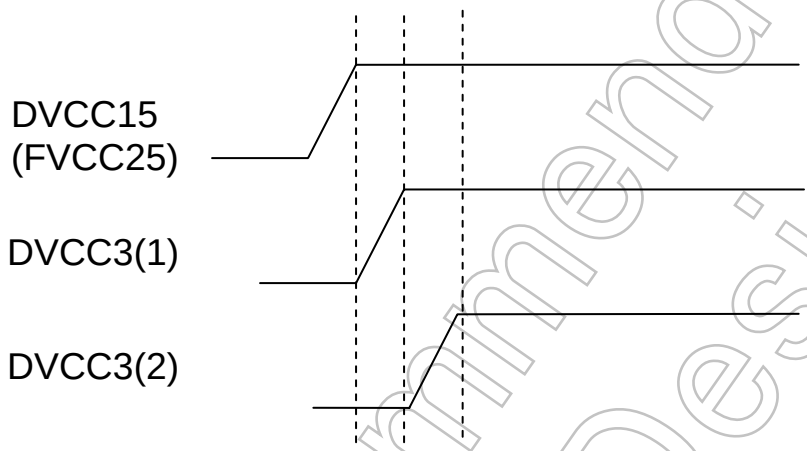
3. Core Processor

The TMP1962 contains a high-performance 32-bit core processor called the TX19. For a detailed description of the core processor, refer to the TX19 Family Architecture manual.

Functions unique to the TMP1962, which are not covered in the architecture manual, are described below.

Note: All references to register addresses in the following descriptions assume that the TMP1962 is operating in Big-Endian mode.

We recommend the power on sequence of this device ,firstly turn on a power to core (Flash: FVCC=2.5V, Mask: DVCC15=1.5V) before other power on in this device.



DVCC3(1)、(2)･･･I/O Power，ADC Power

3.1 Reset Operation

To reset the TMP1962, RESET must be asserted for at least 12 system clock periods after the power supply voltage and the internal high-frequency oscillator have stabilized. This time is typically 2.37 μs at 40.5 MHz when the on-chip PLL is utilized.

After a reset, either the PLL-multiplied clock or an external clock is selected, depending on the logic state of the PLLOFF pin. By default, the selected clock is geared down to 1/8 for internal operation.

The following occurs as a result of a reset:

- The System Control Coprocessor (CP0) registers within the TX19 core processor are initialized. For details, refer to the architecture manual.
- The Reset exception is taken. Program control is transferred to the exception handler at a predefined address. This predefined location is called an exception vector, which directly indicates the start of the actual exception handler routine. The Reset exception is always vectored to virtual address 0xBFC0_0000 (which is the same as for the Nonmaskable Interrupt exception).
- All on-chip I/O peripheral registers are initialized.
- All port pins, including those multiplexed with on-chip peripheral functions, are configured as either general-purpose inputs or general-purpose outputs.

Note 1: The TMP1962 must be powered up with $\overline{\text{RESET}}$ asserted. The reset state should not be terminated until after the power supply voltage stabilizes within the valid operating range.

Note 2: A wait time of at least 500 µs is required between the time when the power supply voltages for the stabilize and the time when the reset state is terminated.

When you use the FLASH product.

When the TMP1962 is powered up, terminating the reset state causes the core processor to wait 30 µs before starting operation. This time is required to initialize the on-chip flash memory controller. The TMP1962 outputs the SYSRDY signal to notify an external device that the core processor has started. Once the core processor exits from the reset state, the SYSRDY signal is driven from low to high. Subsequent non-power-up reset operations are controlled with bit 7 (FLRMSK) of the Flash Control/Status Register (FLCS) in the flash memory controller. If the FLRMSK bit is cleared to 0 (default), the flash memory controller is always initialized upon a reset. Setting the FLRMSK bit to 1 prevents the flash memory controller from being initialized upon a reset (data in on-chip flash memory can still be read correctly). In the latter case, the 30 µs (T.B.D.) wait time, described above, is not required after a reset so that the core processor starts immediately, driving SYSRDY high. The setting of the FLRMSK bit is held until the TMP1962 is powered off. Usually, FLRMSK should be set to 1 as part of initialization after a reset.

FLCS (0xFFFF_E520)

	7	6	5	4	3	2	1	0
Bit symbol	FLRMSK					RDY/BSY		
Read/Write	W					R		
Reset value	0					1	0	
Function	Flash reset mask 0: Reset the flash memory controller. 1: Do not reset the flash memory controller.					Ready/Busy 0: Automatic operation in progress 1: Automatic operation completed	Must be written as 0.	

(The FLCS is a 32-bit register.)

Figure 3.1 Flash Control/Status Register

The FLCS register doesn't exist in the mask rom type an irregular value is read when leading.

Memory Map

Figure 0.1 shows memory assignment for the TMP1962.

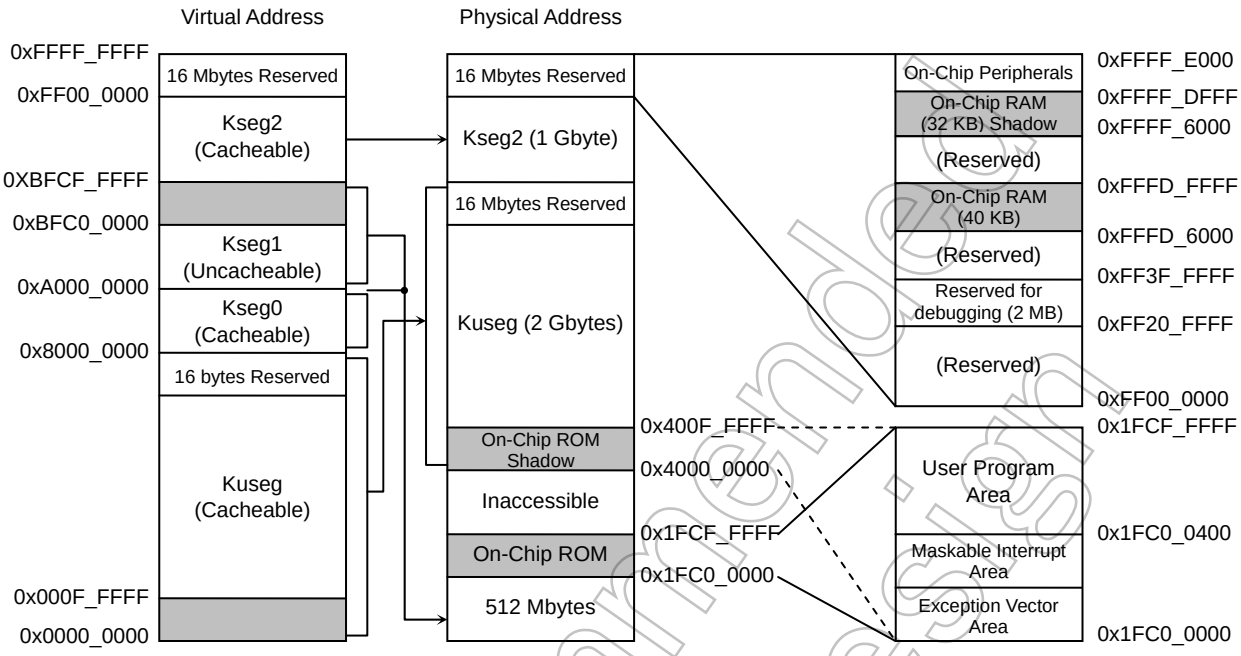


Figure 0.1 Memory Map

Note 1: The on-chip 1-Mbyte ROM is mapped to the addresses from 0x1FC0_0000 through 0x1FCF_FFFF and the on-chip 40-Kbyte RAM is mapped to the addresses from 0xFFFD_6000 through 0xFFFD_FFFF.

Note 2: The on-chip ROM is located in a linear address space beginning at physical address 0x1FC0_0000. All types of exceptions are vectored to the on-chip ROM when the BEV bit of the System Control Coprocessor's Status register is set to the default value of 1. (When BEV = 0, not all exception vectors reside in contiguous locations.) When external memory is used, the BEV bit can be cleared to 0.

However, using the 32K-byte virtual address range beginning at 0x0000_0000 helps to improve code efficiency, as shown below. The shaded area starting at physical address 0x4000_0000 has a size equal to the on-chip ROM size. References to this range (mapped from the virtual address space starting at 0x0000_0000) are rerouted to the on-chip ROM.

Examples: 32-bit ISA

- **Accessing the 0x0000_0000±32-KB region**

```
ADDIU   r2, r0, 7          ; r 2 ← (0x0000_0007)
SW      r2, Io (_t) (r0)   ; 0x0000_xxxx ← (r2)  ← Accessed with a single instruction
```

- **Accessing other regions**

```
LUI     r3, hi (_f)        ; ← Upper 16 bits of address are loaded into r3.
ADDIU   r2, r0, 8          ; r 2← (0x0000_0008)
SW      r2, Io (_f) (r3)   ; Lower 16-bits of address must be added to upper 16 bits.
```

Note 3: In the TMP1962, the on-chip 40-Kbyte RAM is mapped to the addresses from 0xFFFD_6000 through 0xFFFD_FFFF. This area is shadowed to a 32-Kbyte address range from 0xFFFF_6000 through 0xFFFF_DFFF. References to this range are rerouted to the on-chip RAM.

Note 4: The TMP1962 has access to only 16 Mbytes of external physical address space. The 16-Mbyte physical memory can be located anywhere within the CPU's 3.5-Gbyte physical address space through use of programmable chip select signals. However, any address references to the on-chip memory, on-chip peripheral or reserved regions override external memory access.

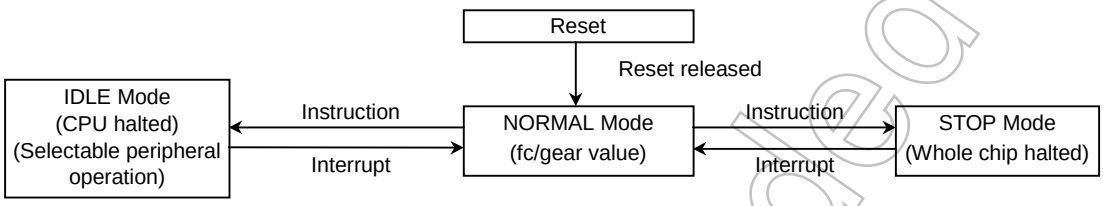
Note 5: No instruction should be placed in the last four words of the physical address space.

- **If only on-chip ROM is used: 0x1FCF_FFF0 through 0x1FCF_FFFF**
- **If ROM is added off-chip: Last four words of the memory installed in the end-user system**

5. Clock/Standby Control

The TMP1962 has the stand-by mode in which the core processor stops to reduce power consumption.

Figure 5.1 shows the transition between clocking modes.



Transition between Clock Modes

Figure 5.1 Standby Mode Flow Diagram

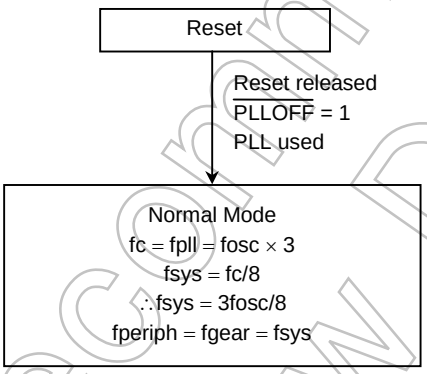


Figure 5.2 Default Clock Frequencies in Normal Mode

fosc:	**Clock frequency supplied via the X1 and X2 pins**
fpll:	**PLL multiplied clock frequency (x3)**
fc:	**Clock frequency selected by the $\overline{\text{PLLOFF}}$ pin**
fgear:	**Clock frequency selected by the GEAR[1:0] bits in the clock generator's system control register (SYSCR1)**
fsys:	**System clock frequency** **The CPU, ROM, RAM, DMAC and INTC operate based on this system clock. On-chip peripherals operate on fsys/2.**
fperiph:	**Clock frequency selected by the FPSEL bit in the SYSCR1 (clock source for the prescalers inside on-chip peripherals)**

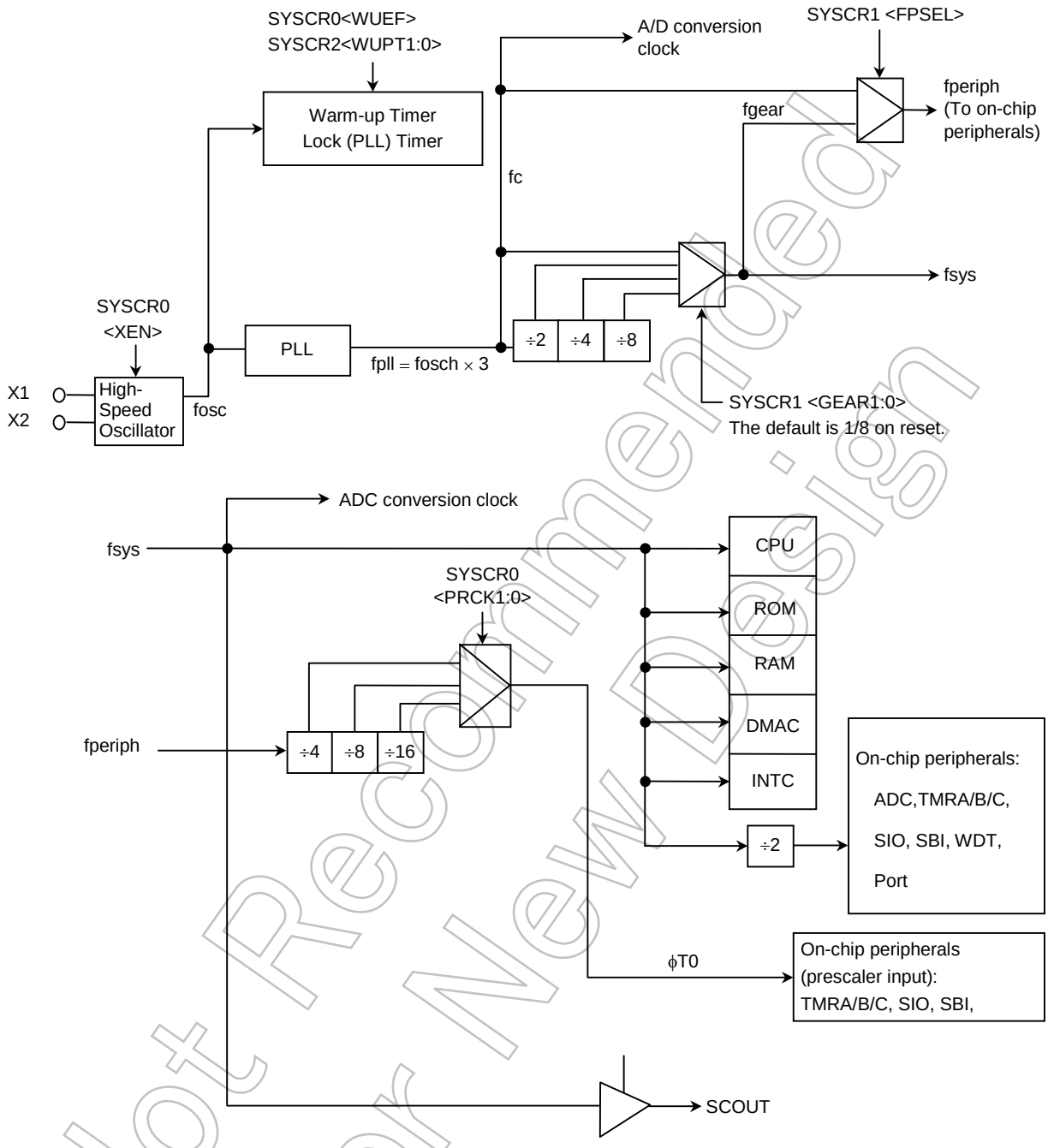
5.1 Clock Generation

5.1.1 Main System Clock

- A crystal can be connected between X1 and X2, or X1 can be externally driven with a clock.
- The on-chip PLL can be enabled or disabled (bypassed) during reset by using the $\overline{\text{PLLOFF}}$ pin. When the PLL is enabled, the input clock frequency is multiplied by three.
- The clock gear can be programmed to divide the clock by 2, 4 or 8. (The default is 1/8 on reset.)
- Input clock frequency

	Input Frequency Range	Maximum Operating Frequency	Minimum Operating Frequency
PLL ON (for both crystal and external clock)	10 to 13.5 (MHz)	40.5 MHz	3.75 MHz

5.1.2 Clock Source Block Diagram



Note 1: When the clock gear is used to reduce the system clock frequency (fsys), the prescalers within on-chip peripherals must be programmed so that the prescaler output (ϕTn) satisfies the following relationship:

ϕTn < fsys/2

Note 2: The prescaler clock source (ϕT0) must not be changed while any of the peripherals to which it is supplied are running.

Figure 5.3 Dual Clock and Standby Block Diagram

5.2 Clock Generator (CG) Registers

5.2.1 System Clock Control Registers

		31	30	29	28	27	26	25	24
SYSCR3	Bit symbol		SCOSEL1	SCOSEL0	ALESEL			LUPFG	LUPTM
(0xFFFF_EE00)	Read/Write		R/W			R			R/W
	Reset value	0	0	1	1	1	1	0	0
	Function		SCOUT output select 00: fs 01: fsys/2 10: fsys 11: (reserved)		ALE output width select 0: fsys × 0.5 1: fsys × 1.5	Must be written as 0 (flash type)	Must be written as 0 (flash type)	PLL lock 0: Locked 1: Unlocked	PLL lock time select 0: 2^{16}/input frequency 1: 2^{12}/input frequency
		23	22	21	20	19	18	17	16
SYSCR2	Bit symbol	DRVOSCH		WUPT1	WUPT0	STBY1	STBY0		DRVE
(0xFFFF_EE01)	Read/Write	R/W		R/W					R/W
	Reset value	0	0	1	0	1	1	0	0
	Function	High-speed oscillator drive capability 0: High 1: Low	Must be written as 0 (flash type)	Oscillator warm-up time (Note 2) 00: No warm-up 01: 2^{8}/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		Standby mode select (Note 1) 00: Reserved 01: STOP mode 10: Reserved 11: IDLE mode			0: Pins are not driven in STOP mode. 1: Pins are driven in STOP mode. (See Table 3.3.9.)
		15	14	13	12	11	10	9	8
SYSCR1	Bit symbol				FPSEL			GEAR1	GEAR0
(0xFFFF_EE02)	Read/Write				R/W			R/W	
	Reset value	0	0	0	0	0	0	1	1
	Function			Must be written as 0 (flash type)	fperiph select 0: fgear 1: fc		Must be written as 0 (flash type)	High-speed clock (fc) gear select 00: fc 01: fc/2 10: fc/4 11: fc/8	
		7	6	5	4	3	2	1	0
SYSCR0	Bit symbol							PRCK1	PRCK0
(0xFFFF_EE03)	Read/Write							R/W	
	Reset value	1	0	1	0	0	0	0	0
	Function	Must be written as 1 (flash type)	Must be written as 0 (flash type)	Must be written as 1 (flash type)	Must be written as 0 (flash type)		Must be written as 0	Prescaler clock select 00: fperiph/16 01: fperiph/8 10: fperiph/4 11: (reserved)	

Note 1: The Config register in the CP0 has the Doze and Halt bits. Setting the Halt bit puts the TMP1962 in one of the standby modes, as specified by the STBY[1:0] bits in the SYSCR2. Setting the Doze bit puts the TMP1962 in IDLE mode, irrespective of the settings of the STBY[1:0] bits.

Note 2: The WUPT[1:0] bits in the SYSCR2 must not be changed during the oscillator warm-up period. The LUPTM bit in the SYSCR3 must not be changed during the PLL lock period.

Note 3: The oscillator warm-up period (WUP) timer is also used as the PLL lock timer.

Note 4: When the PLL is used, the WUPT[1:0] bits in the SYSCR2 must not be set to 00 (no warm-up).

Note 5: When the PLL is not used, the LUPFG bit in the SYSCR3 is always read as 0.

5.2.2 STOP Wake-up Interrupt Control Registers (INTCG Registers)

IMCGA0 (0xFFFF_EE10)

	31	30	29	28	27	26	25	24
Bit symbol			EMCG31	EMCG30				INT3EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT3 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT3 enable 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit symbol			EMCG21	EMCG20				INT2EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT2 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT2 enable 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit symbol			EMCG11	EMCG10				INT1EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT1 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT1 enable 0: Disable 1: Enable
	7	6	5	4	3	2	1	0
Bit symbol			EMCG01	EMCG00				INT0EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT0 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT0 enable 0: Disable 1: Enable

IMCGB0 (0xFFFF_EE14)

	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write			R/W					R/W
Reset value			1	1				0
Function			.					Must be written as 0.
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write			R/W					R/W
Reset value			1	0				0
Function								Must be written as 0.
	15	14	13	12	11	10	9	8
Bit symbol			EMCG51	EMCG50				KWUPEN
Read/Write			R/W					R/W
Reset value			0	1				0
Function			Wake-up KWUP sensitivity 00: Setting prohibited 01: High level 10: Setting prohibited 11: Setting prohibited These bits must be set to 01.					KWUP enable 0: Disable 1: Enable
	7	6	5	4	3	2	1	0
Bit symbol			EMCG41	EMCG40				INT4EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT4 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT4 enable 0: Disable 1: Enable

IMCGC0 (0xFFFF_EE18)

	31	30	29	28	27	26	25	24
Bit symbol			EMCGB1	EMCGB0				INT6EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT6 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT6 enable 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit symbol			EMCGA1	EMCGA0				INT5EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT5 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT5 enable 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write			R/W					R/W
Reset value			1	1				0
Function								Must be written as 0.
	7	6	5	4	3	2	1	0
Bit symbol								
Read/Write			R/W					R/W
Reset value			1	1				0
Function								Must be written as 0.

IMCGD0 (0xFFFF_EE1C)

	31	30	29	28	27	26	25	24
Bit symbol			EMCGF1	EMCGF0				INTAEN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INTA sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INTA enable 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit symbol			EMCGE1	EMCGE0				INT9EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT9 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT9 enable 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit symbol			EMCGD1	EMCGD0				INT8EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT8 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT8 enable 0: Disable 1: Enable
	7	6	5	4	3	2	1	0
Bit symbol			EMCGC1	EMCGC0				INT7EN
Read/Write			R/W					R/W
Reset value			1	0				0
Function			Wake-up INT7 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT7 enable 0: Disable 1: Enable

Note 1: The edge/level sensitivity must be defined for an interrupt pin which is enabled as wake-up signaling to exit STOP mode.

Note 2: Interrupt programming must follow these steps:

1. Configure the pin as an interrupt input, if the pin is multiplexed with a general-purpose port.
2. Set the active state for the interrupt during initialization.
3. Clear any interrupt request.
4. Enable the interrupt.

Note 3: The above steps must be performed with the relevant interrupt pin disabled.

Note 4: The TMP1962 has 15 interrupt sources which can be used for wake-up signaling to exit STOP mode: INT0 to INTA, INTRTC, INTTB2, INTTB3 and KWUP0 to KWUPD. When one of INT0 to INTA is used for STOP wake-up signaling, it must be enabled as a wake-up interrupt source in the CG block and its interrupt sensitivity must be specified in the CG block. When one of KWUP0 to KWUPD is used for STOP wake-up signaling, it must be enabled as a wake-up interrupt source in the CG block and its interrupt sensitivity must be specified in the KWUPSTx. In the INTC block, the sensitivity for all of the above 15 interrupt sources must be set to the high level.

Example: Enabling the INT0 interrupt

IMCGA0<EMCG01:00> = "10"
IMCGA0<INT0EN> = "1" } CG block (Set the INT0 sensitivity to the falling edge.)

IMC0L<EIM11:10> = "01"
IMC0L<IL12:10> = "101" } INTC block (Set the interrupt sensitivity to the high level, and the interrupt priority level to 5.)

All interrupt sources other than those used for STOP wake-up signaling are controlled by the INTC block.

Note 5: When one of INT0 to INTA is used as a normal interrupt source, its settings in the CG block are not necessary while its interrupt sensitivity must be specified in the INTC block. When one of KWUP0 to KWUPD is used as a normal interrupt source, its settings in the CG block are not necessary while its interrupt sensitivity must be specified in the KWUPSTx (in the INTC block, the sensitivity for KWUP0 to KWUPD must be set to the high level). For INTRTC, however, both CG and INTC settings are required even when it is used as a normal interrupt source.
All interrupt sources other than those used for STOP wake-up signaling are controlled by the INTC block.

5.2.3 Exit Stop Mode Interrupt Clear Request Register (EICRCG Register)

EICRCG (0xFFFF_EE20)

	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write								
Reset value								
Function								
	23	22	21	20				
Bit symbol								
Read/Write								
Reset value								
Function								
	15	14	13	12				
Bit symbol								
Read/Write								
Reset value								
Function								
	7	6	5	4	3	2	1	0
Bit symbol					ICRCG3	ICRCG2	ICRCG1	ICRCG0
Read/Write					W			
Reset value					—	—	—	—
Function					Clear interrupt request 0000: INT0 0101: KWUP 1010: INT5 0001: INT1 0110: reserved 1011: INT6 0010: INT2 0111: reserved 1100: INT7 0011: INT3 1000: reserved 1101: INT8 0100: INT4 1001: reserved 1110: INT9 1111: INTA			

Note 6: Interrupt requests for the above 15 interrupt sources which can be used for wake-up signaling to exit STOP mode are cleared as follows:

1. **The clearing of KWUP interrupt sources is controlled through the KWUPCLR register.**
2. **Clearing the INT0 to INTA, INTTB2, INTTB3 and INTRTC interrupt requests requires two register settings: first, the EICRCG register in the CG block, and then the INTCLR register in the INTC block.**
3. **The clearing of other interrupt sources is controlled through the INTCLR register alone.**

5.3 System Clock Control Section

A system reset initializes the SYSCR0.XEN bit to 1, the SYSCR0.XTEN bit to 0 and the SYSCR1.GEAR[1:0] bits to 11, putting the TMP1962 in Single-Clock mode. If the on-chip PLL is enabled, the PLL reference clock is always multiplied by three. By default, the system clock frequency (fsys) is geared down to fc/8, where fc = fosc x 3 (fosc is the oscillator frequency). For example, if a 13.5-MHz crystal is connected between the X1 and X2 pins, the fsys clock operates at 5.0625 MHz (13.5 x 3 x 1/8).

Note: The system clock frequency must be initialized to 3.75 MHz or higher.

5.3.1 Oscillation Stabilization Time

When a crystal is connected between the X1 and X2 pins and/or XT1 and XT2 pins, the integrated warm-up period timer is used to assure oscillation stability. The warm-up period can be selected through the [WUPT1:0] bits of the SYSCR2 to suit the crystal used.

Table 5.1 shows the warm-up periods required when the clocking is switched between NORMAL and SLOW modes.

Note 1: No warm-up is necessary when the TMP1962 is driven by an external oscillator clock which is already stable.
Note 2: Because the warm-up period timer is clocked by the oscillator clock, any frequency fluctuations will lead to small timer errors. Table 5.1 should be considered as approximate values.

Table 5.1 Warm-up Periods

Warm-up Period Select SYSCR2.WUPT[1:0]	High-Speed Clock (fosc)
01 (2^8/oscillation frequency)	19.0 (μs)
10 (2^{14}/oscillation frequency)	1.214 (ms)
11 (2^{16}/oscillation frequency)	4.855 (ms)

Assumption:
fosc = 13.5 MHz

5.3.2 System Clock Output

The fsys, fsys/2 or fs clock can be driven out from the P44/SCOUT pin. The P44/SCOUT pin is configured as SCOUT (system clock output) by programming the Port 4 registers as follows: P4CR.P44C = 1 and P4FC.P44F = 1. The output clock is selected through the SYSCR3.SCOSEL[1:0] bits.

Table 5.2 shows the pin states in each clocking mode when the P44/SCOUT pin is configured as SCOUT.

Table 5.2 SCOUT Output States

Mode / SCOUT Select	NORMAL, SLOW	Standby Modes	
		IDLE	STOP
<SCOSEL1:0> = "01"	The fsys/2 clock is driven out.	Held at either 1 or 0.	
<SCOSEL1:0> = "10"	The fsys clock is driven out.		

Note: The phase difference between the system clock output signal (SCOUT) and the internal clock signal cannot be guaranteed.

5.3.3 Reducing the Oscillator Clock Drive Capability

When a crystal is connected between the X1 and X2 pins and/or between XT1 and XT2 pins, oscillator noise and power consumption can be reduced through the programming of the SYSCR2.

Setting the SYSCR2.DRVOSCH bit reduces the drive capability of the high-speed oscillator. Setting the SYSCR2.DRVOSCL bit reduces the drive capability of the low-speed oscillator clock.

A reset clears both the DRVOSCH and DRVOSCL bits to 0, providing a high drive capability at power-up. Both the high-speed and low-speed oscillator clocks must have a high drive capability (i.e., DRVOSCH = 0, DRVOSCL = 0) when clocking modes are changed.

- Drive capability of the high-speed oscillator

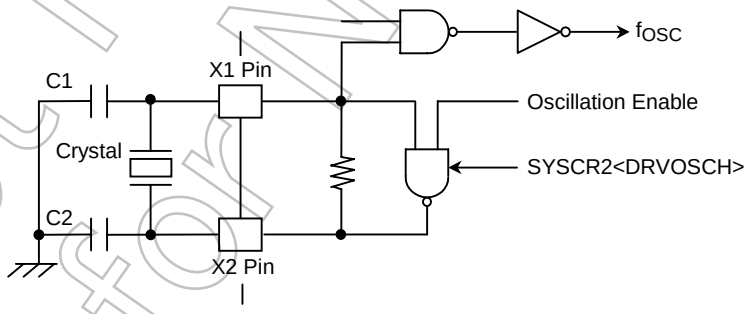


Figure 5.4 Oscillator Clock Drive Capabilities

5.4 Prescaler Clock Control Section

The TMRA01 to TMRAAB, TMRB0 to TMRB3, TMRC, SIO0 to SIO6 and SBI have a clock prescaler. The prescaler clock source (ϕT0) can be selected from fperiph/16, fperiph/8 and fperiph/4 through the PRCK[1:0] bits of the SYSCR0. fperiph can be selected from either fgear or fc through the FPSEL bit of the SYSCR1. The default reset values select fgear as fperiph, and fperiph/16 as ϕT0.

5.5 Clock Frequency Multiplication Section (PLL)

The on-chip PLL multiplies the frequency of the high-speed oscillator clock (fosc) by three to generate the fpll clock. To use the PLL, the PLLOFF pin must be high when RESET is released.

Being an analog circuit, the PLL requires a certain duration of time (called lock time) to stabilize, like an oscillator. The oscillator warm-up period (WUP) timer is also used as the PLL lock timer. The LUPTM bit in the SYSCR3 must be programmed so that the following relationship is satisfied:

PLL lock time ≥ Oscillator warm-up time

At reset, the default lock-up time is 2^{16}/input frequency.

Setting the WUP timer control bit (SYSCR0.WUEF) starts the PLL lock timer. The SYSCR3.LUPTM bit remains set while the PLL is out of lock, and is cleared when the PLL locks.

In real-time applications whose software execution time is critical, once the PLL has gone out of lock in a standby mode, software must determine before resuming operation whether the PLL has locked (after the oscillator warm-up time has expired) in order to assure clock stability.

Note 1: If the $\overline{\text{PLLOFF}}$ pin is low when RESET is released, the PLL will be disabled and the oscillator clock will be driven with no frequency multiplication.

Note 2: The following must be noted when changing the clock gear value.
The clock gear can be changed by the programming of the GEAR[1:0] bits of the SYSCR1. It takes a few clock cycles for a gear change to take effect. Therefore, one or more instructions following the instruction that changed the clock gear value may be executed using the old clock gear value. If subsequent instructions need to be executed with a new clock gear value, a dummy instruction (one that executes a write cycle) should be inserted after the instruction that modifies the clock gear value.
When the clock gear is used, the prescalers within on-chip peripherals must be programmed so that the prescaler output (ϕTn) satisfies the following relationship:
ϕTn < fsys/2
The clock gear must not be changed while a timer/counter or other peripherals are operating.

5.6 Standby Control Section

The TMP1962 provides support for several levels of power reduction. While in NORMAL mode, setting the Halt bit of the Config register within the TX19 core processor causes the TMP1962 to enter one of the standby modes — IDLE, STOP — as specified by the SYSCR2.STBY[1:0] bits. Setting the Doze bit of the Config register causes the TMP1962 to enter IDLE (Doze) mode, irrespective of the setting of SYSCR2.STBY[1:0].

The characteristics of the IDLE, STOP modes are as follows:

IDLE: The CPU stops.

On-chip peripherals can be selectively enabled and disabled through use of a register bit in a given peripheral, as shown in Table 5.3. If an on-chip peripheral has its register bit cleared to disable operation in IDLE mode, it stops when the TMP1962 enters IDLE mode, holding the state in which it is placed when it stops.

Table 5.3 IDLE Mode Register Settings

Peripheral	IDLE Mode Bit
TMRA01 to AB	TAxxRUN<I2TAxx>
TMRB0 to 3	TBxRUN<I2TBx>
TBT	TBTRUN<I2TBT>
SIO0 to 6	SCxMOD1<I2Sx>
SBI	SBIBR1<I2SBIx>
A/D Converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

Note 1: In Halt mode (i.e., a standby mode entered by setting the Halt bit in the Config register), the TMP1962 freezes the TX19 core processor, preserving the pipeline state. In Halt mode, the TMP1962 ignores any external bus requests; so it continues to assume bus mastership.

Note 2: In Doze mode (i.e., a standby mode entered by setting the Doze bit in the Config register), the TMP1962 freezes the TX19 core processor, preserving the pipeline state. In Doze mode, the TMP1962 recognizes external bus requests.

STOP: The whole TMP1962 stops.

5.6.1 TMP1962 Operation in NORMAL and Standby Modes

Table 5.4 TMP1962 Operation in NORMAL and Standby Modes

Operating Mode	Operating States
NORMAL	The TX19 core processor and on-chip peripherals operate at frequencies specified in the CG block.
IDLE (Halt)	The processor and DMAC operations stop; other on-chip peripherals can be selectively disabled.
IDLE (Doze)	Processor operation stops; the DMAC is operational; other on-chip peripherals can be selectively disabled.
STOP	All processor and peripheral operations stop completely.

5.6.2 CG Operation in NORMAL and Standby Modes

Table 5.5 CG States in NORMAL and Standby Modes

Clock Source	Mode	Oscillator	PLL	Clock Supply to Peripherals	Clock Supply to CPU
Crystal	Normal	O	O	O	O
	Idle (Halt)	O	O	Selectable	×
	Idle (Doze)	O	O	Selectable	×
	Stop	×	×	×	×
External Clock	Normal	×	O	O	O
	Idle (Halt)	×	O	Selectable	×
	Idle (Doze)	×	O	Selectable	×
	Stop	×	×	×	×

O: Operational, or clock supplied

×. Stopped, or clock not supplied

5.6.3 Processor and Peripheral Block Operation in Standby Modes

Table 5.6 Processor and Peripheral Blocks in Standby Modes

Circuit Block	Clock Source	IDLE (Doze)	IDLE (Halt)	STOP
TX19 Core Processor	fsys	×	×	×
DMAC		O	×	×
INTC		O	O	×
External Bus Interface		O	×	×
External Bus Mastership		O	×	×
I/O Ports		O	×	×
ADC		Selectable on a block-by-block basis		×
SIO				×
I^2C				×
TMRA				×
TMRB				×
TMRC				×
WDT				×
2-Phase Pulse Input Counter				×
CG	—	O	O	×

O: On
×: Off

5.6.4 Wake-up Signaling

There are two ways to exit a standby mode: an interrupt request or a reset signal. Availability of wakeup signaling depends on the settings of the Interrupt Mask Level bits, CMask[15:13], of the CP0 Status register and the current standby mode (see Table 5.7).

- Wake-up via Interrupt Signaling

  The operation upon return from a standby mode varies, depending on the interrupt priority level programmed before entering a standby mode. If the interrupt priority level is greater than or equal to the processor's interrupt mask level, execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated the standby mode (i.e., the instruction that set the Halt or Doze bit in the Config register). If the interrupt priority level is less than the processor's interrupt mask level, program execution resumes with the instruction that activated the standby mode. The interrupt is left pending.

  Nonmaskable interrupts are always serviced upon return from a standby mode, regardless of the current interrupt mask level.

- Wake-up via Reset Signaling

  Reset signaling always brings the TMP1962 out of any standby mode. A wake-up from STOP mode must, however, allow sufficient time for the oscillator to restart and stabilize (see Table 5.1).

  A reset does not affect the contents of the on-chip RAM, but initializes everything else, whereas an interrupt preserves all internal states that were in effect before the standby mode was entered.

For details of STOP wake-up interrupts and other normal interrupts, refer to Chapter 6, “Interrupts.”

Table 5.7 Wake-up Signaling Sources and Wake-up Operations

Interrupt Masking			Unmasked Interrupt (request_level > mask_level)		Masked Interrupt (request_level ≤ mask_level)	
Standby Mode			IDLE (Programmable)	STOP	IDLE (Programmable)	STOP
Wake-up Signaling Sources	Interrupts	$\overline{\text{NMI}}$	♦	♦ (Note 1)	♦	♦ (Note 1)
		INTWDT	♦	×	♦	×
		INT0 to A	♦	♦ (Note 1)	O	O (Note 1)
		KWUP0 to D	♦	♦ (Note 1)	O	O (Note 1)
		INTTB0 to 3	♦	×	O	×
		INTTA0 to D	♦	×	O	×
		INTRX0 to 6, TX0 to 6	♦	×	O	×
		INTS	♦	×	O	×
		INTAD/ADHP/ADM	♦	×	O	×
	RESET		♦	♦	♦	♦

♦: Execution resumes with the interrupt service routine. (RESET initializes the whole TMP1962.)

O: Execution resumes with the instruction that activated the standby mode. The interrupt is left pending.

×: Cannot be used to exit a standby mode.

Note 1: The TMP1962 exits the stanby mode after the warm-up period timer expires.

Note 2: If the interrupt request level is greater than the mask level, an interrupt signal which is programmed as level-sensitive must be held active until interrupt processing begins. Otherwise, the interrupt will not be serviced successfully.

Note 3: If interrupts are disabled in the CPU, all interrupts other than those used for wake-up signaling must also be disabled in the Interrupt Controller (INTC) before a stanby mode is entered. Otherwise, any interrupt could take the TMP1962 out of the stanby mode.

5.6.5 STOP Mode

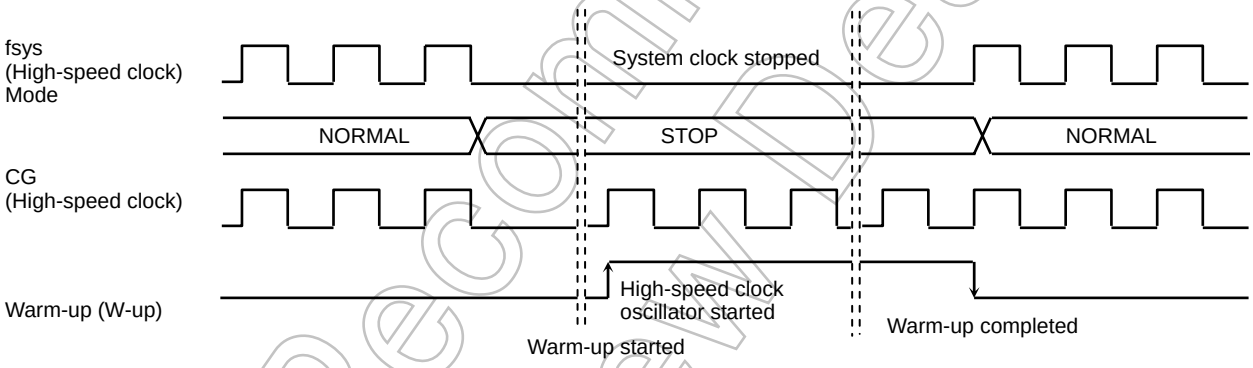
The STOP mode stops the whole TMP1962, including the on-chip oscillator. Pin states in STOP mode depend on the setting of the SYSCR2.DRVE bit, as shown in Table 5.8. Upon detection of wake-up signaling, the warm-up period timer should be activated to allow sufficient time for the oscillator to restart and stabilize before exiting STOP mode. After that, the system clock output can restart. On exiting STOP mode, the TMP1962 starts operation in the mode (NORMAL or SLOW) in which it was in before entering STOP mode.

Applicable register bits must be programmed prior to the instruction that activates a standby mode. The warm-up period is chosen through the SYSCR2.WUPT[1:0] bits.

Note: In the TMP1962F10AXB, the SYSCR2.WUPT[1:0] bits (warm-up time) must not be set to 00 or 01 when the mode is changed from NORMAL to STOP, because this does not allow enough time (at least 150 μs) for the internal system to resume when the TMP1962 exits STOP mode.

5.6.6 Returning from STOP Mode

(1) Mode transitions from NORMAL to STOP to NORMAL



When fosc = 13.5 MHz

W-up Time Select SYSCR2.WUPT[1:0]	W-up Time (fosc)
01 (2^8/fosc)	Setting prohibited
10 (2^{14}/fosc)	1.214 ms
11 (2^{16}/fosc)	4.855 ms

Note: The WUPT[1:0] bits must not be set to 01 because this does not allow enough time for the internal system to resume.

Table 5.8 Pin States in STOP Mode (1/2)

Pins	Input/Output	SYSCR2. DRVE = 0	SYSCR2. DRVE = 1
P00 to P07	Input mode	—	—
	Output mode	—	Output
	AD0 to AD7, D0 to D7	—	—
P10 to P17	Input mode	—	—
	Output mode, A8 to A15	—	Output
	AD8 to AD15, D8 to D15	—	—
P20 to P27	Input mode	—	—
	Output mode, A0 to A7/A16 to A23	—	Output
P30 (/RD), P31 (/WR)	Output pin		Output
P32 to P36	Input mode	PU*	Input
	Output mode	PU*	Output
P37 (ALE)	Input mode	—	Input
	Output mode	—	Output
	ALE (Output mode)	Output Low	Output Low
P40 to P43	Input mode	PU*	Input
	Output mode	PU*	Output
P44 (SCOUT)	Input mode	—	Input
	Output mode	—	Output
P50 to P57	Input mode	—	—
	Output mode, A0 to A7	—	Output
P60 to P67	Input mode	—	—
	Output mode, A8 to A15	—	Output
P7, P8, P9	Input pin	—	—
PA0 to PA7	Input mode	—	Input
	Output mode	—	Output
PB0, PB1, PB4, PB7	Input mode	—	Input
	Output mode	—	Output
PB2, PB3, PB5, PB6	Input mode	—	Input
	Output mode	—	Output
	INT5 to INT8 (Input mode)	Input	Input
PC0 to PC7	Input mode	—	Input
	Output mode	—	Output
PD0 to PD6	Input mode	—	Input
	Output mode	—	Output
PD7	Input mode	—	Input
	Output mode	—	Output
	KEY8 (Input mode)	Input	Input
PE0 to PE2	Input mode	—	Input
	Output mode	—	Output
PE3 to PE7	Input mode	—	Input
	Output mode	—	Output
	KEY9 to KEYD (Input mode)	Input	Input
PF, PG, PH, PI0, PI7	Input mode	—	Input
	Output mode	—	Output
PI1 to PI6	Input mode	—	Input
	Output mode	—	Output
	INT1 to 4,INT9,INTA (Input mode)	Input	Input
PJ0	Input mode	—	Input
	Output mode	—	Output
	INT0 (Input mode)	Input	Input
PJ1 to PJ4	Input mode	—	Input
	Output mode	—	Output
PK0 to PK7	Input mode	—	Input
	Output mode	—	Output
	KEY0 to KEY7 (Input mode)	Input	Input

Table 5.8 Pin States in STOP Mode (2/2)

Pins	Input/Output	SYSCR2. DRVE = 0	SYSCR2. DRVE = 1
PL, PM, PN, PO, PP	Input mode Output mode	— —	Input Output
$\overline{NMI}$	Input pin	Input	Input
$\overline{RESET}$	Input pin	Input	Input
BM0, BM1	Input pin	Input	Input
$\overline{PLLOFF}$	Input pin	Input	Input
RSTPUP	Input pin	Input	Input
SYSRDY	Output pin	Output High	Output High
X1	Input pin	—	—
X2	Output pin	Output High	Output High

—: Pins configured for input mode and input-only pins are disabled. Pins configured for output mode and output-only pins assume the high-impedance state.

Input: The input gate is active; the input voltage must be held at either the high or low level to keep the input pin from floating.

Output: Pin direction is output.

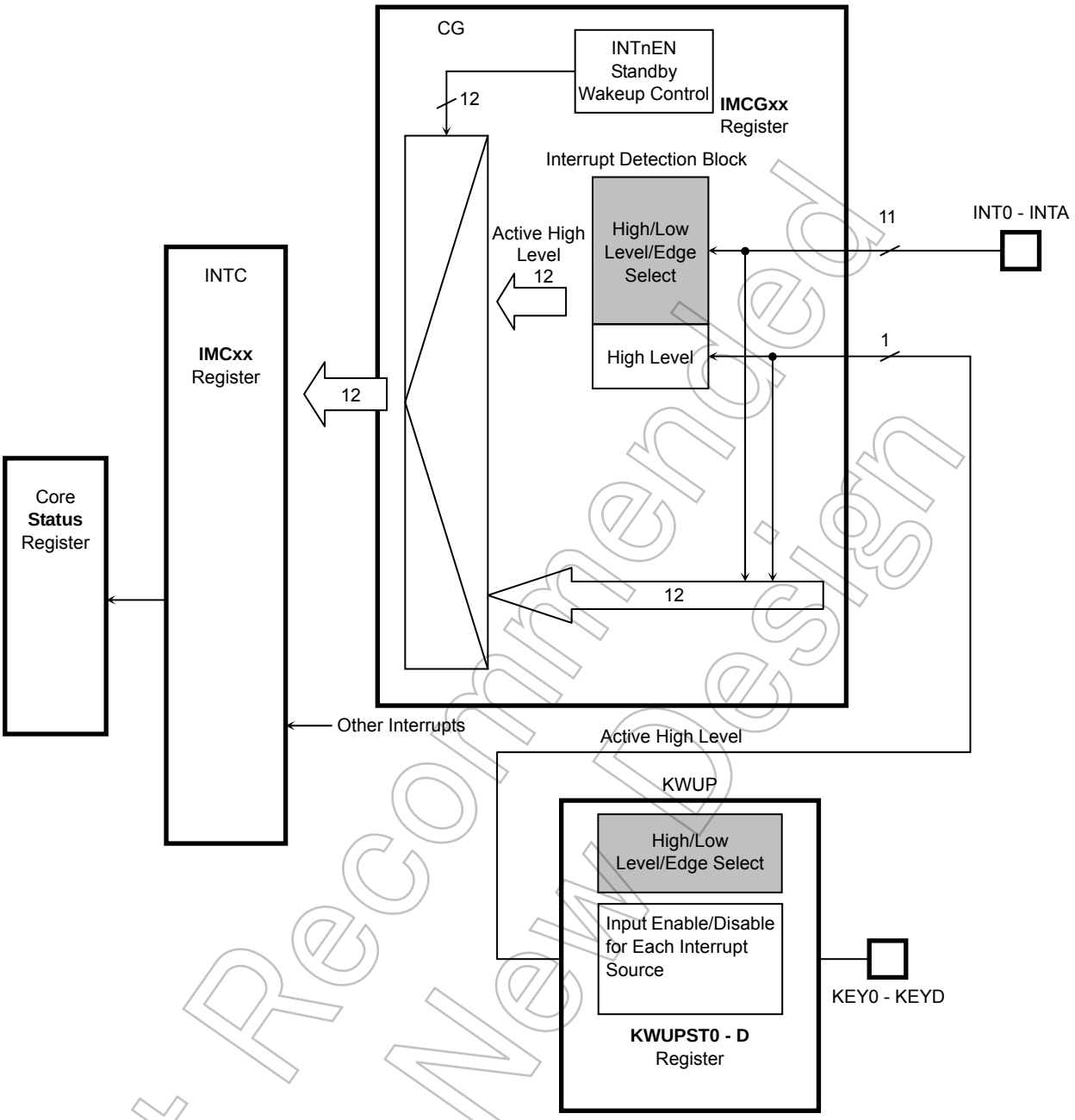
PU*: Programmable pull-up. Because the input gate is always disabled, no overlap current flows while in high-impedance state.

6. Interrupts

Interrupt processing is coordinated between the CP0 Status register, the Interrupt Controller (INTC) and the Clock Generator (CG). The Status register contains the Interrupt Mask Level field (CMask[15:13]) and the Interrupt Enable bit (IEc). For interrupt processing, also refer to Chapter 9, "Exception Handling" in the TX19 Architecture manual.

The TMP1962 interrupt mechanism includes the following features:

- 4 CPU internal interrupts (software interrupts)
- 26 external interrupt pins ($\overline{\text{NMI}}$, INT0-INTA, KWUP0-KWUPD)
- 56 on-chip peripheral interrupts (including a WDT interrupt)
- Vector generation for each interrupt source
- Programmable priority for each interrupt source (7 levels)
- DMA trigger on interrupt



Note: There are interrupt enable and polarity bits in these registers:

1. **Interrupt Mode Control registers (IMCxx) in the INTC**
2. **IMCGxx registers in the CG**
3. **KWUP Status registers (KWUPSTx) in the KWUP**

Figure 6.1 General Interrupt Mechanism

(1) External interrupts INT0-INT4, KWUP0-KWUPD, INTRTC, INTTB2 and INTTB3 (2-phase pulse input counter)

1) INT0-INTA

- When enabled for STOP wake-up signaling
  - The EMCGxx field in the CG's IMCGxx register defines the interrupt polarity. (Refer to Section 5.2.2, "INTCG Registers.")
  - The INTxEN bit in the CG's IMCGxx register controls whether these interrupt sources are enabled as wake-up signal sources (1 = enable). (Refer to Section 5.2.2, "INTCG Registers.")
  - If enabled, the interrupt polarity (EIMxx) field in the INTC's IMCxx register has no effect, but must be set to 01, or high level. (Refer to Section 6.4, "INTC Registers.")
- When disabled for STOP wake-up signaling
  - The interrupt polarity (EIMxx) field in the INTC's IMCxx register defines the interrupt polarity. (Refer to Section 6.4, "INTC Registers.")

2) KWUP0-KWUPD

- When enabled for STOP wake-up signaling
  - The EMCG5[1:0] field in the CG's IMCGB0 register has no effect, but must be set to 01, or high level. (Refer to Section 5.2.2, "INTCG Registers.")
  - The KWUPEN bit in the CG's IMCGB0 register controls whether these interrupt sources are enabled as wake-up signal sources (1 = enable). (Refer to Section 5.2.2, "INTCG Registers.")
  - The interrupt polarity (EIM6[1:0]) field in the INTC's IMC1 register has no effect, but must be set to 01, or high level. (Refer to Section 6.4, "INTC Registers.")
  - For each of these interrupt sources, the KWUPSTx register in the KWUP block defines the interrupt polarity and controls whether interrupts are enabled.
- When disabled for STOP wake-up signaling
  - The interrupt polarity (EIM6[1:0]) field in the INTC's IMC1 register has no effect, but must be set to 01, or high level. (Refer to Section 6.4, "INTC Registers.")
  - For each of these interrupt sources, the KWUPSTx register in the KWUP block defines the interrupt polarity and controls whether interrupts are enabled.

(2) Internal interrupts (except INTRTC and INTTB2/INTTB3 in 2-Phase Pulse Count mode)

These interrupts are programmable through the INTC.

The INTC collects interrupt events, prioritizes them and presents the highest-priority request to the TX19 core processor.

Interrupt		Programming	Interrupt Sensing
INT0 – INTA		IMCGx reg.In CG IMCx reg.In INTC	When enabled for STOP wake-up signaling, the polarity field in the INTC has no effect, but must always be set to "high-level." The actual sensitivity is programmed in the CG. When disabled for STOP wake-up signaling, interrupt sensitivity is programmed in the INTC. In either case, each interrupt source is individually configurable as negative or positive polarity, and as edge-triggered or level-sensitive.
KWUP0 – KWUPD		IMCGx reg.In CG IMCx reg.In INTC KWUPSTn	The polarity field in the INTC has no effect, but must always be set to "high-level." When enabled for STOP wake-up signaling, the polarity field in the CG has no effect, but must always be set to "high-level." The actual sensitivity is programmed in the KWUPnST. When disabled for STOP wake-up signaling, the CG need not be programmed. In either case, each interrupt source is individually configurable as negative or positive polarity, and as edge-triggered or level-sensitive.
On-Chip Peripherals	INTDMAn	IMCx reg.In INTC	Falling edge
	Others	IMCx reg.In INTC	Rising edge

- Example register settings

  Here are example register settings required to enable and disable the INT0 interrupt as a source of the STOP wake-up signal (negative-edge triggered).

  a. Enabling the interrupt

IMCGA0<EMCG01:00> = “10” :	Configure INT0 as negative-edge triggered	CG block
EICRCG<ICRCG2:0> = “000” :	Clear INT0 request	
IMCGA0<INT0EN> = “1” :	Enable INT0 for wake-up signaling	
IMC0L<EIM11:10> = “01” :	Configure INT0 as high-level sensitive	INTC block
INTCLR<EICLR5:0> = “000001” :	Clear INT0 request	
IMC0L<IL12:10> = “101” :	Set INT0 priority level to 5	
Status<IEc> = “1”, <CMask> = “xxx”		TX19 core processor

  b. Disabling the interrupt

Status<IEc> = “0”		TX19 core processor
IMC0L<IL12:10> = “000” :	Disable INT0 interrupt	INTC block
INTCLR<EICLR5:0> = “000001” :	Clear INT0 request	
IMCGA0<INT0EN> = “0” :	Disable INT0 for wake-up signaling	CG block
EICRCG<ICRCG2:0> = “000” :	Clear INT0 request	

6.1 Interrupt Sources

The TMP1962 provides a reset interrupt, nonmaskable interrupts, and maskable interrupts:

(1) Reset and nonmaskable interrupts

The $\overline{\text{RESET}}$ pin causes a Reset interrupt. The $\overline{\text{NMI}}$ pin functions as a nonmaskable interrupt. The on-chip Watchdog Timer (WDT) is also capable of being a source of a nonmaskable interrupt (INTWDT). Reset and nonmaskable interrupts are always vectored to virtual address 0xBFC0_0000.

(2) Maskable interrupts

The TMP1962 supports two types of maskable interrupts: software and hardware interrupts. Maskable interrupts are vectored to virtual addresses 0xBFC0_0210 through 0xBFC0_0260, as shown below.

Interrupt Source			Virtual Vector Address
Reset			0xBFC0_0000
Nonmaskable			
Maskable	Software	Swi0	0xBFC0_0210
		Swi1	0xBFC0_0220
		Swi2	0xBFC0_0230
		Swi3	0xBFC0_0240
	Hardware		0xBFC0_0260

Note 1: The above table shows the vector addresses when the BEV bit in the CP0 Status register is set to 1. When BEV = 1, all exception vectors reside in the on-chip ROM space.

Note 2: Software interrupts are posted by setting one of the Sw[3:0] bits in the CP0 Cause register. Software interrupts are distinct from the "Software Set" interrupt which is one of the hardware interrupt sources. A Software Set interrupt is posted from the INTC to the TX19 core processor when the IL0[2:0] field in the INTC's IMC0 register is set to a non-zero value.

Table 6.1 Hardware Interrupt Sources

Interrupt Number	IVR[9 : 0]	Interrupt Source	Interrupt Control Register	Address
0	000	Software Set	IMC0	0xFFFF_E000
1	010	INT0 pin		
2	020	INT1 pin		
3	030	INT2 pin		
4	040	INT3 pin	IMC1	0xFFFF_E004
5	050	INT4 pin		
6	060	KWUP		
7	070	reserved		
8	080	INTRX6: SIO receive (channel.6)	IMC2	0xFFFF_E008
9	090	INTTX6: SIO transmit (channel.6)		
10	0A0	INT5 pin		
11	0B0	INT6 pin		
12	0C0	INT7 pin	IMC3	0xFFFF_E00C
13	0D0	INT8 pin		
14	0E0	INT9 pin		
15	0F0	INTA pin		
16	100	INTRX0: SIO receive (channel.0)	IMC4	0xFFFF_E010
17	110	INTTX0: SIO transmit (channel.0)		
18	120	INTRX1: SIO receive (channel.1)		
19	130	INTTX1: SIO transmit (channel.1)		
20	140	INTS0: Serial Bus Interface 0	IMC5	0xFFFF_E014
21	150	INTRX2: SIO receive (channel.2)		
22	160	INTTX2: SIO transmit (channel.2)		
23	170	INTADHP: High-priority A/D conversion complete		
24	180	INTADM: A/D conversion monitoring	IMC6	0xFFFF_E018
25	190	INTTAG0: 8-bit timer group 0		
26	1A0	INTTAG1: 8-bit timer group 1		
27	1B0	INTTAG2: 8-bit timer group 2		
28	1C0	reserved	IMC7	0xFFFF_E01C
29	1D0	INTTB0: 16-bit timer 0		
30	1E0	INTTB1: 16-bit timer 1		
31	1F0	INTRX3: SIO receive (channel.3)		
32	200	INTTX3: SIO transmit (channel.3)	IMC8	0xFFFF_E020
33	210	INTRX4: SIO receive (channel.4)		
34	220	INTTX4: SIO transmit (channel.4)		
35	230	INTRX5: SIO receive (channel.5)		
36	240	INTTX5: SIO transmit (channel.5)	IMC9	0xFFFF_E024
37	250	reserved		
38	260	reserved		
39	270	reserved		
40	280	INTCAPG0: Input capture group 0	IMCA	0xFFFF_E028
41	290	INTCAPG1: Input capture group 1		
42	2A0	reserved		
43	2B0	INTCMP0: Compare 0		
44	2C0	INTCMP1: Compare 1	IMCB	0xFFFF_E02C
45	2D0	INTCMP2: Compare 2		
46	2E0	INTCMP3: Compare 3		
47	2F0	INTCMP4: Compare 4		
48	300	INTCMP5: Compare 5	IMCC	0xFFFF_E030
49	310	INTTB2: 16-bit timer 2		
50	320	INTTB3: 16-bit timer 3		
51	330	INTCMP6: Compare 6		
52	340	INTCMP7: Compare 7	IMCD	0xFFFF_E034
53	350	reserved		
54	360	INTDMA0: DMA complete (Channel 0)		
55	370	INTDMA1: DMA complete (Channel 1)		
56	380	INTDMA2: DMA complete (Channel 2)	IMCE	0xFFFF_E038
57	390	INTDMA3: DMA complete (Channel 3)		
58	3A0	reserved		
59	3B0	INTAD: A/D conversion complete		
60	3C0	INTDMA4: DMA complete (Channel 4)	IMCF	0xFFFF_E03C
61	3D0	INTDMA5: DMA complete (Channel 5)		
62	3E0	INTDMA6: DMA complete (Channel 6)		
63	3F0	INTDMA7: DMA complete (Channel 7)		

6.2 Interrupt Detection

When enabled as a STOP wake-up signal, the polarities of INT0-INT4 are programmed in the EMCGxx field of the IMCGxx register within the CG; in this case, the EIMxx field of the IMCx register within the INTC has no effect; however, it must be set to "high-level sensitive". For each of KWUP0-KWUPD, the KWUPSTn register within the KWUP block defines the interrupt polarity and controls whether interrupts are enabled. The EMCG field of the IMCGB register within the CG and the EIMxx field of the IMCx register within the INTC have no effect; however, they must be set to "high-level sensitive". The polarity of INTRTC must be configured as "rising-edge triggered" in the EMCGxx field of the IMCGxx register within the CG; in this case, the EIMxx field of the IMCx register within the INTC has no effect; however, it must be set to "high-level sensitive". All other interrupts are always programmed in the EMCGxx field of the INTC's IMCx register. Each interrupt source is individually configurable as negative or positive polarity, and as edge-triggered or level-sensitive. When a selected transition is detected, an interrupt request is issued to the INTC (except for the NMI and INTWDT interrupts, which are directly delivered to the TX19 core processor). When the above interrupts are disabled for STOP wake-up signaling, the CG need not be programmed. When INT0-INTA are disabled for wake-up signaling, only the INTC has to be programmed. When KWUP0-KWUPD are disabled for wake-up signaling, only the INTC and KWUPSTx have to be programmed.

It is the responsibility of software (an interrupt handler routine) to determine the cause of an interrupt and to clear the interrupt condition. INT0-INTA and INTRC require software access to two registers: the EICRCG register (ICRCG field) in the CG and the INTCLR register (EICLR field) in the INTC. KWUP0-KWUPD require software access to the KWUPCLR. Other interrupts can be cleared by writing their assigned value to the EICLR field in the INTC's INTCLR register. For an external interrupt configured as level-sensitive, software must explicitly address the device in question and clear the interrupt condition. A level-sensitive interrupt signal must be held active until the TX19 core processor reads its interrupt vector from the Interrupt Vector Register (IVR).

Note: To use an interrupt for wake-up signaling, define the polarity, clear the interrupt request and then enable the interrupt, always in the stated order.

(Example register settings required to enable the INT0 interrupt as a source of the STOP wake-up signal)

IMCGA0<EMCG01:00> = "10" :	Configure INT0 as negative-edge triggered	CG block
EICRCG<ICRCG2:0> = "000" :	Clear INT0 request	
IMCGA0<INT0EN> = "1" :	Enable INT0 for wake-up signaling	
IMC0L<EIM11:10> = "01" :	Configure INT0 as high-level sensitive	INTC block
INTCLR<EICLR5:0> = "000001" :	Clear INT0 request	
IMC0L<IL12:10> = "101" :	Set INT0 priority level to 5	
Status<IEc> = "1", <CMask> = "xxx" :		TX19 core processor

6.3 Resolving Interrupt Priority

(1) Seven interrupt priority levels

The Interrupt Mode Control registers (IMCx) contain a 3-bit interrupt priority level (ILx[2:0]) field for each interrupt source, which ranges from level 0 to level 7, with level 7 being the highest priority. Level 0 indicates that the interrupt is disabled.

(2) Interrupt level notification

When an interrupt event occurs, the INTC sends its priority level to the TX19 core processor. The processor can determine the priority level of an interrupt being requested by reading the IL field in the CP0 Cause register. If multiple interrupt events having different priority levels occur simultaneously, the INTC sends the highest priority level.

(3) Interrupt vector (interrupt source notification)

Whenever an interrupt request is made, the INTC automatically sets its vector in the IVR. The TX19 core processor can determine the exact cause of an interrupt by reading the IVR. If multiple interrupt requests occur at the same level, the interrupt with the smallest interrupt number is delivered. When no interrupt is pending, the IVR[9:4] field in the IVR contains a value of zero.

When the TX19 core processor responds to a request with an interrupt acknowledge cycle, the INTC forwards the interrupt vector for that interrupt request. At this time, the TX19 core processor saves the priority level value in the CMask field of the CP0 Status register.

6.4 Register Description

Table 6.2 INTC Register Map

Address	Symbol	Register Name	Corresponding Interrupt Number
0xFFFF_E060	INTCLR	Interrupt Request Clear Control Register	ALL (63 – 0)
0xFFFF_E040	IVR	Interrupt Vector Register	ALL (63 – 0)
0xFFFF_E03C	IMCF	Interrupt Mode Control Register F	63 – 60
0xFFFF_E038	IMCE	Interrupt Mode Control Register E	59 – 56
0xFFFF_E034	IMCD	Interrupt Mode Control Register D	55 – 52
0xFFFF_E030	IMCC	Interrupt Mode Control Register C	51 – 48
0xFFFF_E02C	IMCB	Interrupt Mode Control Register B	47 – 44
0xFFFF_E028	IMCA	Interrupt Mode Control Register A	43 – 40
0xFFFF_E024	IMC9	Interrupt Mode Control Register 9	39 – 36
0xFFFF_E020	IMC8	Interrupt Mode Control Register 8	35 – 32
0xFFFF_E01C	IMC7	Interrupt Mode Control Register 7	31 – 28
0xFFFF_E018	IMC6	Interrupt Mode Control Register 6	27 – 24
0xFFFF_E014	IMC5	Interrupt Mode Control Register 5	23 – 20
0xFFFF_E010	IMC4	Interrupt Mode Control Register 4	19 – 16
0xFFFF_E00C	IMC3	Interrupt Mode Control Register 3	15 – 12
0xFFFF_E008	IMC2	Interrupt Mode Control Register 2	11 – 8
0xFFFF_E004	IMC1	Interrupt Mode Control Register 1	7 – 4
0xFFFF_E000	IMC0	Interrupt Mode Control Register 0	3 – 0

6.4.1 Interrupt Vector Register (IVR)

This register indicates the vector for the interrupt source when there is an interrupt event.

IVR (0xFFFF_E040)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	15	14	13	12	11	10	9	8
Bit Symbol							IVR9	IVR8
Read/Write	R/W						R	
Reset Value	0	0	0	0	0	0	0	0
Function							Interrupt vector for the source of the current interrupt	
	7	6	5	4	3	2	1	0
Bit Symbol	IVR7	IVR6	IVR5	IVR4				
Read/Write	R							
Reset Value	0	0	0	0	0	0	0	0
Function	Interrupt vector for the source of the current interrupt							

6.4.2 Interrupt Mode Control Registers

These registers control the interrupt priority level, active polarity, either level or edge sensitivity, and DMA triggering.

IMC0 (0xFFFF_E000)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM31	EIM30	DM3	IL32	IL31	IL30
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 3 as DMA trigger	When DM3 = 0 Interrupt number 3 (INT2) priority level 000: Interrupt disabled 001-111: 1-7 When DM3 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		
	23	22	21	20	19	18	17	16
Bit Symbol			EIM21	EIM20	DM2	IL22	IL21	IL20
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 2 as DMA trigger	When DM2 = 0 Interrupt number 2 (INT1) priority level 000: Interrupt disabled 001-111: 1-7 When DM2 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		
	15	14	13	12	11	10	9	8
Bit Symbol			EIM11	EIM10	DM1	IL12	IL11	IL10
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 1 as DMA trigger	When DM1 = 0 Interrupt number 1 (INT0) priority level 000: Interrupt disabled 001-111: 1-7 When DM1 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		
	7	6	5	4	3	2	1	0
Bit Symbol			EIM01	EIM00	DM0	IL02	IL01	IL00
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: Setting prohibited 10: Setting prohibited 11: Setting prohibited Must be set to 00.		DMA trigger 0: Disable 1: Enable interrupt number 0 as DMA trigger	When DM0 = 0 Interrupt number 0 (Software Set) priority level 000: Interrupt disabled 001-111: 1-7 When DM0 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC1 (0xFFFF_E004)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM71	EIM70	DM7	IL72	IL71	IL70
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM61	EIM60	DM6	IL62	IL61	IL60
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: High level 10: Setting prohibited 11: Setting prohibited Must be set to 01.		DMA trigger 0: Disable 1: Enable interrupt number 6 as DMA trigger	When DM6 = 0 Interrupt number 6 (KWUP) priority level 000: Interrupt disabled 001-111: 1-7 When DM6 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM51	EIM50	DM5	IL52	IL51	IL50
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 5 as DMA trigger	When DM5 = 0 Interrupt number 5 (INT4) priority level 000: Interrupt disabled 001-111: 1-7 When DM5 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM41	EIM40	DM4	IL42	IL41	IL40
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 4 as DMA trigger	When DM4 = 0 Interrupt number 4 (INT3) priority level 000: Interrupt disabled 001-111: 1-7 When DM4 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC2 (0xFFFF_E008)

	31	30	29	28	27	26	25	24
Bit Symbol			EIMB1	EIMB0	DMB	ILB2	ILB1	ILB0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 11 as DMA trigger	When DMB = 0 Interrupt number 11 (INT6) priority level 000: Interrupt disabled 001-111: 1-7 When DMB = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIMA1	EIMA0	DMA	ILA2	ILA1	ILA0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 10 as DMA trigger	When DMA = 0 Interrupt number 10 (INT5) priority level 000: Interrupt disabled 001-111: 1-7 When DMA = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM91	EIM90	DM9	IL92	IL91	IL90
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 9 as DMA trigger	When DM9 = 0 Interrupt number 9 (INTTX6) priority level 000: Interrupt disabled 001-111: 1-7 When DM9 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM81	EIM80	DM8	IL82	IL81	IL80
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 8 as DMA trigger	When DM8 = 0 Interrupt number 8 (INTRX6) priority level 000: Interrupt disabled 001-111: 1-7 When DM8 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC3 (0xFFFF_E00C)

	31	30	29	28	27	26	25	24
Bit Symbol			EIMF1	EIMF0	DMF	ILF2	ILF1	ILF0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 15 as DMA trigger	When DMF = 0 Interrupt number 15 (INTA) priority level 000: Interrupt disabled 001-111: 1-7 When DMF = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIME1	EIME0	DME	ILE2	ILE1	ILE0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 14 as DMA trigger	When DME = 0 Interrupt number 15 (INT9) priority level 000: Interrupt disabled 001-111: 1-7 When DME = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIMD1	EIMD0	DMD	ILD2	ILD1	ILD0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 13 as DMA trigger	When DMD = 0 Interrupt number 13 (INT8) priority level 000: Interrupt disabled 001-111: 1-7 When DMD = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIMC1	EIMC0	DMC	ILC2	ILC1	ILC0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable interrupt number 12 as DMA trigger	When DMC = 0 Interrupt number 12 (INT7) priority level 000: Interrupt disabled 001-111: 1-7 When DMC = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC4 (0xFFFF_E010)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM131	EIM130	DM13	IL132	IL131	IL130
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 19 as DMA trigger	When DM13 = 0 Interrupt number 19 (INTTX1) priority level 000: Interrupt disabled 001-111: 1-7 When DM13 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM121	EIM120	DM12	IL122	IL121	IL120
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 18 as DMA trigger	When DM12 = 0 Interrupt number 18 (INTRX1) priority level 000: Interrupt disabled 001-111: 1-7 When DM12 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM111	EIM110	DM11	IL112	IL111	IL110
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 17 as DMA trigger	When DM11 = 0 Interrupt number 17 (INTTX0) priority level 000: Interrupt disabled 001-111: 1-7 When DM11 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM101	EIM100	DM10	IL102	IL101	IL100
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 16 as DMA trigger	When DM10 = 0 Interrupt number 16 (INTRX0) priority level 000: Interrupt disabled 001-111: 1-7 When DM10 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC5 (0xFFFF_E014)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM171	EIM170	DM17	IL172	IL171	IL170
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 23 as DMA trigger	When DM17 = 0 Interrupt number 23 (INTADHP) priority level 000: Interrupt disabled 001-111: 1-7 When DM17 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM161	EIM160	DM16	IL162	IL161	IL160
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 22 as DMA trigger	When DM16 = 0 Interrupt number 22 (INTTX2) priority level 000: Interrupt disabled 001-111: 1-7 When DM16 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM151	EIM150	DM15	IL152	IL151	IL150
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 21 as DMA trigger	When DM15 = 0 Interrupt number 21 (INTRX2) priority level 000: Interrupt disabled 001-111: 1-7 When DM15 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM141	EIM140	DM14	IL142	IL141	IL140
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 20 as DMA trigger	When DM14 = 0 Interrupt number 20 (INTS0) priority level 000: Interrupt disabled 001-111: 1-7 When DM14 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC6 (0xFFFF_E018)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM1B1	EIM1B0	DM1B	IL1B2	IL1B1	IL1B0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 27 as DMA trigger	When DM1B = 0 Interrupt number 27 (INTTAG2) priority level 000: Interrupt disabled 001-111: 1-7 When DM1B = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM1A1	EIM1A0	DM1A	IL1A2	IL1A1	IL1A0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 26 as DMA trigger	When DM1A = 0 Interrupt number 26 (INTTAG1) priority level 000: Interrupt disabled 001-111: 1-7 When DM1A = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM191	EIM190	DM19	IL192	IL191	IL190
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 25 as DMA trigger	When DM19 = 0 Interrupt number 25 (INTTAG0) priority level 000: Interrupt disabled 001-111: 1-7 When DM19 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM181	EIM180	DM18	IL182	IL181	IL180
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 24 as DMA trigger	When DM18 = 0 Interrupt number 24 (INTADM) priority level 000: Interrupt disabled 001-111: 1-7 When DM18 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC7 (0xFFFF_E01C)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM1F1	EIM1F0	DM1F	IL1F2	IL1F1	IL1F0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 31 as DMA trigger	When DM1F = 0 Interrupt number 31 (INTRX3) priority level 000: Interrupt disabled 001-111: 1-7 When DM1F = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM1E1	EIM1E0	DM1E	IL1E2	IL1E1	IL1E0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 30 as DMA trigger	When DM1E = 0 Interrupt number 30 (INTTB1) priority level 000: Interrupt disabled 001-111: 1-7 When DM1E = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM1D1	EIM1D0	DM1D	IL1D2	IL1D1	IL1D0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 29 as DMA trigger	When DM1D = 0 Interrupt number 29 (INTTB0) priority level 000: Interrupt disabled 001-111: 1-7 When DM1D = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM1C1	EIM1C0	DM1C	IL1C2	IL1C1	IL1C0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		

IMC8 (0xFFFF_E020)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM231	EIM230	DM23	IL232	IL231	IL230
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 35 as DMA trigger	When DM23 = 0 Interrupt number 35 (INTRX5) priority level 000: Interrupt disabled 001-111: 1-7 When DM23 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM221	EIM220	DM22	IL222	IL221	IL220
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 34 as DMA trigger	When DM22 = 0 Interrupt number 34 (INTTX4) priority level 000: Interrupt disabled 001-111: 1-7 When DM22 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM211	EIM210	DM21	IL212	IL211	IL210
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 33 as DMA trigger	When DM21 = 0 Interrupt number 33 (INTRX4) priority level 000: Interrupt disabled 001-111: 1-7 When DM21 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM201	EIM200	DM20	IL202	IL201	IL200
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 32 as DMA trigger	When DM20 = 0 Interrupt number 32 (INTTX3) priority level 000: Interrupt disabled 001-111: 1-7 When DM20 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMC9 (0xFFFF_E024)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM271	EIM270	DM27	IL272	IL271	IL270
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		
	23	22	21	20	19	18	17	16
Bit Symbol			EIM261	EIM260	DM26	IL262	IL261	IL260
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		
	15	14	13	12	11	10	9	8
Bit Symbol			EIM251	EIM250	DM25	IL252	IL251	IL250
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		
	7	6	5	4	3	2	1	0
Bit Symbol			EIM241	EIM240	DM24	IL242	IL241	IL240
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 36 as DMA trigger	When DM24 = 0 Interrupt number 36 (INTTX5) priority level 000: Interrupt disabled 001-111: 1-7 When DM24 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMCA (0xFFFF_E028)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM2B1	EIM2B0	DM2B	IL2B2	IL2B1	IL2B0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 43 as DMA trigger	When DM2B = 0 Interrupt number 43 (INTCMP0) priority level 000: Interrupt disabled 001-111: 1-7 When DM2B = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM2A1	EIM2A0	DM2A	IL2A2	IL2A1	IL2A0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM291	EIM290	DM29	IL292	IL291	IL290
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 41 as DMA trigger	When DM29 = 0 Interrupt number 41 (INTCAPG1) priority level 000: Interrupt disabled 001-111: 1-7 When DM29 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM281	EIM280	DM28	IL282	IL281	IL280
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 40 as DMA trigger	When DM28 = 0 Interrupt number 40 (INTCAPG0) priority level 000: Interrupt disabled 001-111: 1-7 When DM28 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMCB (0xFFFF_E02C)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM2F1	EIM2F0	DM2F	IL2F2	IL2F1	IL2F0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 47 as DMA trigger	When DM2F = 0 Interrupt number 47 (INTCMP4) priority level 000: Interrupt disabled 001-111: 1-7 When DM2F = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM2E1	EIM2E0	DM2E	IL2E2	IL2E1	IL2E0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 46 as DMA trigger	When DM2E = 0 Interrupt number 46 (INTCMP3) priority level 000: Interrupt disabled 001-111: 1-7 When DM2E = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM2D1	EIM2D0	DM2D	IL2D2	IL2D1	IL2D0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 45 as DMA trigger	When DM2D = 0 Interrupt number 45 (INTCMP2) priority level 000: Interrupt disabled 001-111: 1-7 When DM2D = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM2C1	EIM2C0	DM2C	IL2C2	IL2C1	IL2C0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 44 as DMA trigger	When DM2C = 0 Interrupt number 44 (INTCMP1) priority level 000: Interrupt disabled 001-111: 1-7 When DM2C = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMCC (0xFFFF_E030)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM331	EIM330	DM33	IL332	IL331	IL330
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 51 as DMA trigger	When DM33 = 0 Interrupt number 51 (INTCMP6) priority level 000: Interrupt disabled 001-111: 1-7 When DM33 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM321	EIM320	DM32	IL322	IL321	IL320
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: High level 10: Setting prohibited 11: Rising edge Must be set to 01 when used for STOP wake-up signaling; otherwise, must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 50 as DMA trigger	When DM32 = 0 Interrupt number 50 (INTTB3) priority level 000: Interrupt disabled 001-111: 1-7 When DM32 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM311	EIM310	DM31	IL312	IL311	IL310
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: High level 10: Setting prohibited 11: Rising edge Must be set to 01 when used for STOP wake-up signaling; otherwise, must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 49 as DMA trigger	When DM31 = 0 Interrupt number 49 (INTTB2) priority level 000: Interrupt disabled 001-111: 1-7 When DM31 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM301	EIM300	DM30	IL302	IL301	IL300
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 48 as DMA trigger	When DM30 = 0 Interrupt number 48 (INTCMP5) priority level 000: Interrupt disabled 001-111: 1-7 When DM30 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMCD (0xFFFF_E034)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM371	EIM370	DM37	IL372	IL371	IL370
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Rising edge Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 55 as DMA trigger	When DM37 = 0 Interrupt number 55 (INTDMA1) priority level 000: Interrupt disabled 001-111: 1-7 When DM37 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM361	EIM360	DM36	IL362	IL361	IL360
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Rising edge Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 54 as DMA trigger	When DM36 = 0 Interrupt number 54 (INTDMA0) priority level 000: Interrupt disabled 001-111: 1-7 When DM36 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM351	EIM350	DM35	IL352	IL351	IL350
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Must be set to 00.		Must be set to 0.	Must be set to 000.		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM341	EIM340	DM34	IL342	IL341	IL340
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 52 as DMA trigger	When DM34 = 0 Interrupt number 52 (INTCMP7) priority level 000: Interrupt disabled 001-111: 1-7 When DM34 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMCE (0xFFFF_E038)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM3B1	EIM3B0	DM3B	IL3B2	IL3B1	IL3B0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: Rising edge Must be set to 11.		DMA trigger 0: Disable 1: Enable interrupt number 59 as DMA trigger	When DM3B = 0 Interrupt number 59 (INTAD) priority level 000: Interrupt disabled 001-111: 1-7 When DM3B = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM3A1	EIM3A0	DM3A	IL3A2	IL3A1	IL3A0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: High level 10: Setting prohibited 11: Setting prohibited Must be set to 01.		DMA trigger 0: Disable 1: Enable interrupt number 58 as DMA trigger	When DM3A = 0 Interrupt number 58 (INTRTC) priority level 000: Interrupt disabled 001-111: 1-7 When DM3A = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM391	EIM390	DM39	IL392	IL391	IL390
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Setting prohibited Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 57 as DMA trigger	When DM39 = 0 Interrupt number 57 (INTDMA3) priority level 000: Interrupt disabled 001-111: 1-7 When DM39 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM381	EIM380	DM38	IL382	IL381	IL380
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Setting prohibited Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 56 as DMA trigger	When DM38 = 0 Interrupt number 56 (INTDMA2) priority level 000: Interrupt disabled 001-111: 1-7 When DM38 = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

IMCF (0xFFFF_E03C)

	31	30	29	28	27	26	25	24
Bit Symbol			EIM3F1	EIM3F0	DM3F	IL3F2	IL3F1	IL3F0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Setting prohibited Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 63 as DMA trigger	When DM3F = 0 Interrupt number 63 (INTDMA7) priority level 000: Interrupt disabled 001-111: 1-7 When DM3F = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	23	22	21	20	19	18	17	16
Bit Symbol			EIM3E1	EIM3E0	DM3E	IL3E2	IL3E1	IL3E0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Setting prohibited Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 62 as DMA trigger	When DM3E = 0 Interrupt number 62 (INTDMA6) priority level 000: Interrupt disabled 001-111: 1-7 When DM3E = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	15	14	13	12	11	10	9	8
Bit Symbol			EIM3D1	EIM3D0	DM3D	IL3D2	IL3D1	IL3D0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Setting prohibited Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 61 as DMA trigger	When DM3D = 0 Interrupt number 61 (INTDMA5) priority level 000: Interrupt disabled 001-111: 1-7 When DM3D = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

	7	6	5	4	3	2	1	0
Bit Symbol			EIM3C1	EIM3C0	DM3C	IL3C2	IL3C1	IL3C0
Read/Write			R/W					
Reset Value			0	0	0	0	0	0
Function			Interrupt sensitivity 00: Setting prohibited 01: Setting prohibited 10: Falling edge 11: Setting prohibited Must be set to 10.		DMA trigger 0: Disable 1: Enable interrupt number 60 as DMA trigger	When DM3C = 0 Interrupt number 60 (INTDMA4) priority level 000: Interrupt disabled 001-111: 1-7 When DM3C = 1 DMAC channel select 000-011: 0-3 100-111: 4-7		

Note 1: Interrupt sensitivity must be programmed when interrupts are enabled.

Note 2 When an interrupt is used to trigger a DMAC channel, that DMAC channel must be put in Ready state after the programming of the INTC.

6.4.3 Interrupt Request Clear Register

Loading the EICLR[5:0] field of this register with the IVR[9:4] value of the IVR causes the corresponding interrupt to be cleared.

INTCLR (0xFFFF_E060)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write								
Reset Value								
Function								
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write								
Reset Value								
Function								
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write								
Reset Value								
Function								
	7	6	5	4	3	2	1	0
Bit Symbol			EICLR5	EICLR4	EICLR3	EICLR2	EICLR1	EICLR0
Read/Write			W					
Reset Value			—	—	—	—	—	—
Function			IVRL[9:4] value for an interrupt to be cleared					

Note 1: An interrupt request must not be cleared before the TX19 core processor reads the IVR value.

Note 2: Follow the steps below to disable a particular interrupt with the Interrupt Controller (INTC).

1. **Globally disable the acceptance of interrupts by the core processor by clearing the IEc bit of the Status register.**
2. **Disable the desired interrupt with the INTC by clearing the ILx[2:0] field of the IMCxx register.**
3. **Execute the SYNC instruction.**
4. **Enable the acceptance of interrupts by the core processor by setting the IEc bit of the Status register.**

Example:

```
mtc0    r0,    r31    ;  _DI () ;
sb      r0,    IMC**  ;  IMC** = 0 ;
sync                  ;  _SYNC () ;
mtc0    $sp,   r31    ;  _EI () ;
```

6.4.4 INTCG Registers (STOP Wake-up Signaling)

STOP Wake-up Signaling

IMCGA0 (0xFFFF_EE10)

	31	30	29	28	27	26	25	24
Bit Symbol			EMCG31	EMCG30				INT3EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT3 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT3 for standby wake-up signaling 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit Symbol			EMCG21	EMCG20				INT2EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT2 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT2 for standby wake-up signaling 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit Symbol			EMCG11	EMCG10				INT1EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT1 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT1 for standby wake-up signaling 0: Disable 1: Enable
	7	6	5	4	3	2	1	0
Bit Symbol			EMCG01	EMCG00				INT0EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT0 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT0 for standby wake-up signaling 0: Disable 1: Enable

IMCGB0
(0xFFFF_EE14)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write			R/W					R/W
Reset Value			1	1				0
Function								Must be set to 0.
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write			R/W					R/W
Reset Value			1	0				0
Function								Must be set to 0.
	15	14	13	12	11	10	9	8
Bit Symbol			EMCG51	EMCG50				KWUPEN
Read/Write			R/W					R/W
Reset Value			0	1				0
Function			KWUP standby wake-up interrupt sensitivity 00: Setting prohibited 01: High level 10: Setting prohibited 11: Setting prohibited Must be set to 01.					KWUP for standby wake-up signaling 0: Disable 1: Enable
	7	6	5	4	3	2	1	0
Bit Symbol			EMCG41	EMCG40				INT4EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT4 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT4 for standby wake-up signaling 0: Disable 1: Enable

IMCGC0 (0xFFFF_EE18)

	31	30	29	28	27	26	25	24
Bit Symbol			EMCGB1	EMCGB0				INT6EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT6 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT6 for standby wake-up signaling 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit Symbol			EMCGA1	EMCGA0				INT5EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT5 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT5 for standby wake-up signaling 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write			R/W					R/W
Reset Value			1	1				0
Function								Must be set to 0.
	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write			R/W					R/W
Reset Value			1	1				0
Function								Must be set to 0.

IMCGD0
(0xFFFF_EE1C)

	31	30	29	28	27	26	25	24
Bit Symbol			EMCGF1	EMCGF0				INTAEN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INTA standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INTA for standby wake-up signaling 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
Bit Symbol			EMCGE1	EMCGE0				INT9EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT9 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT9 for standby wake-up signaling 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
Bit Symbol			EMCGD1	EMCGD0				INT8EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT8 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT8 for standby wake-up signaling 0: Disable 1: Enable

	7	6	5	4	3	2	1	0
Bit Symbol			EMCGC1	EMCGC0				INT7EN
Read/Write			R/W					R/W
Reset Value			1	0				0
Function			INT7 standby wake-up interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT7 for standby wake-up signaling 0: Disable 1: Enable

Note 1: Interrupt sensitivity must be programmed when interrupts are enabled for STOP wake-up signaling.

Note 2: Follow the steps below to use an interrupt.

1. **Enable the corresponding pin as an interrupt source if it is also used as a general-purpose port pin or has other functions.**
2. **Specify the interrupt sensitivity during initialization.**
3. **Clear any corresponding interrupt request.**
4. **Enable the interrupt source.**

Note 3: The interrupt sensitivity and other settings must be programmed when the interrupt is disabled.

Note 4: The TMP1962 supports the use of 15 interrupt sources for STOP wake-up signaling: INT0-INTA, INTRTC, INTTB2/INTTB3 and KWUP0-KWUPD. For INT0-INTA, the CG block controls whether these interrupt sources are enabled as wake-up signal sources and defines the interrupt sensitivity. For KWUP0-KWUPD, the CG block controls whether these interrupt sources are enabled as wake-up signal sources while the KWUPSTx defines the interrupt sensitivity. For the above 15 interrupt sources, the interrupt sensitivity field in the INTC has no effect, but it must be set to "high-level."

Example: Enabling the INT0 interrupt

IMCGA0<EMCG01:00> = “10”
IMCGA0<INT0EN> = “1” } **CG block (Configure INT0 as falling-edge triggered)**

IMC0L<EIM11:10> = “01”
IMC0L<IL12:10> = “101” } **INTC block (Configure INT0 as high-level sensitive and set INT0 priority level to 5)**

Interrupts other than those used for STOP wake-up signaling are programmed in the INTC block.

Note 5: When INT0-INTA are used as general-purpose interrupts, the INTC defines the interrupt sensitivity; the CG need not be programmed. When KWUP0-KWUPD are used as general-purpose interrupts, the KWUPSTn register defines the interrupt sensitivity; the CG need not be programmed, but the interrupt sensitivity field in the INTC must be set to "high-level." INTRTC requires settings in both the CG and INTC even when it is used as a general-purpose interrupt.

Interrupts other than those used for STOP wake-up signaling are programmed in the INTC block.

EICRCG (0xFFFF_EE20)

		31	30	29	28	27	26	25	24
	Bit Symbol								
	Read/Write								
	Reset Value								
	Function								
		23	22	21	20				
	Bit Symbol								
	Read/Write								
	Reset Value								
	Function								
		15	14	13	12				
	Bit Symbol								
	Read/Write								
	Reset Value								
	Function								
		7	6	5	4	3	2	1	0
	Bit Symbol					ICRCG3	ICRCG2	ICRCG1	ICRCG0
	Read/Write					W			
	Reset Value					—	—	—	—
	Function					Clear the corresponding interrupt request. 0000: INT0 0101: KWUP 1010: INT5 0001: INT1 0110: reserved 1011: INT6 0010: INT2 0111: reserved 1100: INT7 0011: INT3 1000: reserved 1101: INT8 0100: INT4 1001: reserved 1110: INT9 1111: INTA			

Note 6: To clear interrupts used for STOP wake-up signaling, program the following registers:

1. **For KWUP, program the KWUPCLR.**
2. **For INT0-INTA, INTTB2, INTTB3, and INTRTC, program both the EICRCG register in the CG block, shown above, and the INTCLR register in the INTC block.**
3. **For other interrupt sources, program the INTCRL register in the INTC block.**

7. I/O Ports

7.1 Port 0 (P00 - P07)

Eight Port 0 pins can be individually programmed to function as discrete general-purpose I/O pins, the D[0:7] bits of the data bus, or the AD[0:7] bits of the address/data bus. The P0CR register controls the direction of the Port 0 pins. Upon reset, the P0CR register bits are cleared, configuring all Port 0 pins as inputs.

During external memory accesses, Port 0 pins are automatically configured as D[0:7] or AD[0:7], with the P0CR register bits all cleared.

If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected (D[0:7]). If the BUSMD pin is driven high upon reset, Multiplexed Bus mode is selected (AD[0:7]).

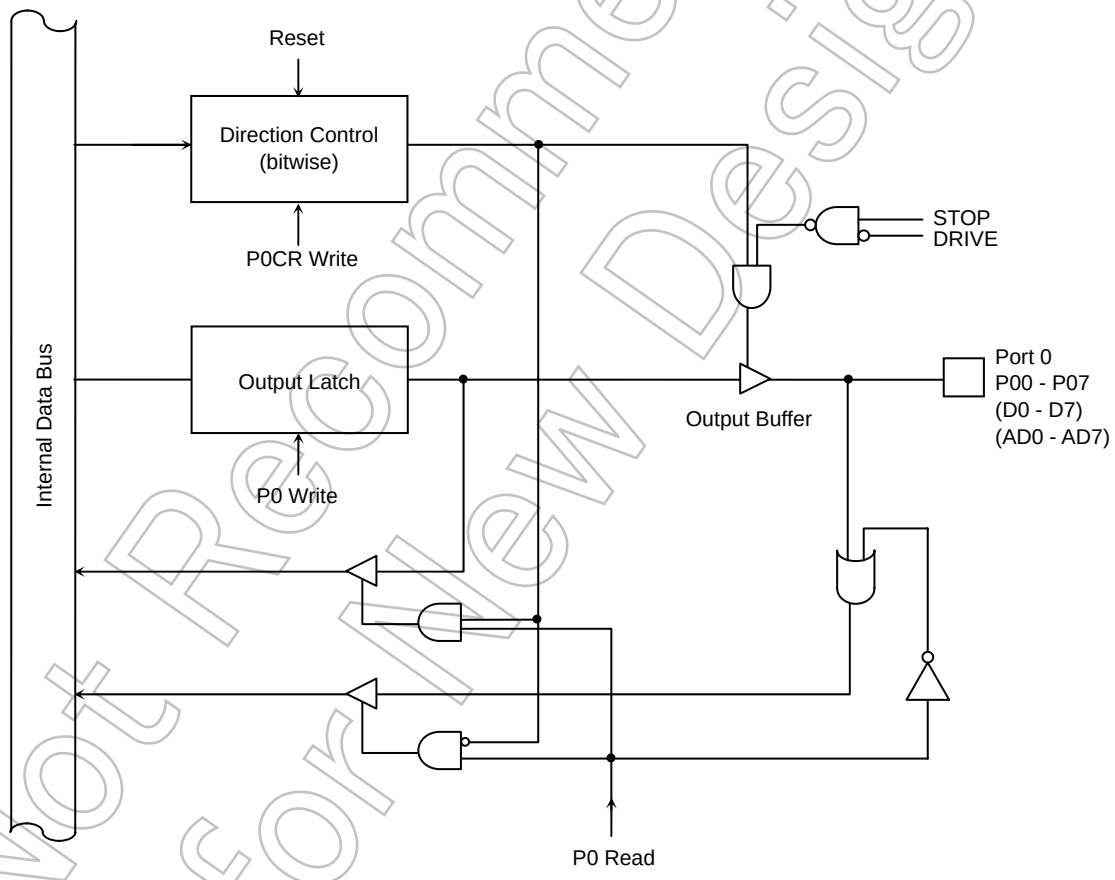


Figure 7.1 Port 0 (P00 - P07)

Note: The above diagram does not depict the address/data bus function.

Port 0 Register

		7	6	5	4	3	2	1	0
P0	Bit Symbol	P07	P06	P05	P04	P03	P02	P01	P00
(0xFFFF_F003)	Read/Write	R/W							
	Reset Value	Input mode (The output latch is cleared to 0.)							

Port 0 Control Register

		7	6	5	4	3	2	1	0
P0CR	Bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
(0xFFFF_F001)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: In, 1: Out (Functions as D7-D0 or AD7-AD0 during external memory accesses, with all bits cleared.)							

Figure 7.2 Port 0 Registers

7.2 Port 1 (P10 - P17)

Eight Port 1 pins can be individually programmed to function as discrete general-purpose I/O pins, the D[8:15] bits of the data bus, the AD[8:15] bits of the address/data bus, or the A[8:15] bits of the address bus. The P1CR and P1FC registers select the direction and function of the Port 1 pins. Upon reset, the output latch (P1) bits are cleared to all 0s, and the P1CR and P1FC register bits are cleared to all 0s, configuring all Port 1 pins as input port pins. For external memory accesses, Port 1 pins must be configured as the address bus or address/data bus through the programming of the P1CR and P1FC.

If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected (D[8:15]). If the BUSMD pin is driven high upon reset, Multiplexed Bus mode is selected (AD[8:15] or A[8:15]).

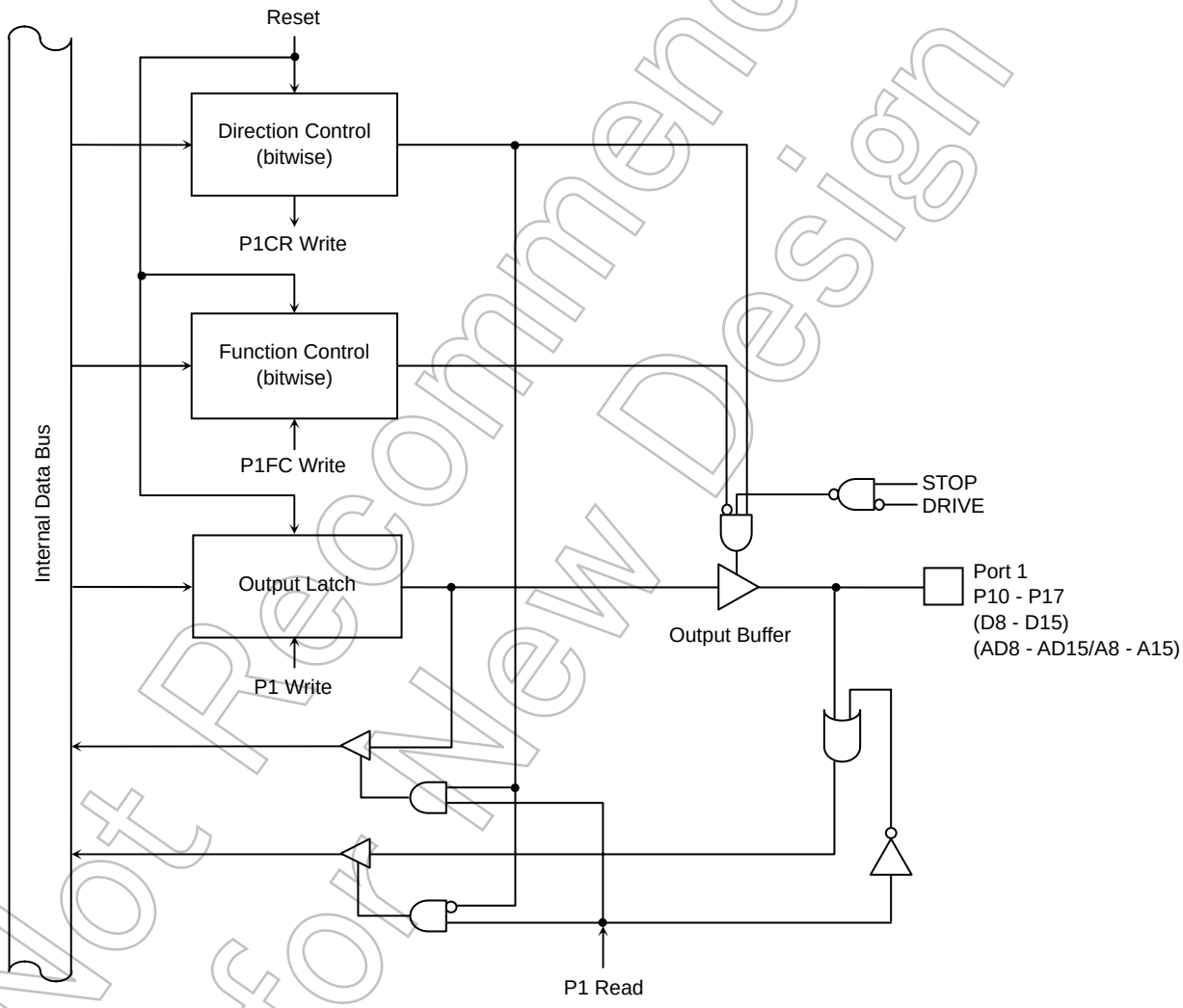


Figure 7.3 Port 1 (P10 - P17)

Note: The above diagram does not depict the address/data bus function.

Port 1 Register

		7	6	5	4	3	2	1	0
P1	Bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
(0xFFFF_F002)	Read/Write	R/W							
	Reset Value	Input mode (The output latch is cleared to 0.)							

Port 1 Control Register

		7	6	5	4	3	2	1	0
P1CR	Bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
(0xFFFF_F007)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	Refer to P1FC.							

Port 1 Function Register

		7	6	5	4	3	2	1	0
P1FC	Bit Symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
(0xFFFF_F006)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	P1FC/P1CR = 00: Input port, 01: Output port, 10: D15-D8 or AD15-AD8, 11: A15-A8							

→ Port 1 Function Settings

	P1CR <P1xC>	P1FC<P1xF>	
		0	1
Separate Bus mode (BUSMD = L)	0	Input port	Data bus (D15-D8)
	1	Output port	Address bus (A15-A8)
Multiplexed Bus mode (BUSMD = H)	0	Input port	Address/Data bus (AD15-AD8)
	1	Output port	Address bus (A15-A8)

Figure 7.4 Port 1 Registers

7.3 Port 2 (P20 - P27)

Eight Port 2 pins can be individually programmed to function as discrete general-purpose I/O pins, the A[0:7] bits of the address bus, or the A[16:23] bits of the address bus. The P2CR and P2FC registers select the direction and function of the Port 2 pins. Upon reset, the output latch (P2) bits are cleared to all 0s, and the P2CR and P2FC register bits are cleared to all 0s, configuring all Port 2 pins as input port pins. For external memory accesses, Port 2 pins must be configured as the address bus through the programming of the P2CR and P2FC.

If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected (A[16:23]). If the BUSMD pin is driven high upon reset, Multiplexed Bus mode is selected (A[0:7] or A[16:23]).

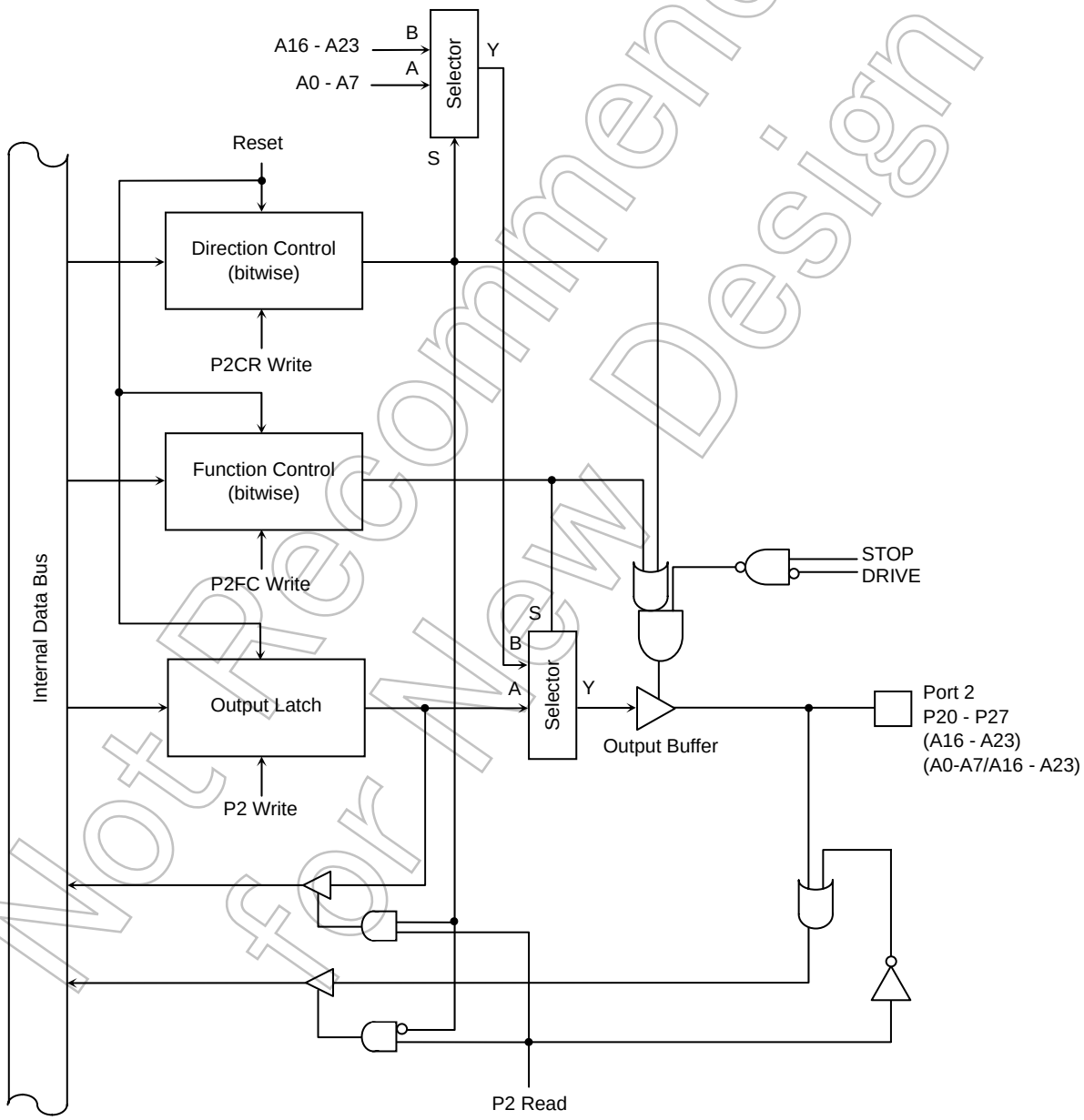


Figure 7.5 Port 2 (P20 - P27)

Port 2 Register

		7	6	5	4	3	2	1	0
P2	Bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
(0xFFFF_F011)	Read/Write	R/W							
	Reset Value	Input mode (The output latch is cleared to 0.)							

Port 2 Control Register

		7	6	5	4	3	2	1	0
P2CR	Bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
(0xFFFF_F017)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	Refer to P2FC.							

Port 2 Function Register

		7	6	5	4	3	2	1	0
P2FC	Bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
(0xFFFF_F016)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	P2FC/P2CR = 00: Input port, 01: Output port, 10: A7-A0, 11: A23-A16							

→ Port 2 Function Settings

	P2CR <P2xC>	P2FC<P2xF> 0	P2FC<P2xF> 1
Separate Bus mode (BUSMD = L)	0	Input port	Address bus (A7-A0)
	1	Output port	Address bus (A23-A16)
Multiplexed Bus mode (BUSMD = H)	0	Input port	Address bus (A7-A0)
	1	Output port	Address bus (A23-A16)

Figure 7.6 Port 2 Registers

7.4 Port 3 (P30 - P37)

Eight Port 3 pins can be individually programmed to function as either discrete general-purpose I/O pins or CPU control/status pins. In either case, P30 and P31 are output-only pins. The P3CR and P3FC registers select the direction and function of the Port 3 pins.

Upon reset, the P30 and P31 output latch bits are set to 1 and the P32-P36 output latch bits are set to 1 if the RSTPUP pin is high, or cleared to 0 if the RSTPUP pin is low. If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected, causing the P37 output latch bit to be set to 1. If the BUSMD pin is driven high upon reset, Multiplexed Bus mode is selected, causing the P37 output latch bit to be cleared to 0. Bits 2 to 6 of the P3CR are cleared to 0 upon reset (bits 0 and 1 are not used). Bit 7 of the P3CR is cleared to 0 in Separate Bus mode, or set to 1 in Multiplexed Bus mode. All bits of the P3FC register are cleared upon reset, configuring P30 and P31 as output port pins (high), P32-P36 as input port pins with pull-up enabled (if the RSTPUP is high) or disabled (if the RSTPUP is low), and P37 as an input port pin (in Separate Bus mode) or output port pin (in Multiplexed Bus mode).

When P30 is configured as $\overline{RD}$ (P3FC.P30F = 1), the Read Strobe signal is activated when external address space is accessed. Likewise, when P31 is configured as $\overline{WR}$ (P3FC.P31F = 1), the Write Strobe signal is activated when external address space is accessed.

While $\overline{BUSAK}$ is asserted, the internal pull-up resistors for P32 and P36 are enabled, if the P3FC.P3xF bit is set to 1.

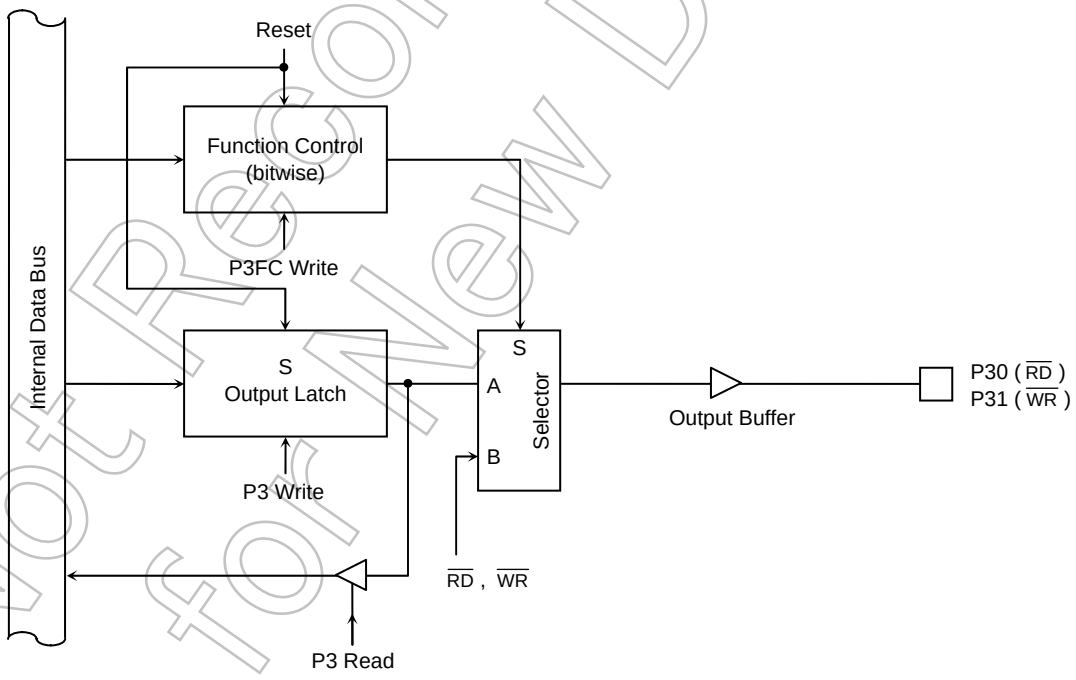


Figure 7.7 Port 3 (P30, P31)

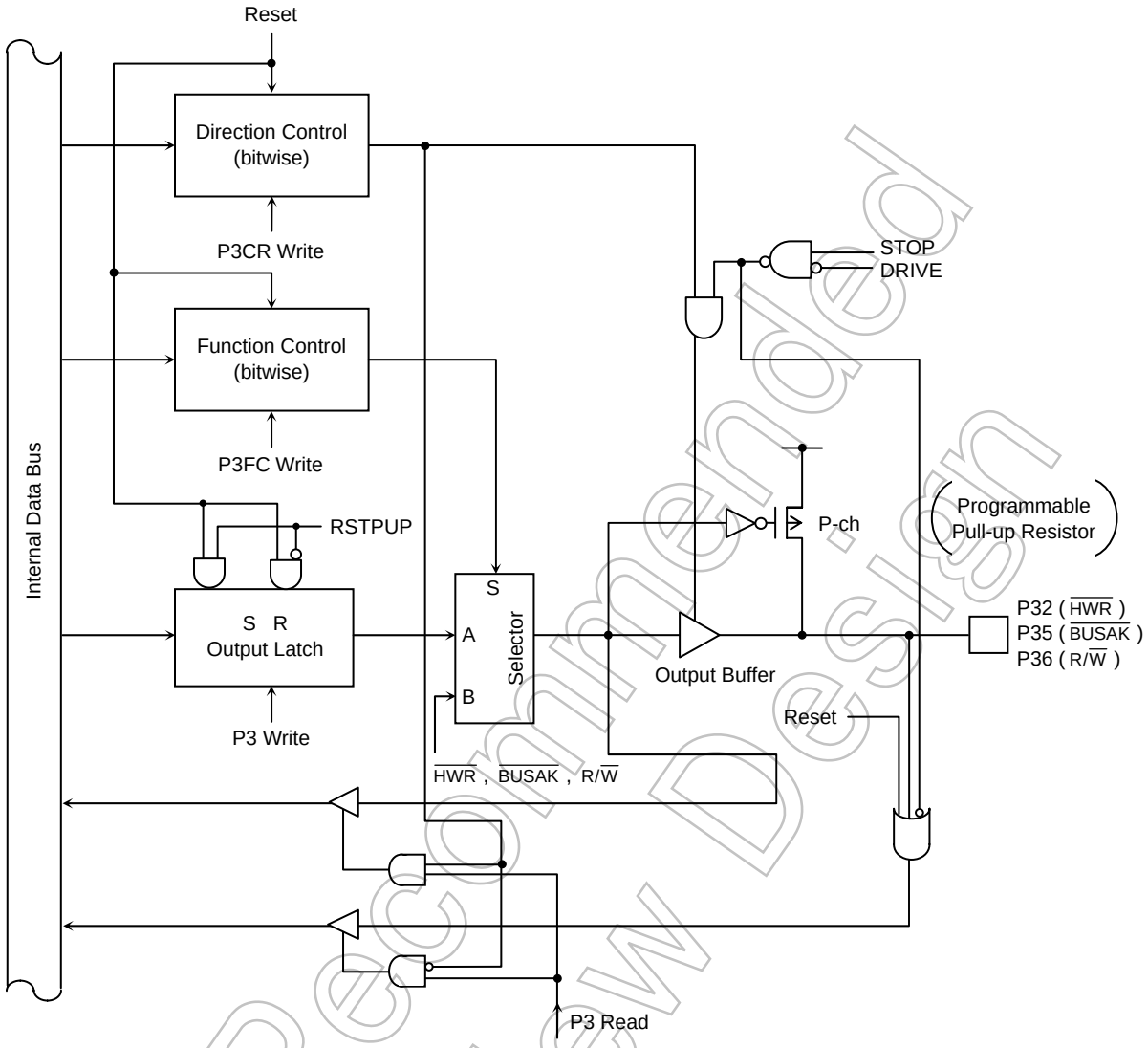


Figure 7.8 Port 3 (P32, P35, P36)

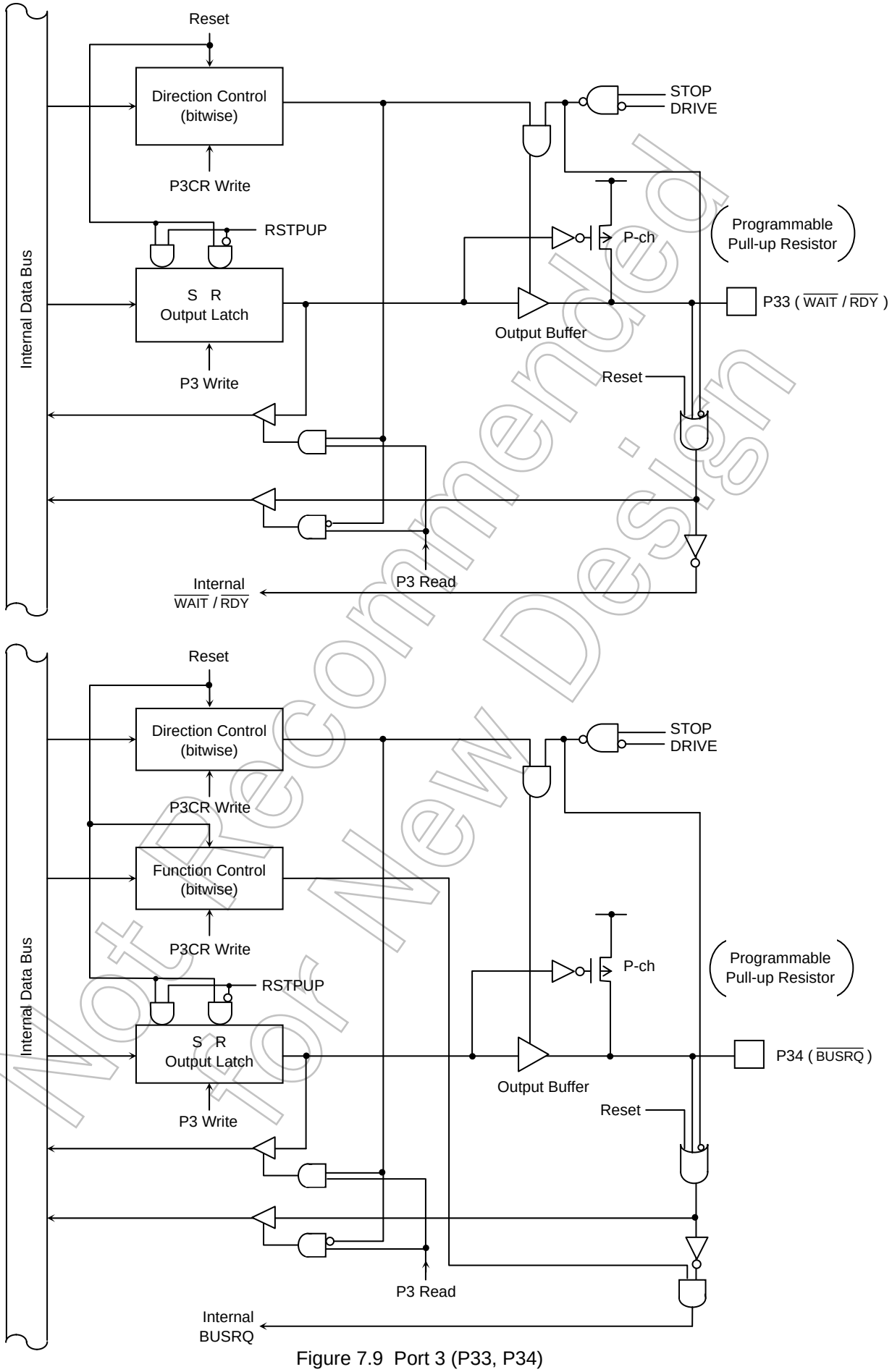


Figure 7.9 Port 3 (P33, P34)

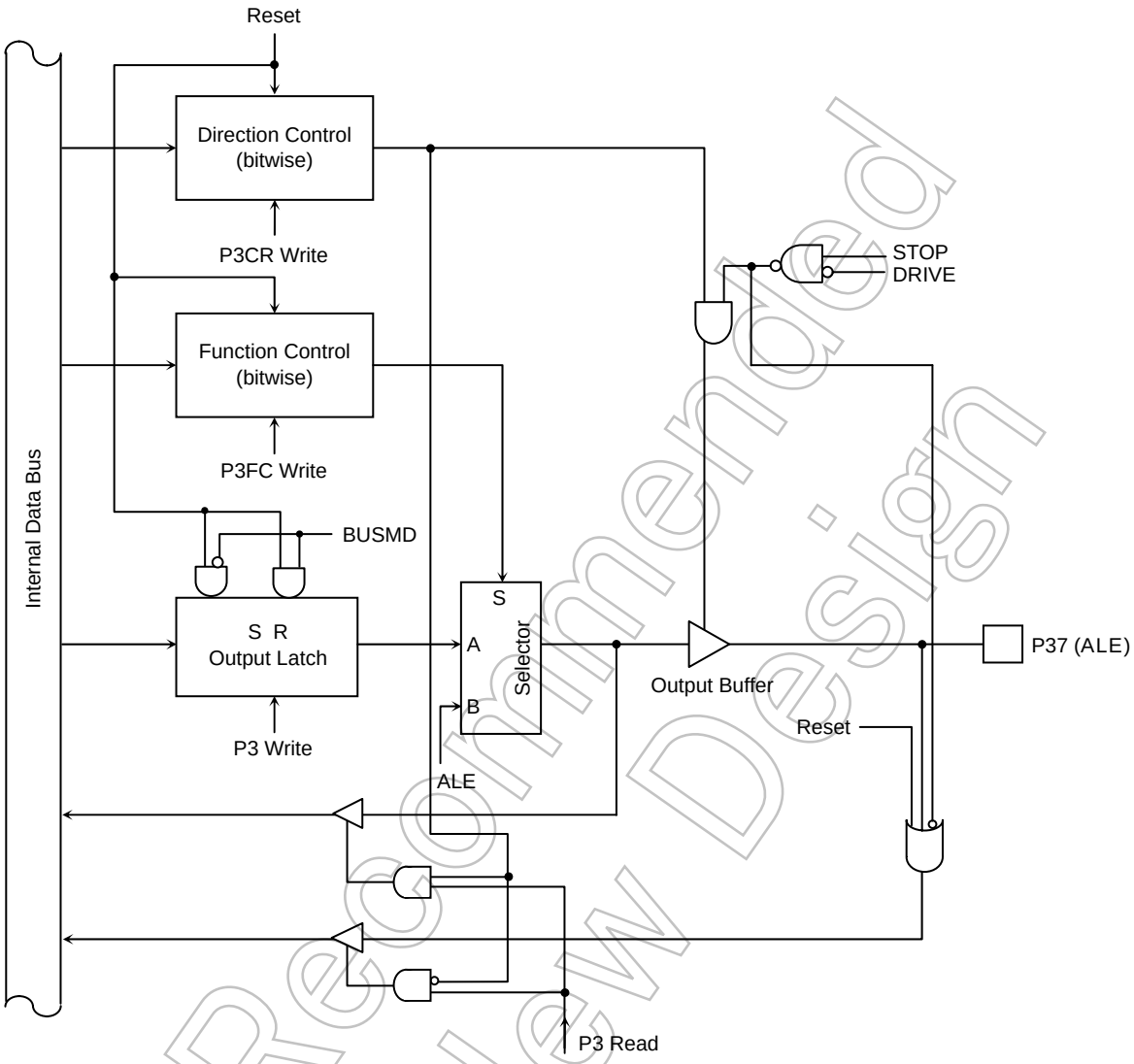


Figure 7.10 Port 3 (P37)

Port 3 Register

P3 (0xFFFF_F01B)

	7	6	5	4	3	2	1	0
Bit Symbol	P37	P36	P35	P34	P33	P32	P31	P30
Read/Write	R/W							
Reset Value	Depends on the bus mode.	Input mode					Output mode	
RSTPUP = 1		1 (Pull-UP)	1 (Pull-UP)	1 (Pull-UP)	1 (Pull-UP)	1 (Pull-UP)	1	1
RSTPUP = 0		0	0	0	0	0		

Port 3 Control Register

P3CR (0xFFFF_F019)

	7	6	5	4	3	2	1	0
Bit Symbol	P37C	P36C	P35C	P34C	P33C	P32C	—	—
Read/Write	W							
Reset Value	Depends on the bus mode.	0	0	0	0	0		
Function		0: Input, 1: Output						

Port 3 Function Register

P3FC (0xFFFF_F018)

	7	6	5	4	3	2	1	0
Bit Symbol	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	0: PORT 1: ALE	0: PORT 1: R/$\overline{W}$	0: PORT 1: $\overline{BUSAK}$	0: PORT 1: $\overline{BUSRQ}$	0: PORT/ $\overline{WAIT}$ 1: PORT/ $\overline{RDY}$	0: PORT 1: $\overline{HWR}$	0: PORT 1: $\overline{WR}$	0: PORT 1: $\overline{RD}$

$\overline{BUSRQ}$ Settings

P3FC<P34F>	1
P3CR<P34C>	0

$\overline{BUSAK}$ Settings

P3FC<P35F>	1
P3CR<P35C>	1

R/$\overline{W}$ Settings

P3FC<P36F>	1
P3CR<P36C>	1

P37 (ALE) Function Settings

	P3CR <P37C>	P3FC<P37F> 0	P3FC<P37F> 1
Separate Bus mode (BUSMD = L)	0	Input port	
	1	Output port	
Multiplexed Bus mode (BUSMD = H)	0	Input	—
	1	Output	ALE output

Separate Bus mode: Upon reset, configured as input port

Multiplexed Bus mode: Upon reset, configured as output port (Outputs a 0.)

P30 ($\overline{RD}$) Function Settings

	<P30> 0	<P30> 1
<P30F>	0	1
0	Output a 0.	Output a 1.
1	Assert $\overline{RD}$ only during external accesses.	

P31 ($\overline{WR}$) Function Settings

	<P31> 0	<P31> 1
<P31F>	0	1
0	Output a 0.	Output a 1.
1	Assert $\overline{WR}$ only during external accesses.	

$\overline{HWR}$ Settings

P3FC<P32F>	1
P3CR<P32C>	1

$\overline{WAIT}$ / $\overline{RDY}$ Settings

	<P33F> 0	<P33F> 1
<P33C>="0"	$\overline{WAIT}$	$\overline{RDY}$

Figure 7.11 Port 3 Registers

7.5 Port 4 (P40 - P44)

P40-P43 can be individually programmed to function as either discrete general-purpose I/O pins or programmable chip select ($\overline{CS0}$ - $\overline{CS3}$) pins. P44 can be programmed to function as either a general-purpose I/O pin or a system clock output (SCOUT) pin. The P4CR and P4FC registers select the direction and function of the Port 4 pins.

Upon reset, the P40-P43 output latch bits are set to 1 if the RSTPUP pin is high, or cleared to 0 if the RSTPUP pin is low. The P44 output latch bit is set to 1 regardless of the state of the RSTPUP pin. The P4CR and P4FC register bits are cleared upon reset, configuring P40-P43 as input port pins with pull-up enabled (if the RSTPUP is high) or disabled (if the RSTPUP is low), and P44 as an input port pin with pull-up disabled (regardless of the state of the RSTPUP pin).

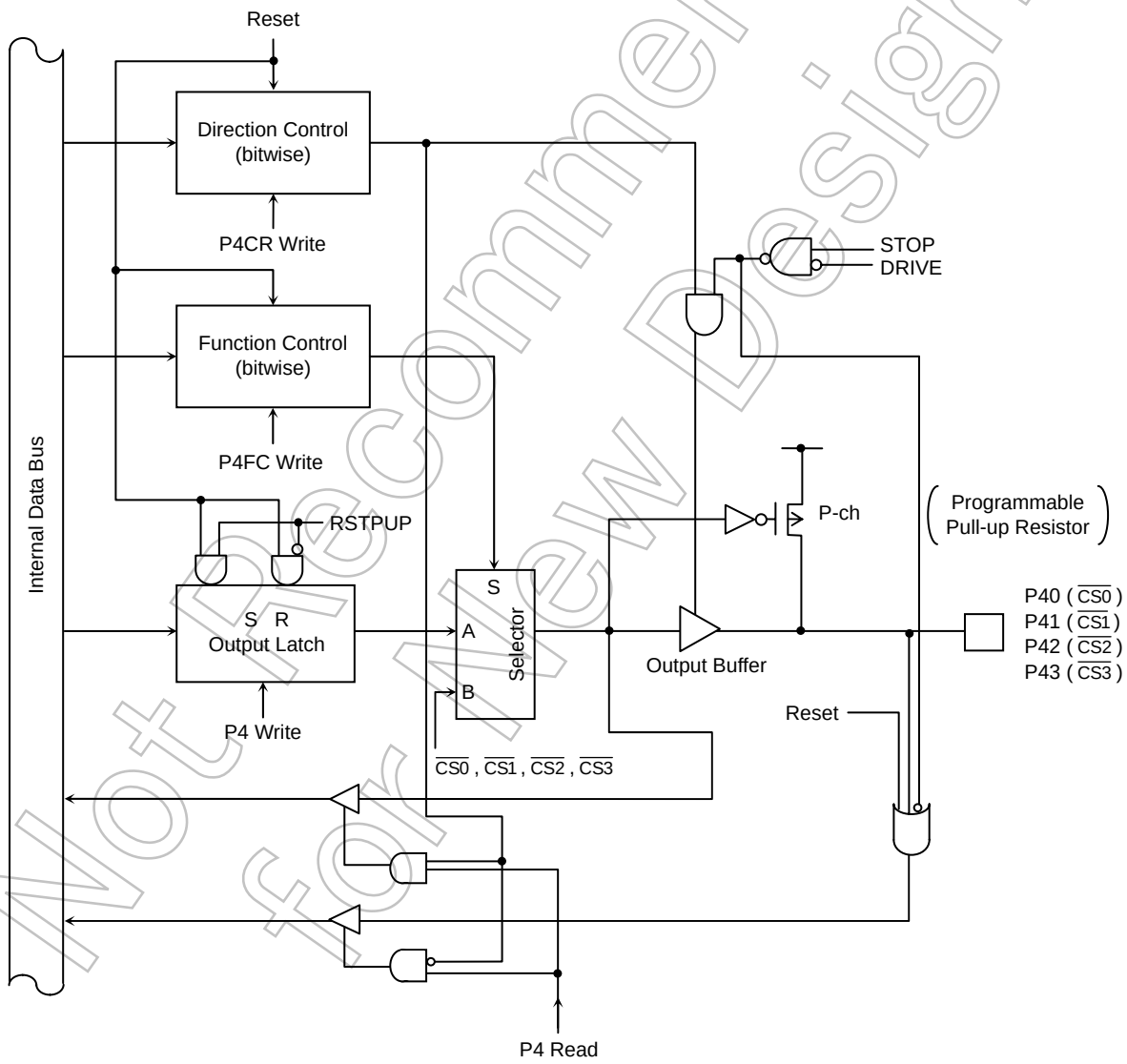


Figure 7.12 Port 4 (P40 - P43)

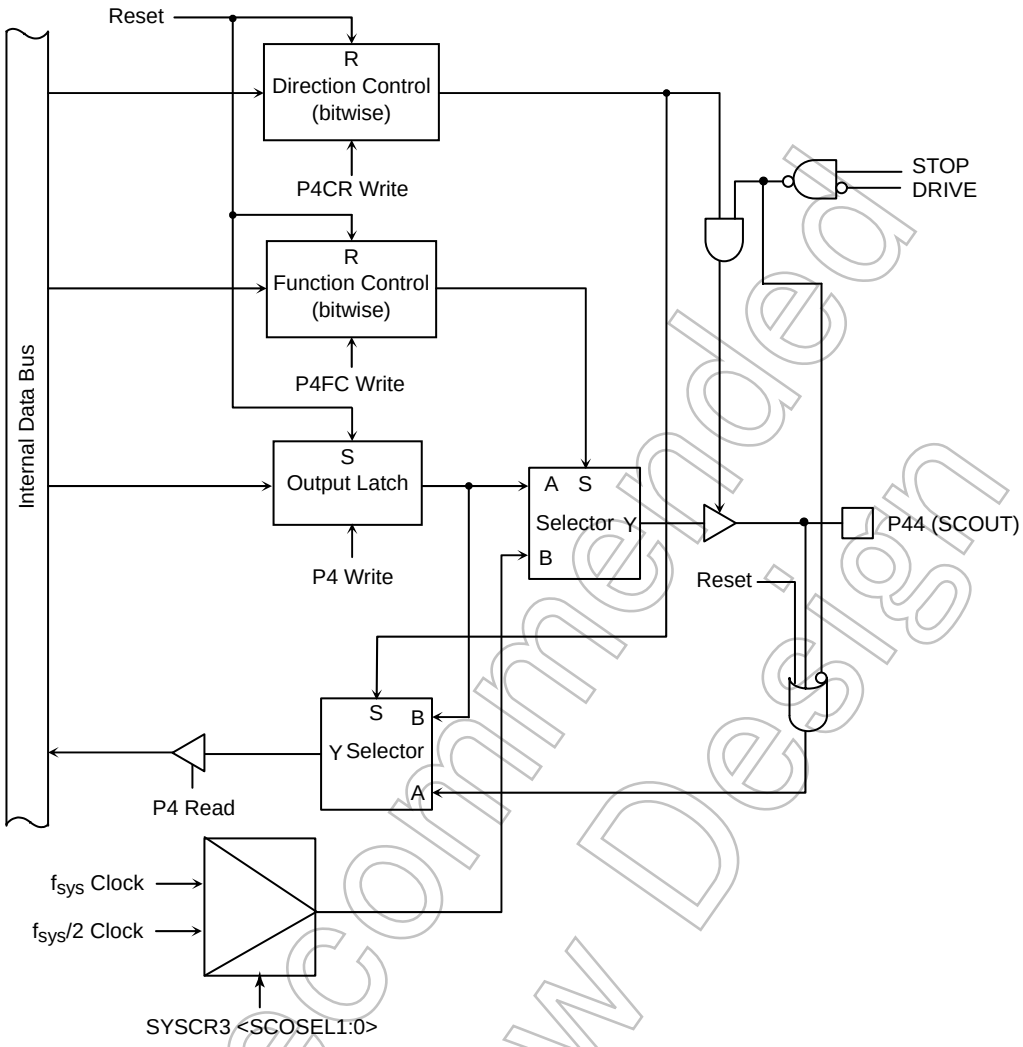


Figure 7.13 Port 4 (P44)

Port 4 Register

		7	6	5	4	3	2	1	0
P4	Bit Symbol	—	—	—	P44	P43	P42	P41	P40
(0xFFFF_F01D)	Read/Write				R/W				
	Reset Value				Input mode				
	RSTPUP=1				1	1 (Pull-Up)	1 (Pull-Up)	1 (Pull-Up)	1 (Pull-Up)
	RSTPUP=0				1	0	0	0	0

Port 4 Control Register

		7	6	5	4	3	2	1	0
P4CR	Bit Symbol	—	—	—	P44C	P43C	P42C	P41C	P40C
(0xFFFF_F023)	Read/Write				W				
	Reset Value				0	0	0	0	0
					0: Input, 1: Output				

Port 4 Function Register

		7	6	5	4	3	2	1	0
P4FC	Bit Symbol	—	—	—	P44F	P43F	P42F	P41F	P40F
(0xFFFF_F022)	Read/Write				W				
	Reset Value				0	0	0	0	0
	Function				0: PORT 1: SCOUT		0: PORT 1: $\overline{CS}$		

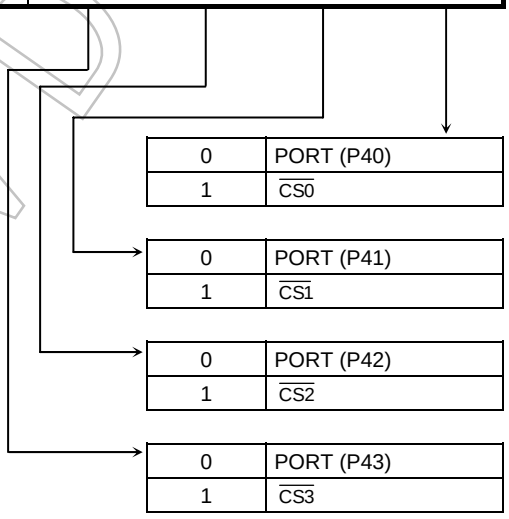


Figure 7.14 Port 4 Registers

7.6 Port 5 (P50 - P57)

Eight Port 5 pins can be individually programmed to function as discrete general-purpose I/O pins or the A[0:7] bits of the address bus. The P5CR and P5FC registers select the direction and function of the Port 5 pins. Upon reset, the output latch (P5) bits are set to all 1s, and the P5CR and P5FC register bits are cleared to all 0s, configuring all Port 5 pins as input port pins.

For external memory accesses, Port 5 pins must be configured as the address bus through the programming of the P5CR and P5FC. Note that Port 5 pins can be used as address bus bits in Separate Bus mode only. If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected.

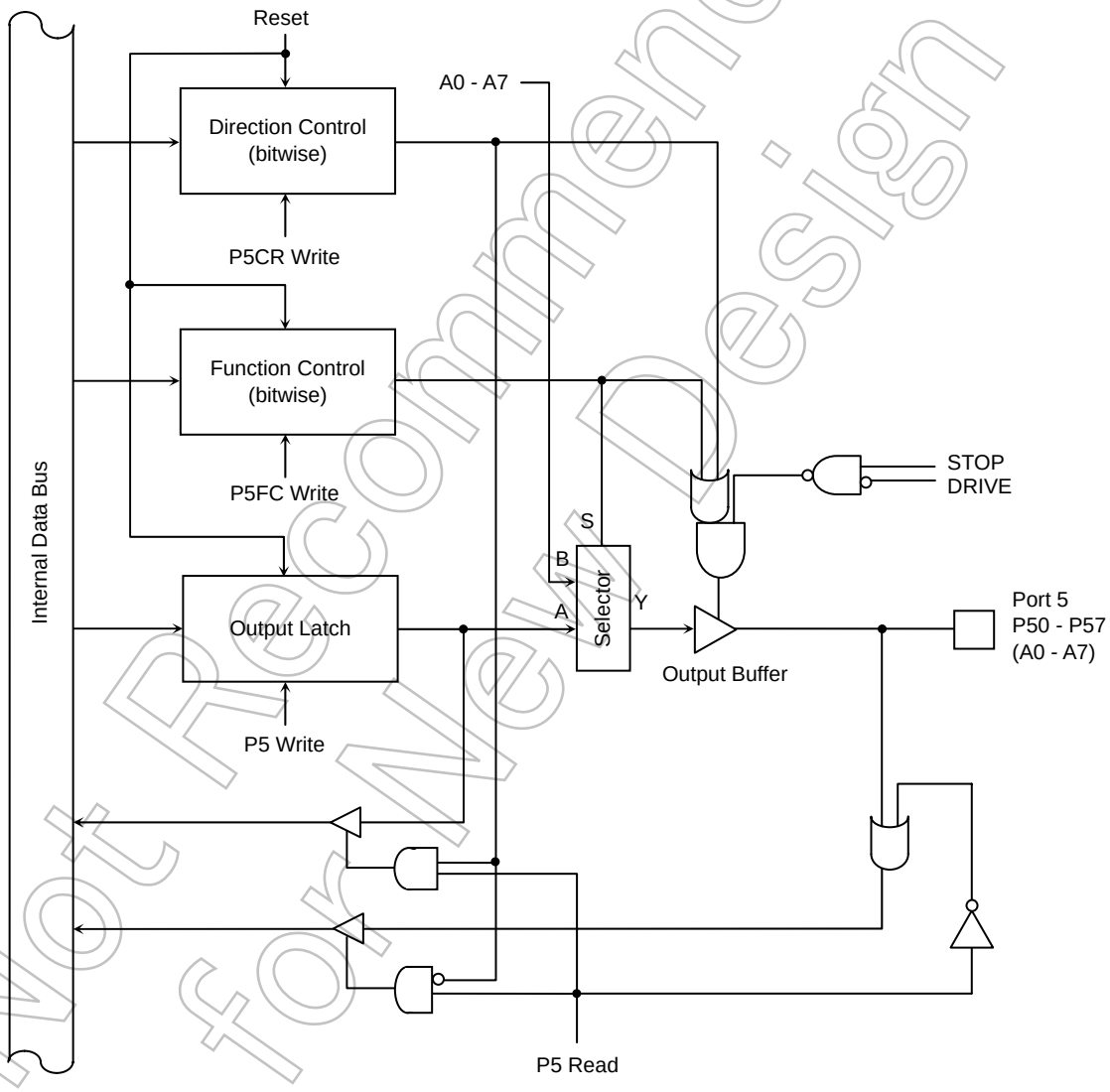


Figure 7.15 Port 5 (P50 - P57)

Port 5 Register

		7	6	5	4	3	2	1	0
P5	Bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
(0xFFFF_F02B)	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port 5 Control Register

		7	6	5	4	3	2	1	0
P5CR	Bit Symbol	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
(0xFFFF_F02F)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	Refer to P5FC.							

Port 5 Function Register

		7	6	5	4	3	2	1	0
P5FC	Bit Symbol	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
(0xFFFF_F02E)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	P5FC/P5CR = 00: Input port, 01: Output port, 10: Input port, 11: A7-A0							

→ Port 5 Function Settings

	P5CR <P5xC>	P5FC<P5xF> 0	P5FC<P5xF> 1
Separate Bus mode (BUSMD = L)	0	Input port	
	1	Output port	Address bus (A7-A0)
Multiplexed Bus mode (BUSMD = H)	0	Input port	
	1	Output port	Address bus (A7-A0)

Figure 7.16 Port 5 Registers

7.7 Port 6 (P60 - P67)

Eight Port 6 pins can be individually programmed to function as discrete general-purpose I/O pins or the A[8:15] bits of the address bus. The P6CR and P6FC registers select the direction and function of the Port 6 pins. Upon reset, the output latch (P6) bits are set to all 1s, and the P6CR and P6FC register bits are cleared to all 0s, configuring all Port 6 pins as input port pins.

For external memory accesses, Port 6 pins must be configured as the address bus through the programming of the P6CR and P6FC. Note that Port 6 pins can be used as address bus bits in Separate Bus mode only. If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected.

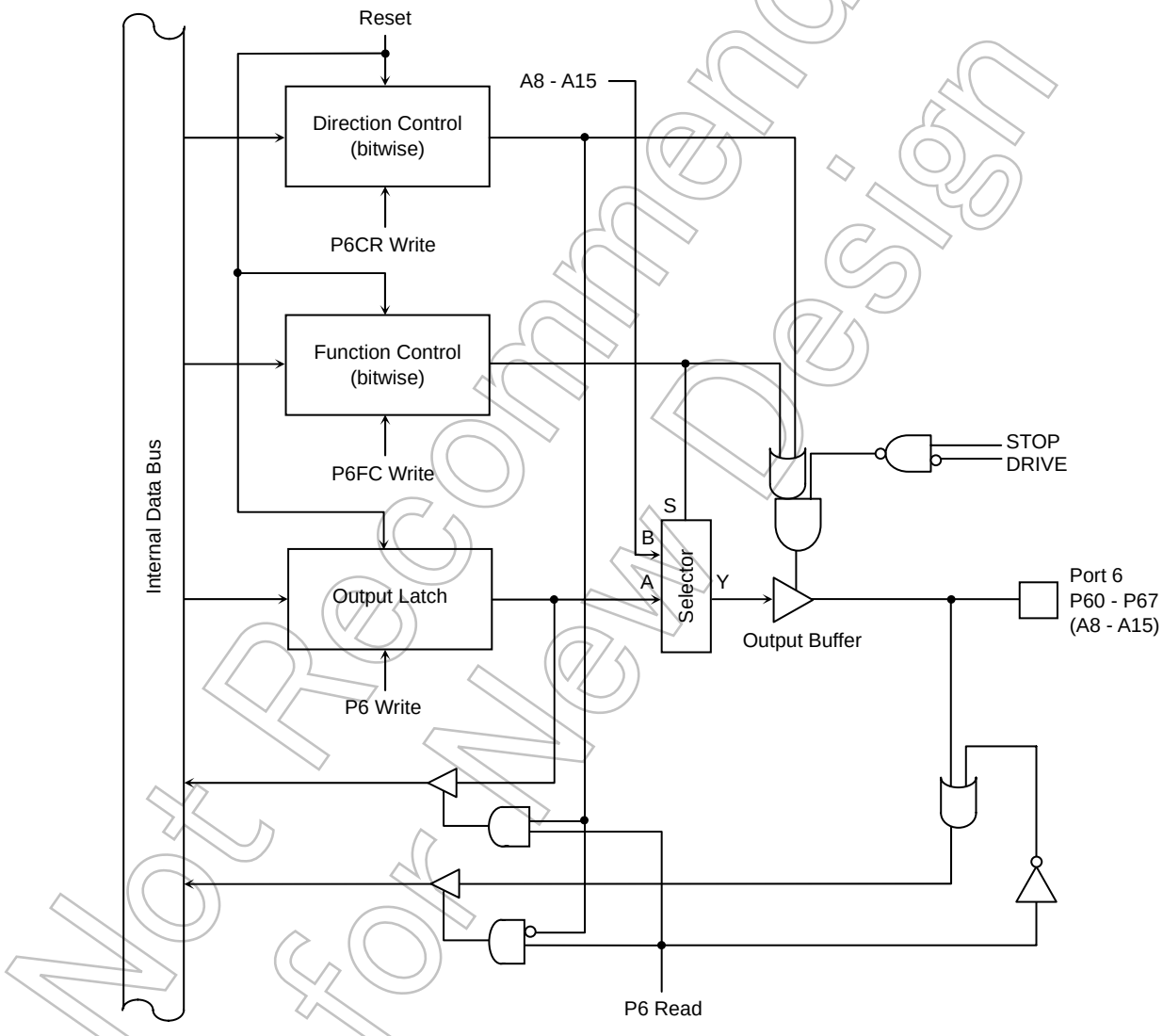


Figure 7.17 Port 6 (P60 - P67)

Port 6 Register

		7	6	5	4	3	2	1	0
P6	Bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
(0xFFFF_F02A)	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port 6 Control Register

		7	6	5	4	3	2	1	0
P6CR	Bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
(0xFFFF_F02D)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	Refer to P6FC.							

Port 6 Function Register

		7	6	5	4	3	2	1	0
P6FC	Bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
(0xFFFF_F02C)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	P6FC/P6CR = 00: Input port, 01: Output port, 10: Input port, 11: A15-A8							

→ Port 6 Function Settings

	P6CR <P6xC>	P6FC<P6xF>	
		0	1
Separate Bus mode (BUSMD = L)	0	Input port	
	1	Output port	Address bus (A15-A8)
Multiplexed Bus mode (BUSMD = H)	0	Input port	
	1	Output port	Address bus (A15-A8)

Figure 7.18 Port 6 Registers

7.8 Port 7 (P70 - 77), Port 8 (P80 - 87) and Port 9 (P90 - 97)

Port 7-9 pins are input-only pins shared with the analog input pins of the A/D Converter (ADC).

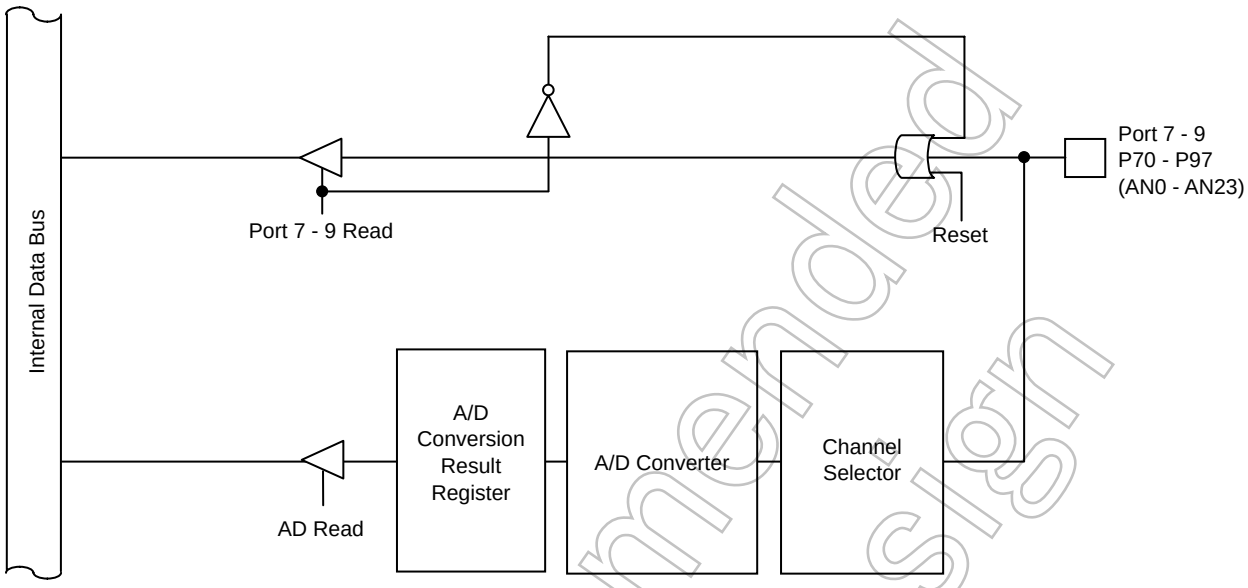


Figure 7.19 Port 7-9 (P70 - 77, P80 - 87, P90 - 97)

Port 7 Register

		7	6	5	4	3	2	1	0
P7	Bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
0xFFFF_F043	Read/Write	R							
	Reset Value	Input mode							

Port 8 Register

		7	6	5	4	3	2	1	0
P8	Bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
0xFFFF_F042	Read/Write	R							
	Reset Value	Input mode							

Port 9 Register

		7	6	5	4	3	2	1	0
P9	Bit Symbol	P97	P96	P95	P94	P93	P92	P91	P90
0xFFFF_F041	Read/Write	R							
	Reset Value	Input mode							

Figure 7.20 Port 7 - 9 Registers

7.9 Port A (PA0 - PA7)

Eight Port A pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PACR register selects the direction of the Port A pins. Upon reset, the PACR register bits are cleared to all 0s, configuring all Port A pins as input port pins. PA0 and PA2 can be programmed as inputs to 8-bit timers. PA1 and PA3-PA7 can be programmed as outputs from 8-bit timers. Setting the PAFC register bits configures the corresponding Port A pins for timer functions. A reset clears the PACR and PAFC register bits, configuring all Port A pins as input port pins.

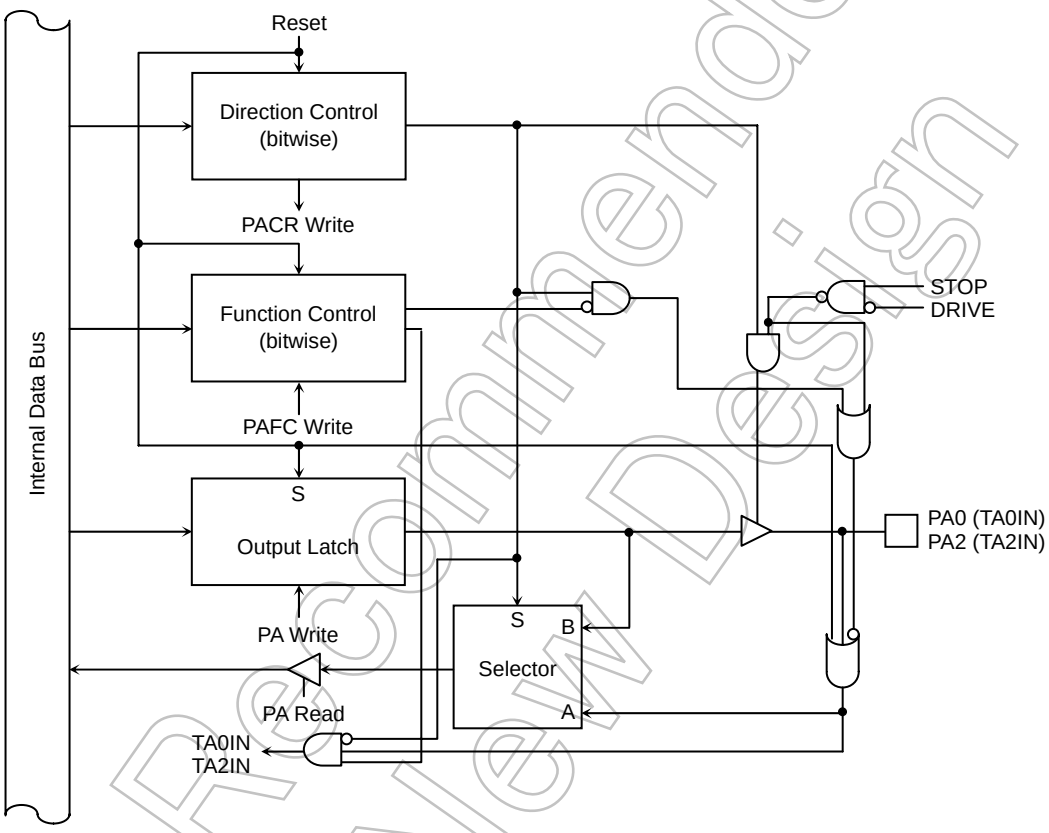


Figure 7.21 Port A (PA0, PA2)

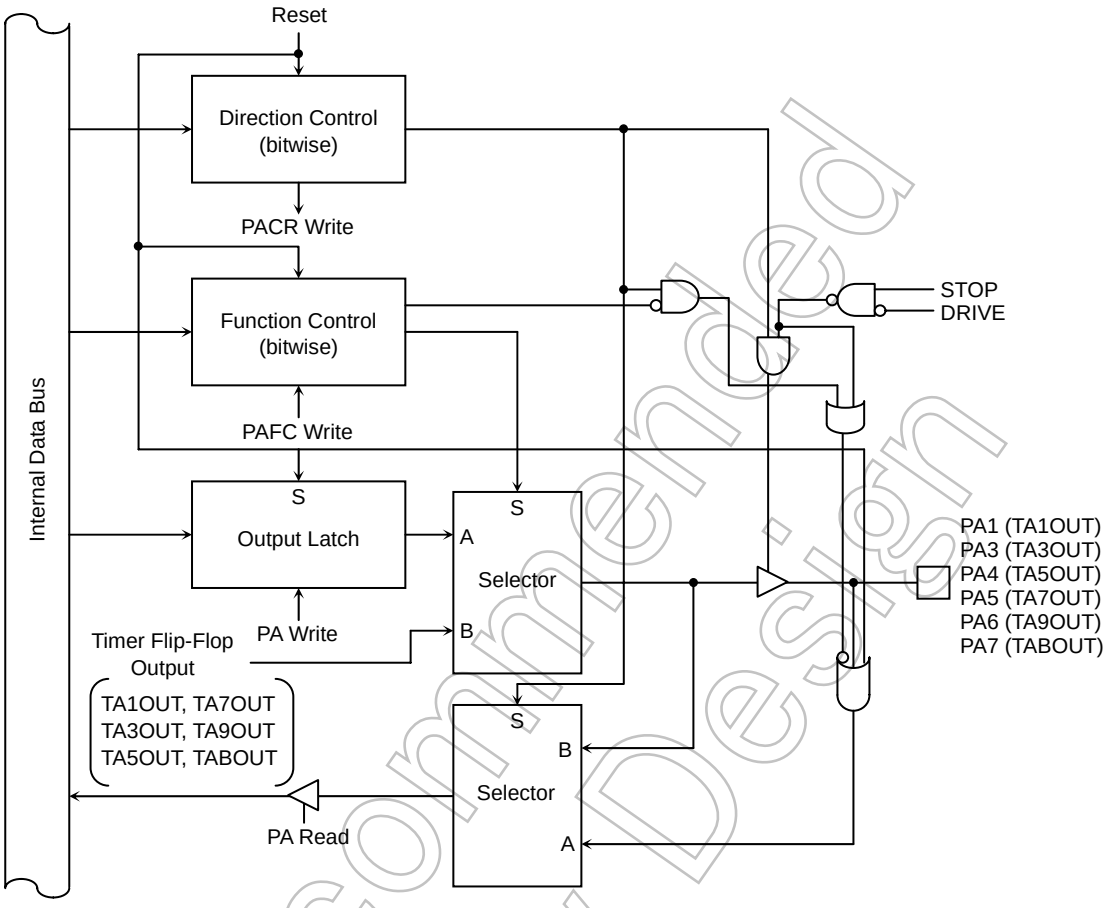


Figure 7.22 Port A (PA1, PA3, PA4, PA5, PA6, PA7)

Port A Register

		7	6	5	4	3	2	1	0
PA	Bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
0xFFFF_F040	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port A Control Register

		7	6	5	4	3	2	1	0
PACR	Bit Symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
0xFFFF_F044	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port A Function Register

		7	6	5	4	3	2	1	0
PAFC	Bit Symbol	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
0xFFFF_F048	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TABOUT	0: PORT 1: TA9OUT	0: PORT 1: TA7OUT	0: PORT 1: TA5OUT	0: PORT 1: TA3OUT	0: PORT 1: TA2IN	0: PORT 1: TA1OUT	0: PORT 1: TA0IN

Figure 7.23 Port A Registers

7.10 Port B (PB0 - PB7)

Eight Port B pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PBCR register selects the direction of the Port B pins. Upon reset, the PBCR register bits are cleared to all 0s, configuring all Port B pins as input port pins. PB0, PB1, PB4 and PB7 can be programmed as outputs from 16-bit timers. PB2, PB3, PB5 and PB6 can be programmed as inputs to 16-bit timers or external interrupt request pins. Setting the PBFC register bits configures the corresponding Port B pins for dedicated functions. A reset clears the PBCR and PBFC register bits, configuring all Port B pins as input port pins.

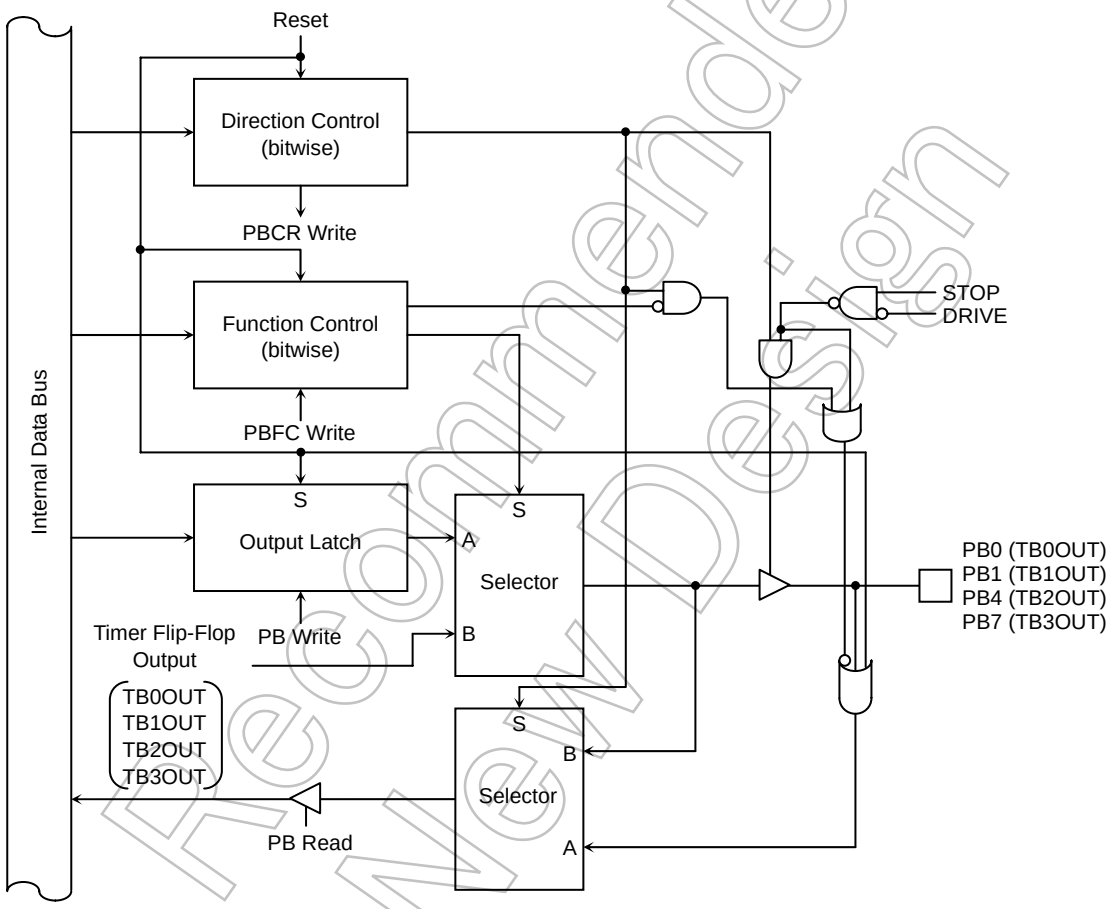


Figure 7.24 Port B (PB0, PB1, PB4, PB7)

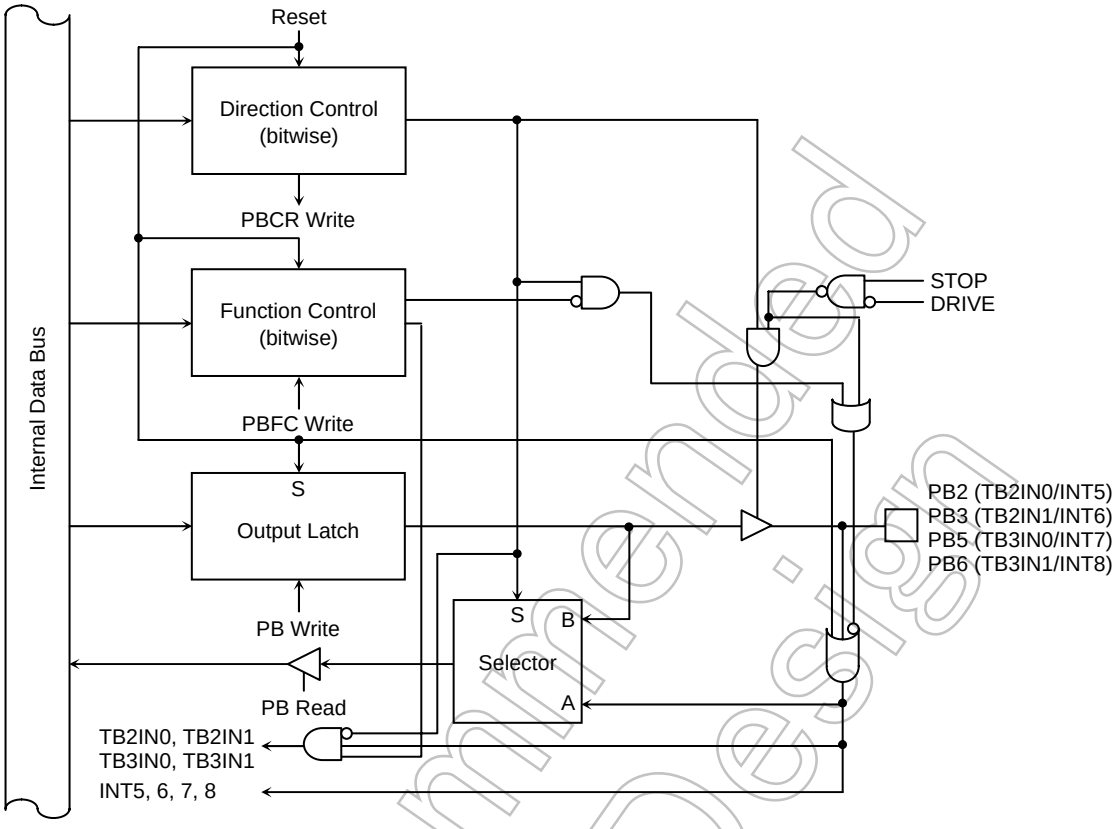


Figure 7.25 Port B (PB2, PB3, PB5, PB6)

Port B Register

		7	6	5	4	3	2	1	0
PB	Bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0xFFFF_F053	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port B Control Register

		7	6	5	4	3	2	1	0
PBCR	Bit Symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
0xFFFF_F057	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port B Function Register

		7	6	5	4	3	2	1	0
PBFC	Bit Symbol	PB7F	PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	PB0F
0xFFFF_F05B	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TB3OUT	0: PORT 1: TB3IN1 INT8	0: PORT 1: TB3IN0 INT7	0: PORT 1: TB2OUT	0: PORT 1: TB2IN1 INT6	0: PORT 1: TB2IN0 INT5	0: PORT 1: TB1OUT	0: PORT 1: TB0OUT

Function	Corresponding Bit in PBFC	Corresponding Bit in PBCR	Port Used
TB0OUT output settings	1	1	PB0
TB1OUT output settings	1	1	PB1
TB2IN0 input settings	1	0	PB2
INT5 input settings	1 (*1)	0	
TB2IN1 input settings	1	0	PB3
INT6 input settings	1 (*1)	0	
TB2OUT output settings	1	1	PB4
TB3IN0 input settings	1	0	PB5
INT7 input settings	1(*1)	0	
TB3IN1 input settings	1	0	PB6
INT8 input settings	1(*1)	0	
TB3OUT output settings	1	1	PB7

***1: This bit must be set when the corresponding interrupt source is used for STOP wake-up signaling with SYSCR.DRVE cleared to 0. Otherwise, the bit need not be set.**

Note: For a port pin assigned two input functions in addition to the port function, the corresponding function modules must be programmed to determine which function is enabled.

Figure 7.26 Port B Registers

7.11 Port C (PC0 - PC7)

Eight Port C pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PCCR register selects the direction of the Port C pins. Upon reset, the PCCR register bits are cleared to all 0s, configuring all Port C pins as input port pins. PC0, PC3 and PC6 can be programmed as SIO data outputs. PC1, PC4 and PC7 can be programmed as SIO data inputs. PC2 and PC5 can be programmed as SIO clock inputs/outputs or CTS inputs. Setting the PCFC register bits configures the corresponding Port C pins for dedicated functions. A reset clears the PCCR and PCFC register bits, configuring all Port C pins as input port pins.

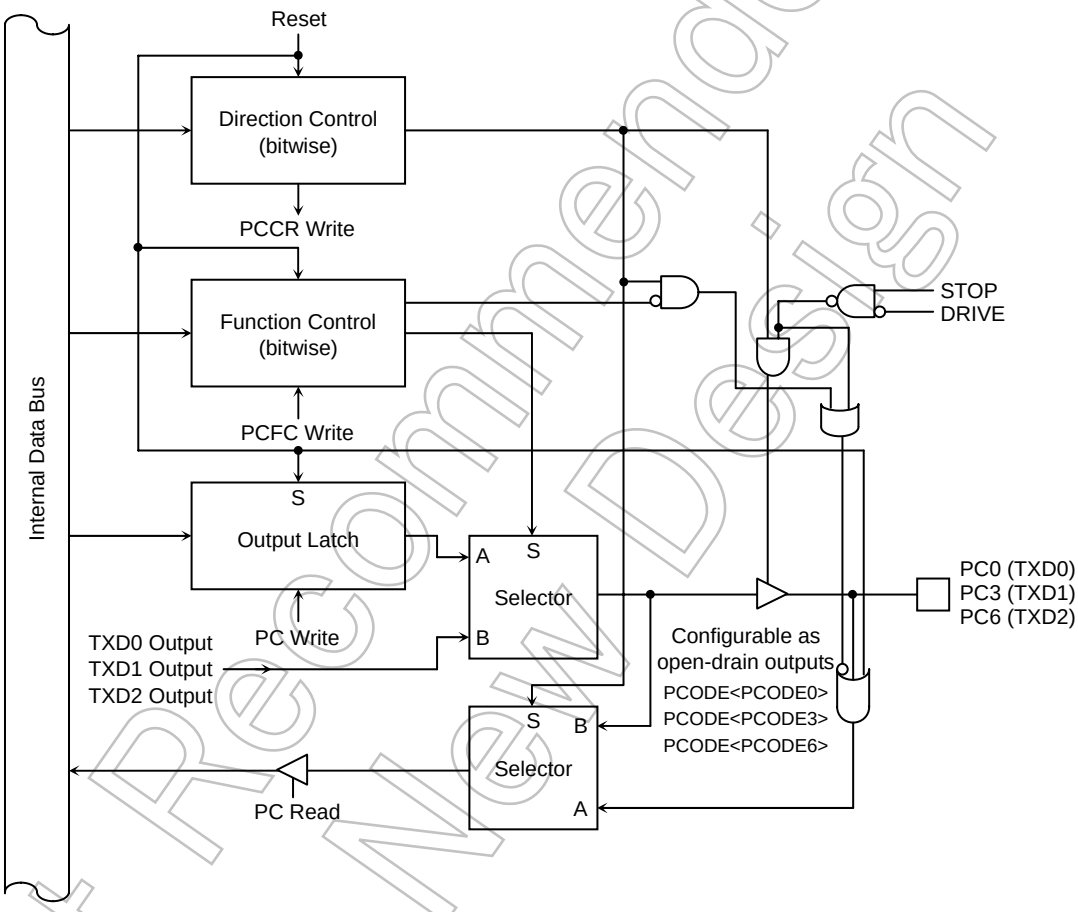


Figure 7.27 Port C (PC0, PC3, PC6)

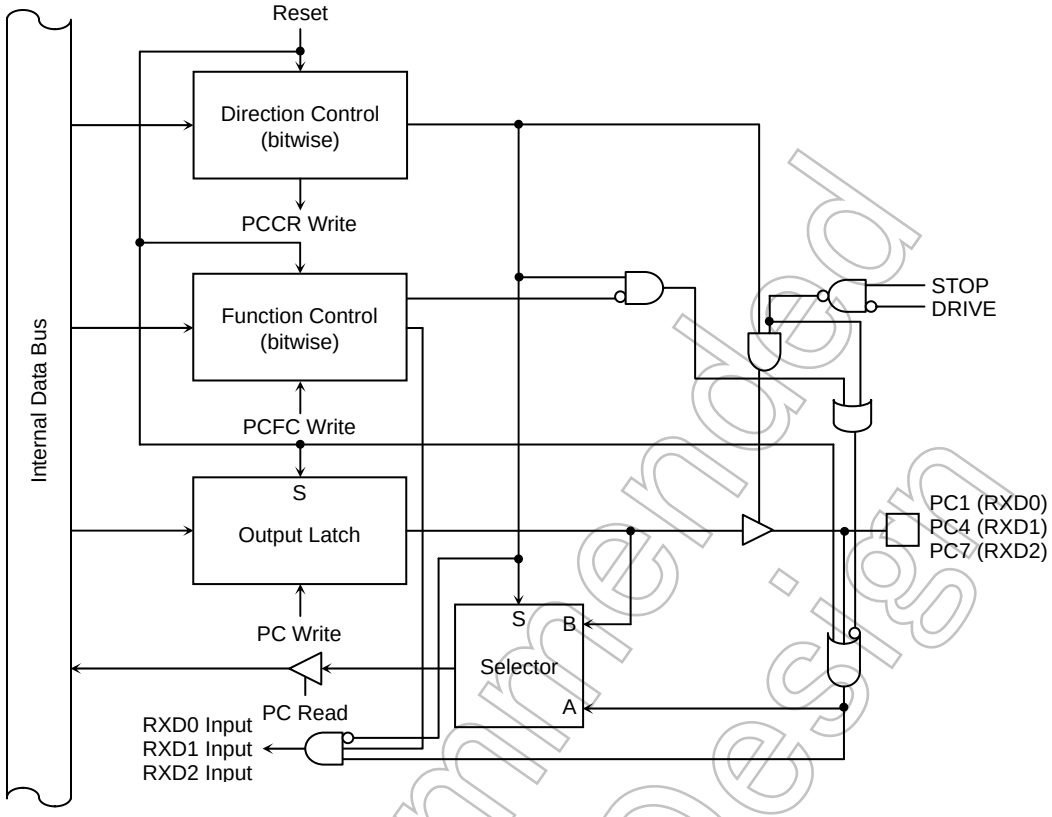


Figure 7.28 Port C (PC1, PC4, PC7)

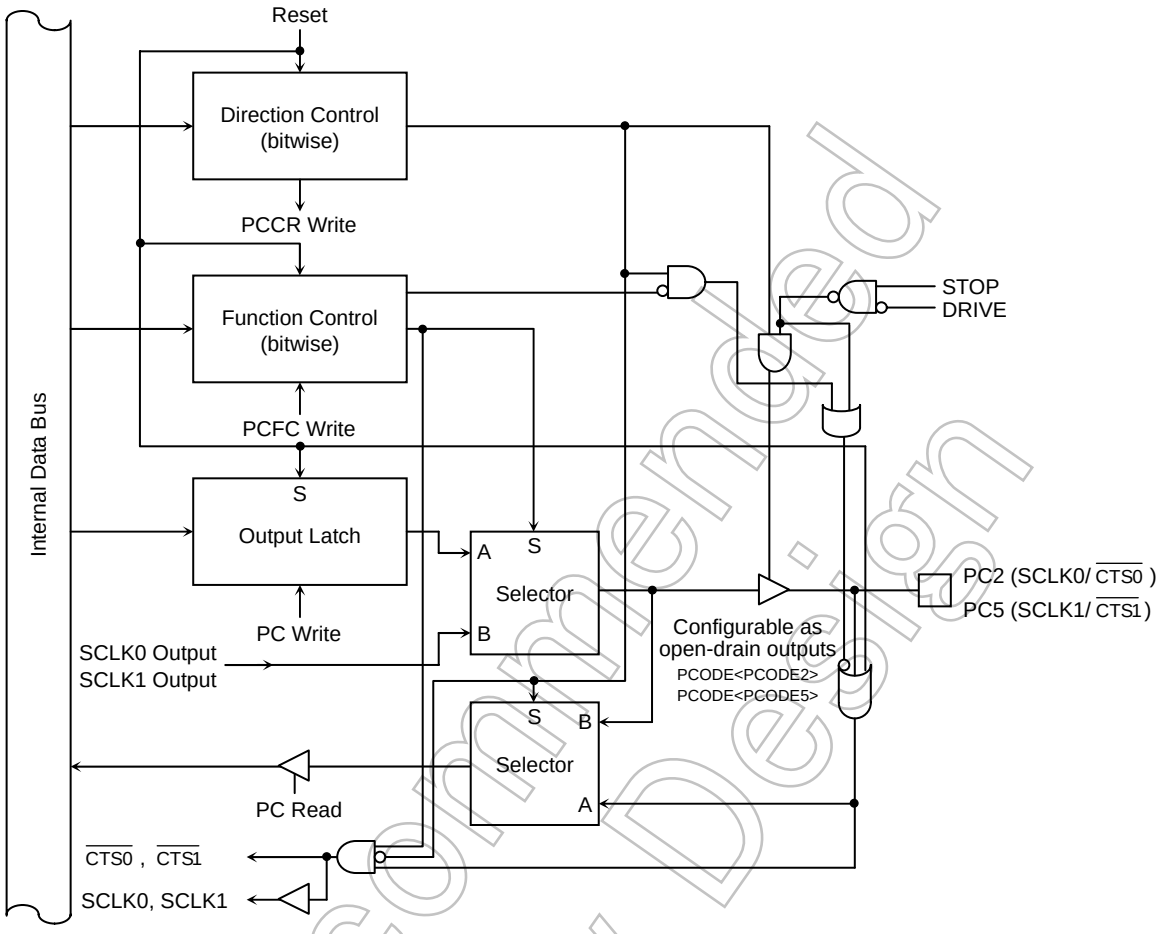


Figure 7.29 Port C (PC2, PC5)

Port C Register

		7	6	5	4	3	2	1	0
PC	Bit Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
0xFFFF_F052	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port C Control Register

		7	6	5	4	3	2	1	0
PCCR	Bit Symbol	PC7C	PC6C	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
0xFFFF_F056	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port C Function Register

		7	6	5	4	3	2	1	0
PCFC	Bit Symbol	PC7F	PC6F	PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
0xFFFF_F05A	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: RXD2	0: PORT 1: TXD2	0: PORT 1: SCLK1 $\overline{CTS1}$	0: PORT 1: RXD1	0: PORT 1: TXD1	0: PORT 1: SCLK0 $\overline{CTS0}$	0: PORT 1: RXD0	0: PORT 1: TXD0

Port C Open-Drain Enable Register

		7	6	5	4	3	2	1	0
PCODE	Bit Symbol	-	PCODE6	PCODE5	-	PCODE3	PCODE2	-	PCODE0
0xFFFF_F05E	Read/Write		W			W			W
	Reset Value		0	0		0	0		0
	Function		0: CMOS 1: Open-drain	0: CMOS 1: Open-drain		0: CMOS 1: Open-drain	0: CMOS 1: Open-drain		0: CMOS 1: Open-drain

Figure 7.30 Port C Registers

7.12 Port D (PD0 - PD7)

Eight Port D pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PDCR register selects the direction of the Port D pins. Upon reset, the PDCR register bits are cleared to all 0s, configuring all Port D pins as input port pins. PD0, PD3 and PD6 can be programmed as SIO clock inputs/outputs or CTS inputs. PD1 and PD4 can be programmed as SIO data outputs. PD2 and PD5 can be programmed as SIO data inputs. PD7 can be programmed as a key-pressed wake-up input. Setting the PDFC register bits configures the corresponding Port D pins for dedicated functions. A reset clears the PDCR and PDFC register bits, configuring all Port D pins as input port pins.

PD7 has an internal pull-up resistor, which is enabled when key input is enabled through the programming of KWUPSTn with the KWUPCNT.KYPE bit set to 1 in the key-pressed wake-up circuit block. For details, refer to Chapter 19. The pull-up resistor is disabled when the PD7 pin is used as a general-purpose I/O pin.

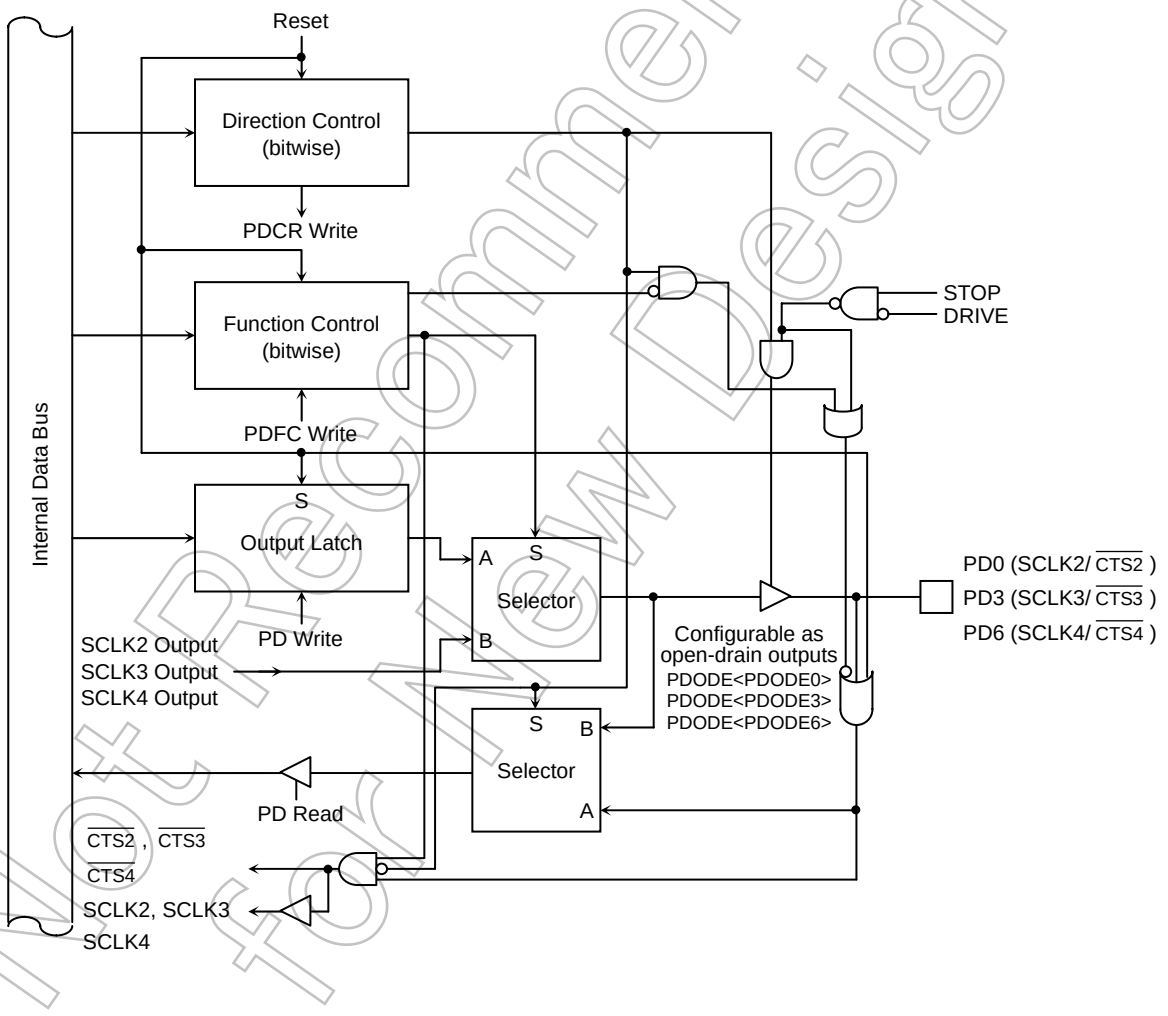


Figure 7.31 Port D (PD0, PD3, PD6)

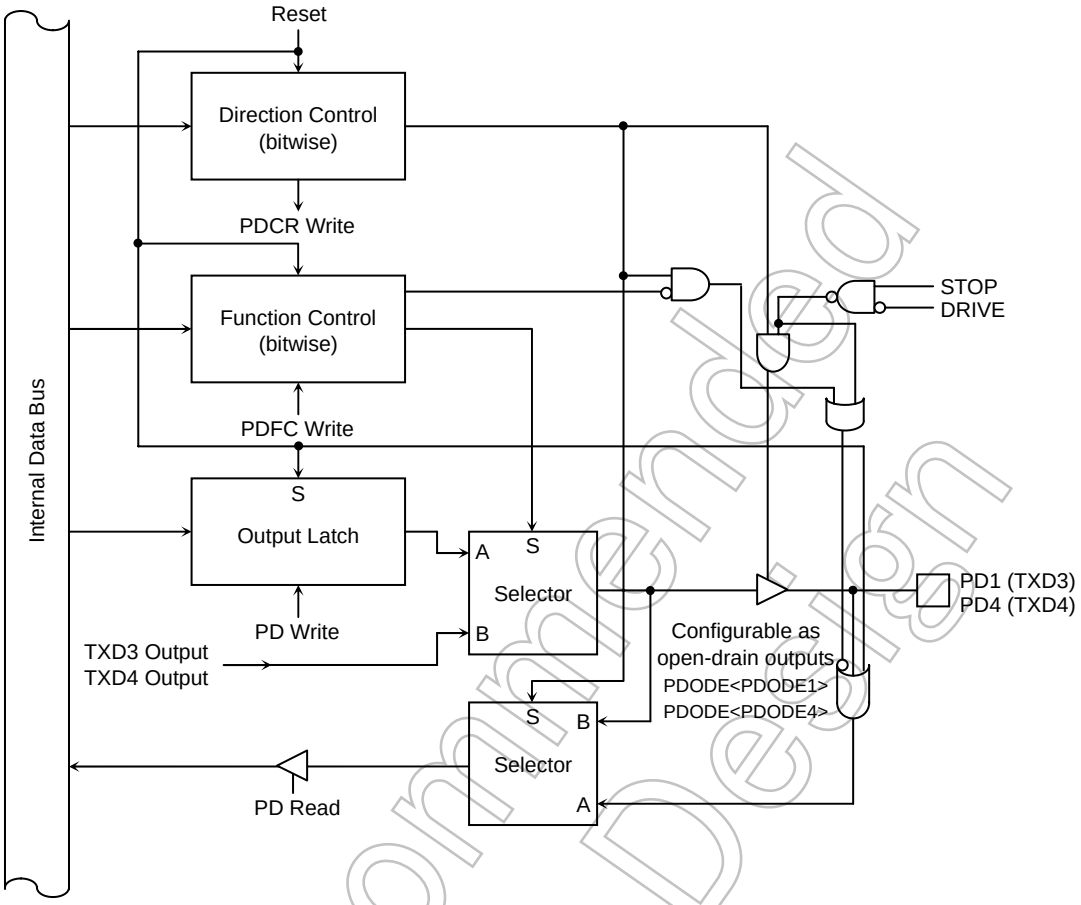


Figure 7.32 Port D (PD1, PD4)

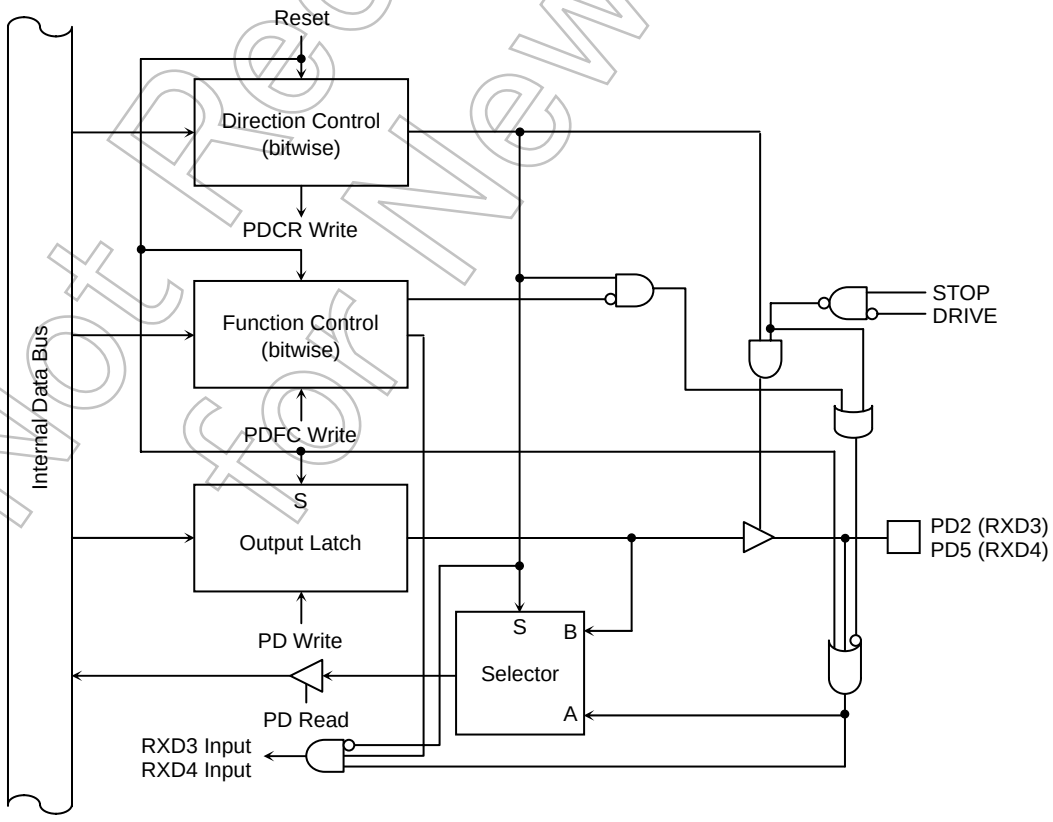


Figure 7.33 Port D (PD2, PD5)

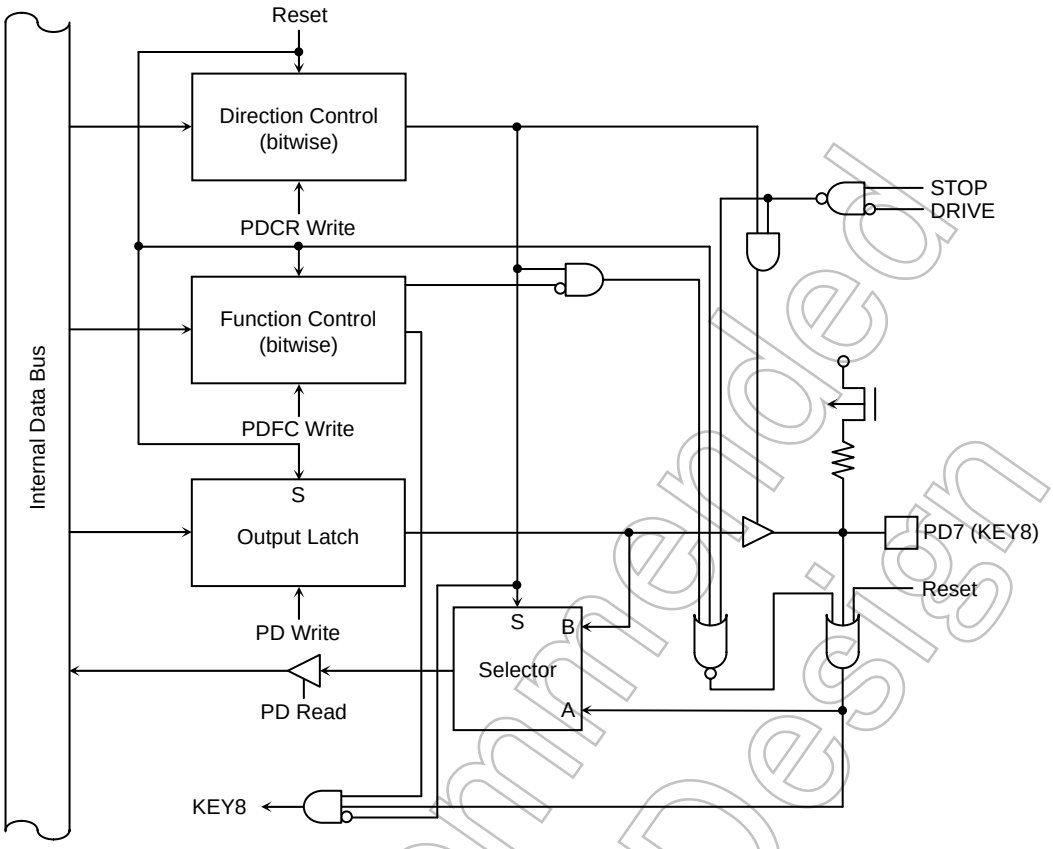


Figure 7.34 Port D (PD7)

Port D Register

		7	6	5	4	3	2	1	0
PD	Bit Symbol	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
0xFFFF_F051	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port D Control Register

		7	6	5	4	3	2	1	0
PDCR	Bit Symbol	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
0xFFFF_F055	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port D Function Register

		7	6	5	4	3	2	1	0
PDFC	Bit Symbol	PD7F	PD6F	PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
0xFFFF_F059	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: KEY8	0: PORT 1: SCLK4 $\overline{\text{CTS4}}$	0: PORT 1: RXD4	0: PORT 1: TXD4	0: PORT 1: SCLK3 $\overline{\text{CTS3}}$	0: PORT 1: RXD3	0: PORT 1: TXD3	0: PORT 1: SCLK2 $\overline{\text{CTS2}}$

Port D Open-Drain Enable Register

		7	6	5	4	3	2	1	0
PDODE	Bit Symbol	—	PDODE6	—	PDODE4	PDODE3	—	PDODE1	PDODE0
0xFFFF_F05D	Read/Write		W		W			W	
	Reset Value		0		0	0		0	0
	Function		0: CMOS 1: Open-drain		0: CMOS 1: Open-drain	0: CMOS 1: Open-drain		0: CMOS 1: Open-drain	0: CMOS 1: Open-drain

Figure 7.35 Port D Registers

7.13 Port E (PE0 - PE7)

Eight Port E pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PECR register selects the direction of the Port E pins. Upon reset, the PECR register bits are cleared to all 0s, configuring all Port E pins as input port pins. PE0 can be programmed as an SIO data output. PE1 can be programmed as an SIO data input. PE2 can be programmed as an SIO clock input/output or CTS input. PE3-PE7 can be programmed as key-pressed wake-up inputs. Setting the PEFC register bits configures the corresponding Port E pins for dedicated functions. A reset clears the PECR and PEFC register bits, configuring all Port E pins as input port pins.

PE3-PE7 have internal pull-up resistors, which are enabled when key input is enabled through the programming of KWUPSTn with the KWUPCNT.KYPE bit set to 1 in the key-pressed wake-up circuit block. For details, refer to Chapter 19. The pull-up resistors are disabled when the PE3-PE7 pins are used as general-purpose I/O pins.

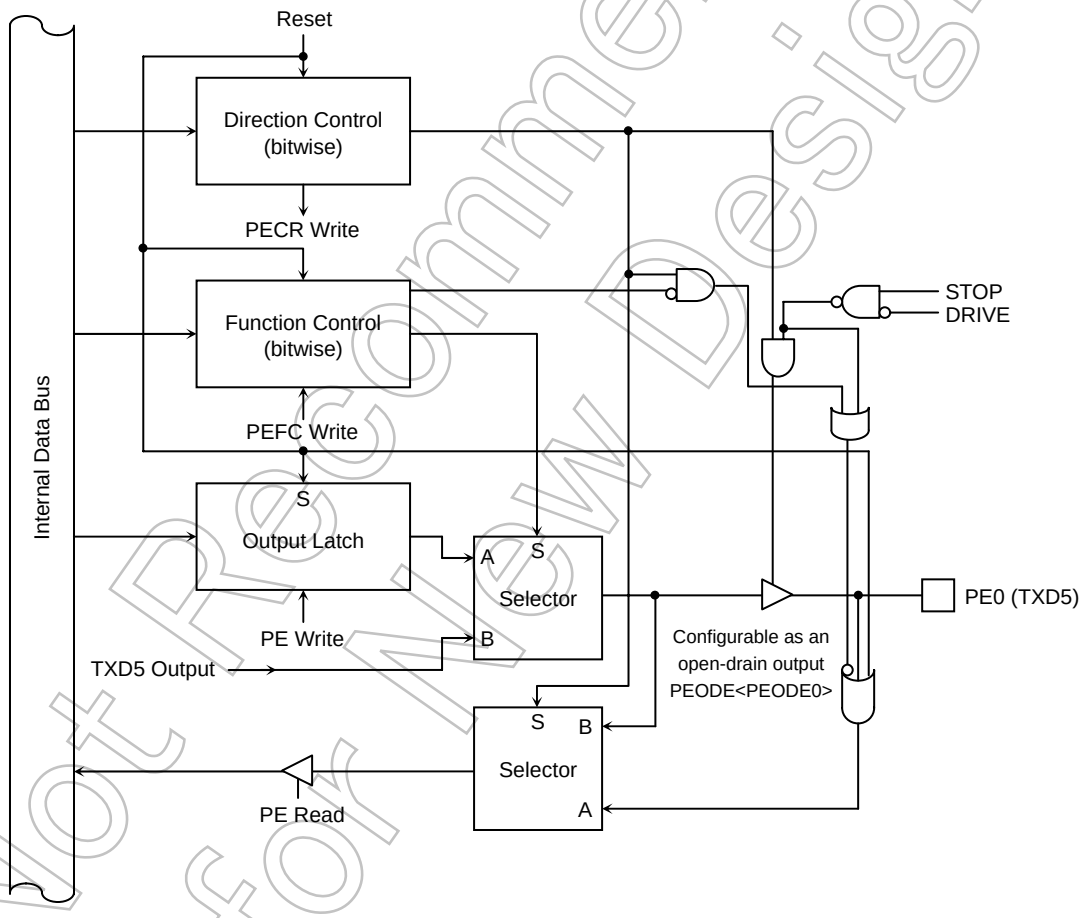


Figure 7.36 Port E (PE0)

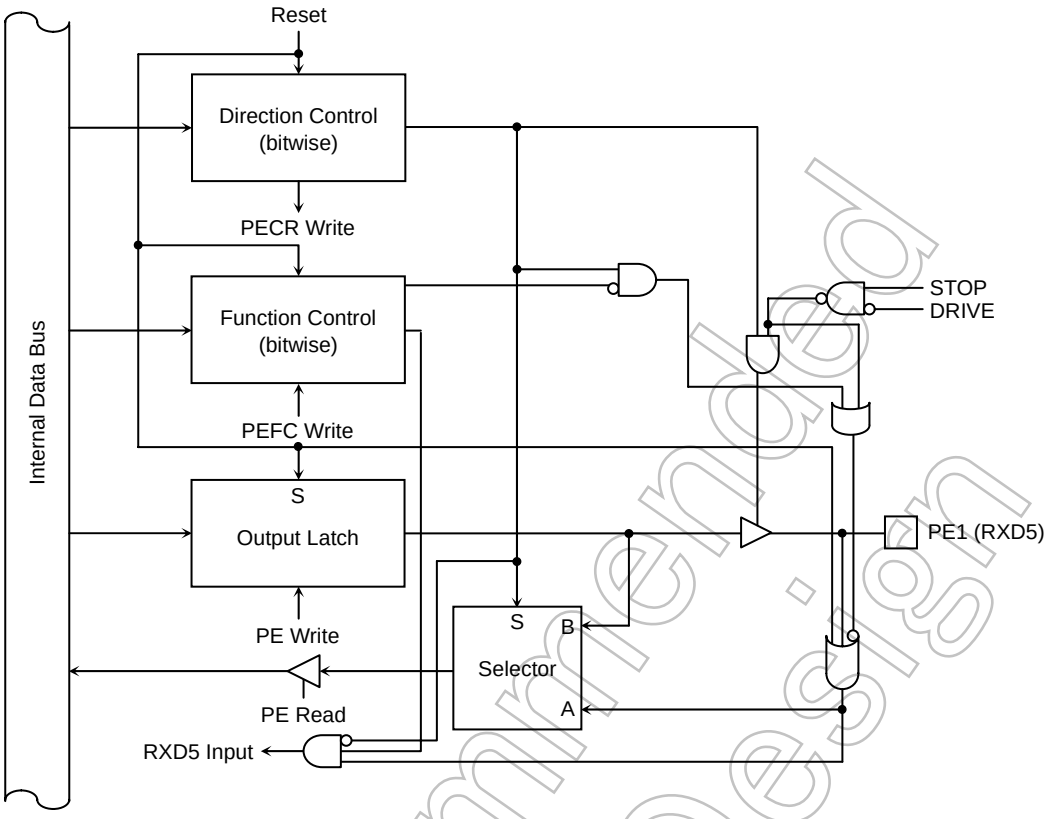


Figure 7.37 Port E (PE1)

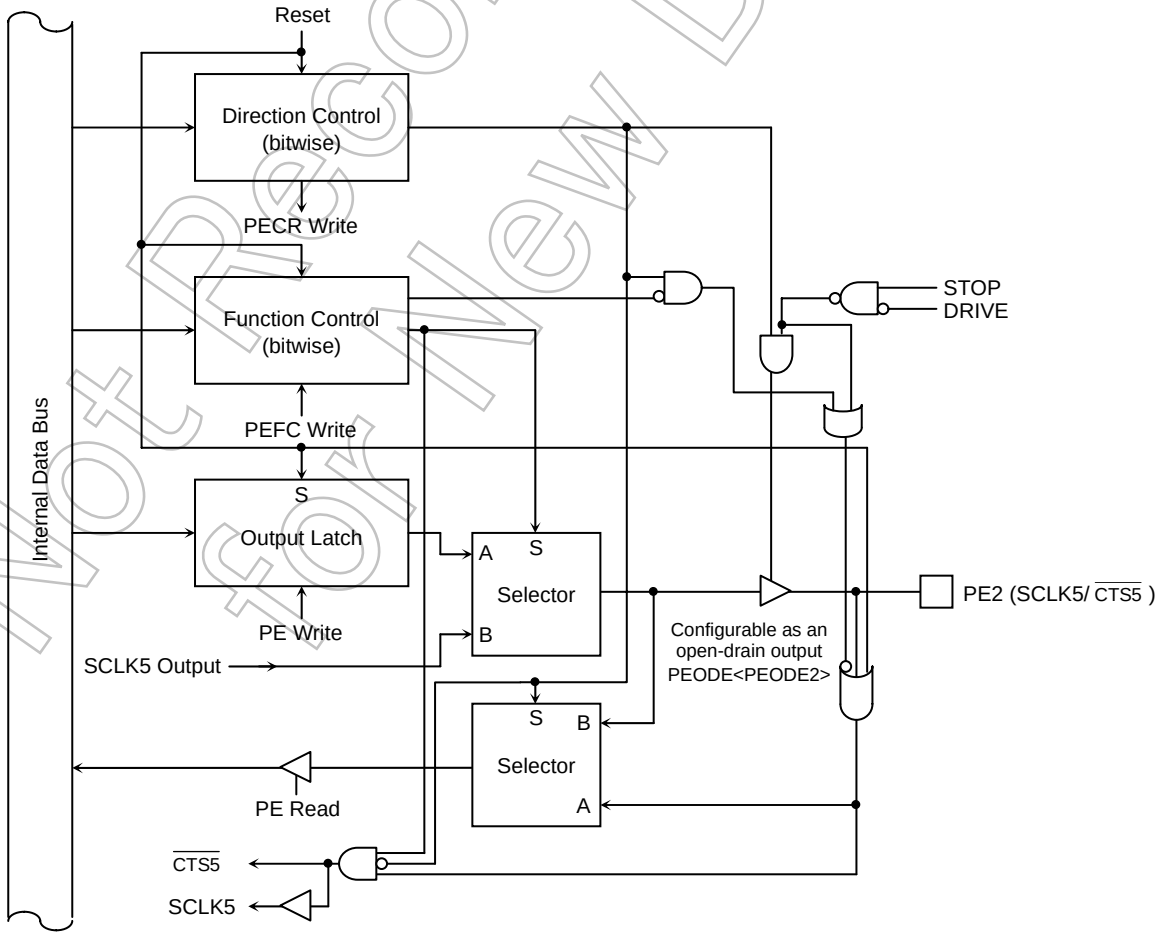


Figure 7.38 Port E (PE2)

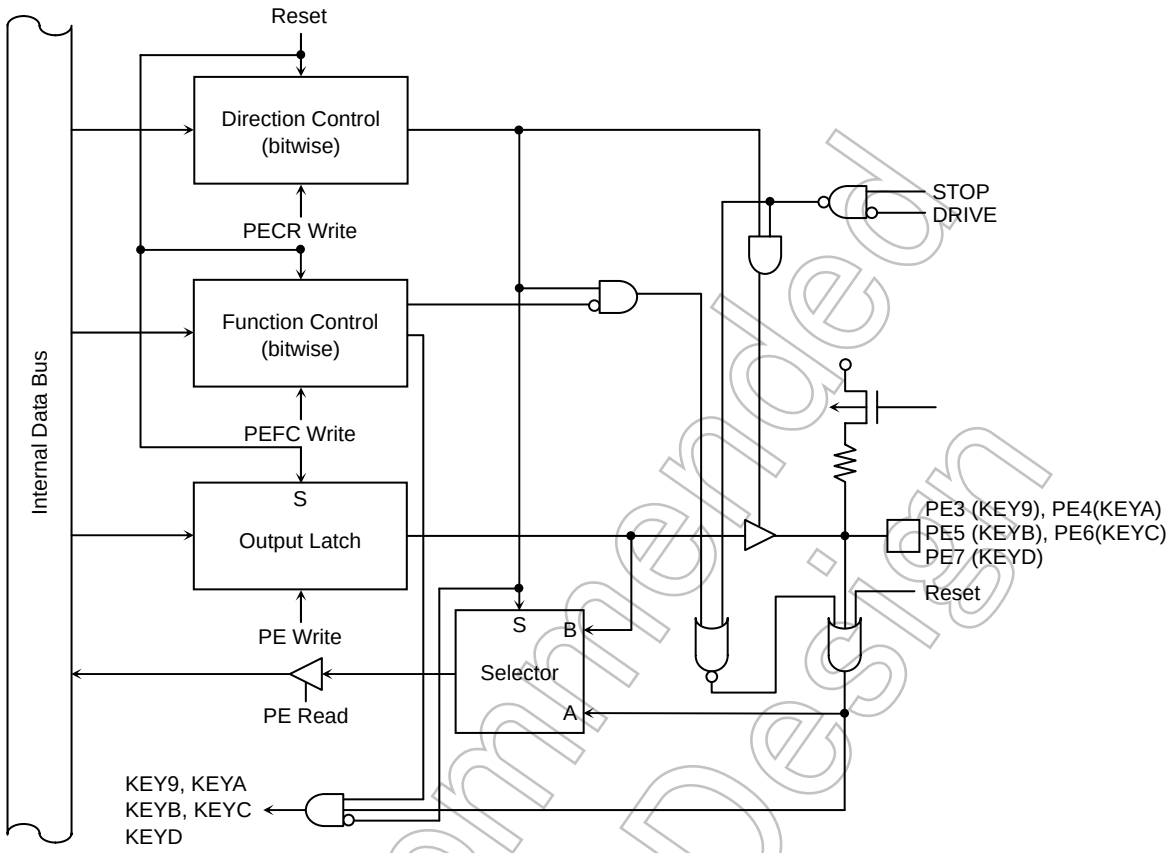


Figure 7.39 Port E (PE3, PE4, PE5, PE6, PE7)

Port E Register

		7	6	5	4	3	2	1	0
PE	Bit Symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
0xFFFF_F050	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port E Control Register

		7	6	5	4	3	2	1	0
PECR	Bit Symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
0xFFFF_F054	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port E Function Register

		7	6	5	4	3	2	1	0
PEFC	Bit Symbol	PE7F	PE6F	PE5F	PE4F	PE3F	PE2F	PE1F	PE0F
0xFFFF_F058	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: KEYD	0: PORT 1: KEYC	0: PORT 1: KEYB	0: PORT 1: KEYA	0: PORT 1: KEY9	0: PORT 1: SCLK5 $\overline{\text{CTS5}}$	0: PORT 1: RXD5	0: PORT 1: TXD5

Port E Open-Drain Enable Register

		7	6	5	4	3	2	1	0
PEODE	Bit Symbol	—	—	—	—	—	PEODE2	—	PEODE0
0xFFFF_F05C	Read/Write						W		W
	Reset Value						0		0
	Function						0: CMOS 1: Open-drain		0: CMOS 1: Open-drain

Figure 7.40 Port E Registers

7.14 Port F (PF0 - PF7)

Eight Port F pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PFCR register selects the direction of the Port F pins. Upon reset, the PFCR register bits are cleared to all 0s, configuring all Port F pins as input port pins. PF0-PF2 can be programmed as SBI inputs/outputs. PF3 and PF5 can be programmed as DMA request signal inputs. PF4 and PF6 can be programmed as DMA acknowledge signal outputs. PF7 can be programmed as a clock source input for the 32-bit time base timer. Setting the PFFC register bits configures the corresponding Port F pins for dedicated functions. A reset clears the PFCR and PFFC register bits, configuring all Port F pins as input port pins.

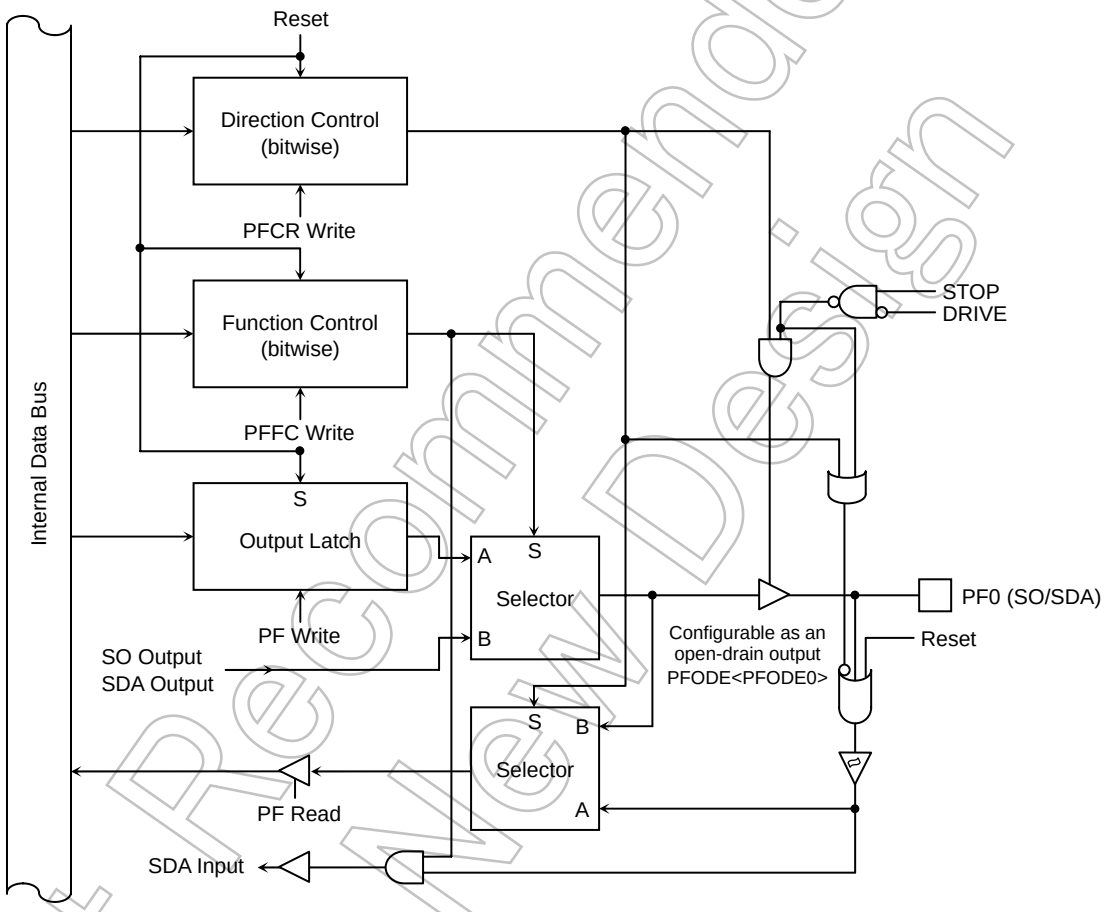


Figure 7.41 Port F (PF0)

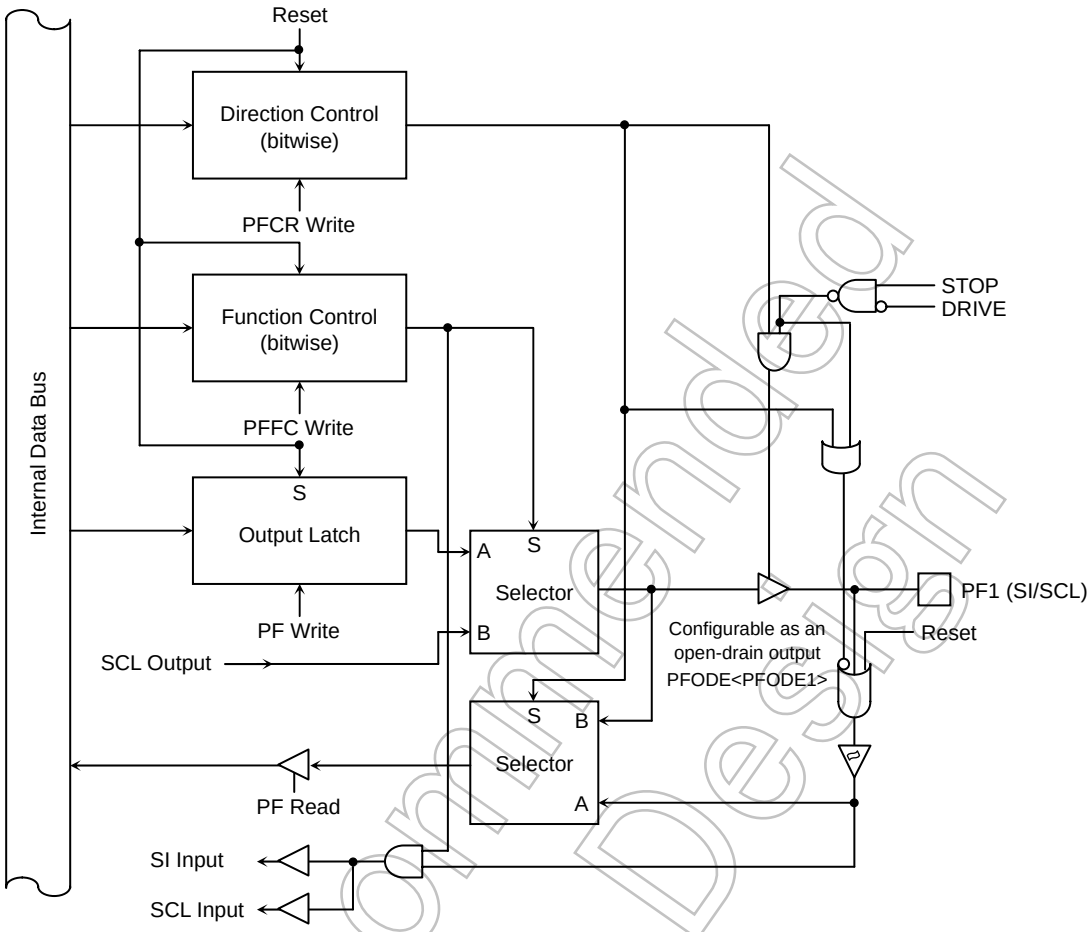


Figure 7.42 Port F (PF1)

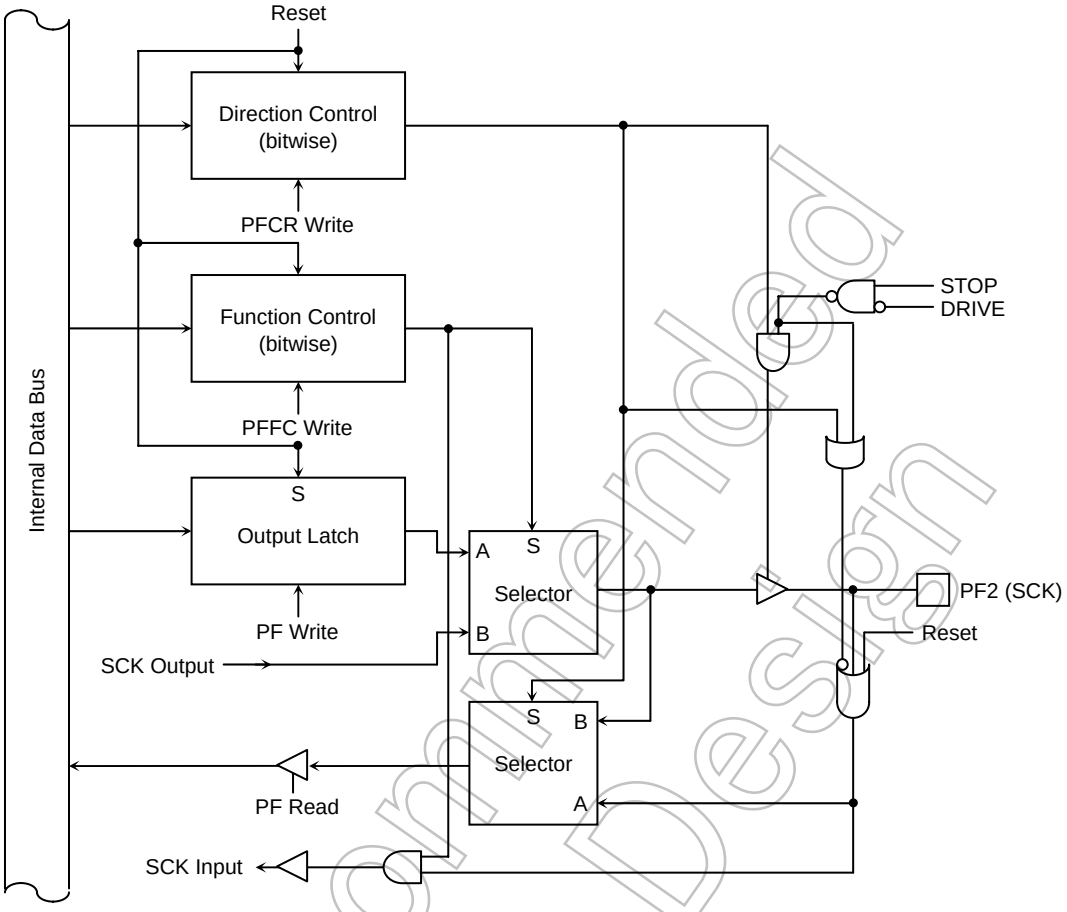


Figure 7.43 Port F (PF2)

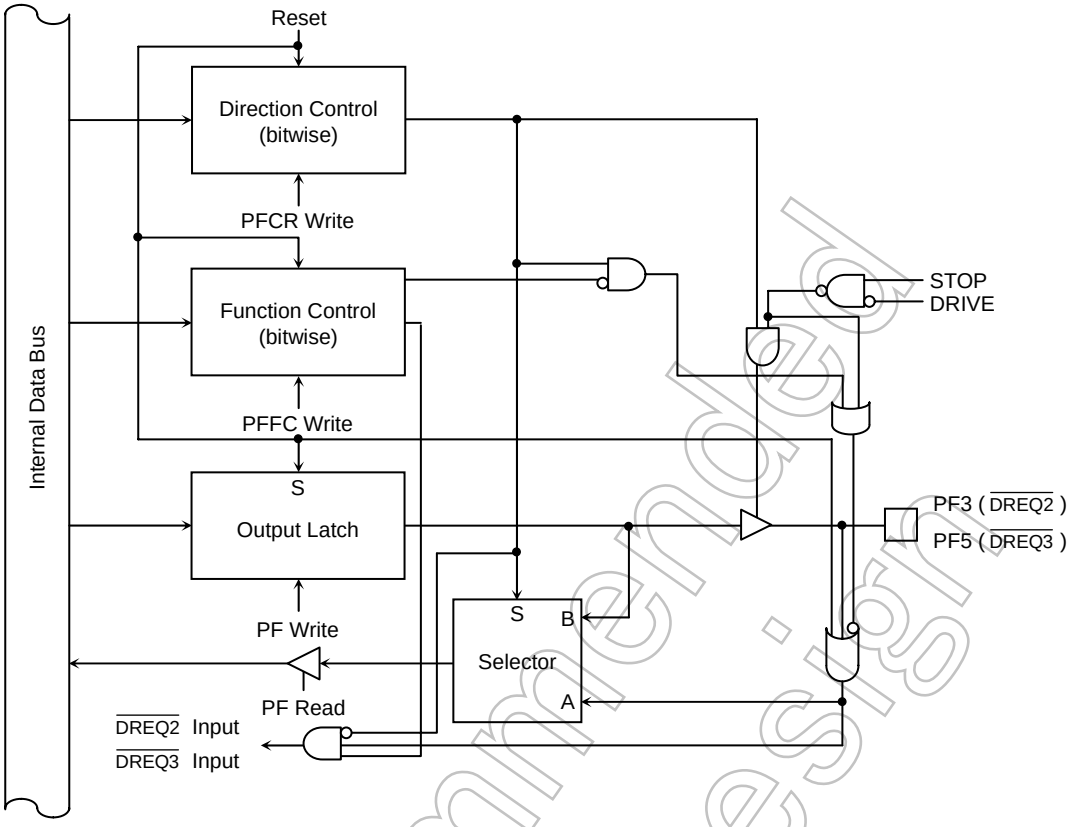


Figure 7.44 Port F (PF3, PF5)

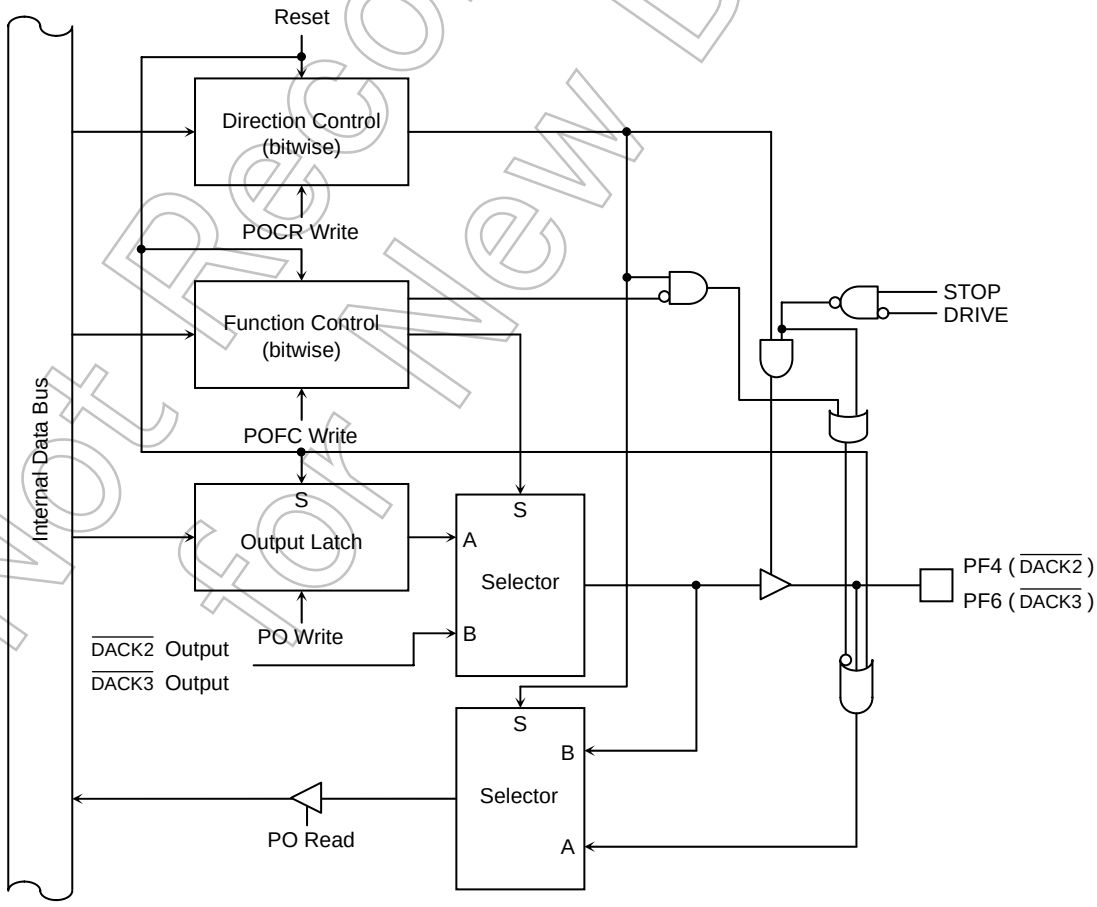


Figure 7.45 Port F (PF4, PF6)

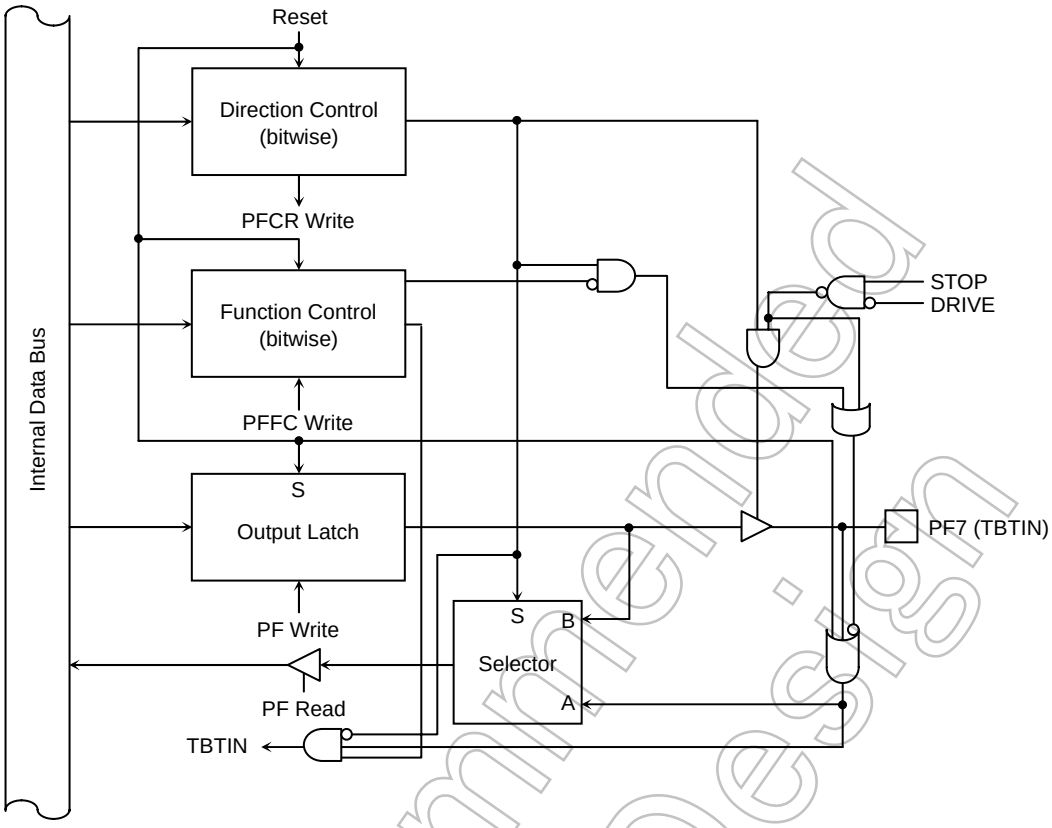


Figure 7.46 Port F (PF7)

Port F Register

		7	6	5	4	3	2	1	0
PF	Bit Symbol	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
0xFFFF_F063	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port F Control Register

		7	6	5	4	3	2	1	0
PFCR	Bit Symbol	PF7C	PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
0xFFFF_F067	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port F Function Register

		7	6	5	4	3	2	1	0
PFFC	Bit Symbol	PF7F	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
0xFFFF_F06B	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TBTIN	0: PORT 1: $\overline{\text{DACK3}}$	0: PORT 1: $\overline{\text{DREQ3}}$	0: PORT 1: $\overline{\text{DACK2}}$	0: PORT 1: $\overline{\text{DREQ2}}$	0: PORT 1: SCK	0: PORT 1: SI SCL	0: PORT 1: SO SDAO

Port F Open-Drain Enable Register

		7	6	5	4	3	2	1	0
PFODE	Bit Symbol	—	—	—	—	—	—	PFODE1	PFODE0
0xFFFF_F06F	Read/Write							W	
	Reset Value							0	0
	Function							0: CMOS 1: Open-drain	0: CMOS 1: Open-drain

Figure 7.47 Port F Registers

7.15 Port G (PG0 - PG7)

Eight Port G pins can be individually programmed to function as discrete general-purpose I/O pins or 32-bit capture trigger input pins. The PGCR register selects the direction of the Port G pins. Upon reset, the PGCR register bits are cleared to all 0s, configuring all Port G pins as input port pins. Setting the PGFC register bits configures the corresponding Port G pins as 32-bit capture trigger inputs. A reset clears the PGCR and PGFC register bits, configuring all Port G pins as input port pins.

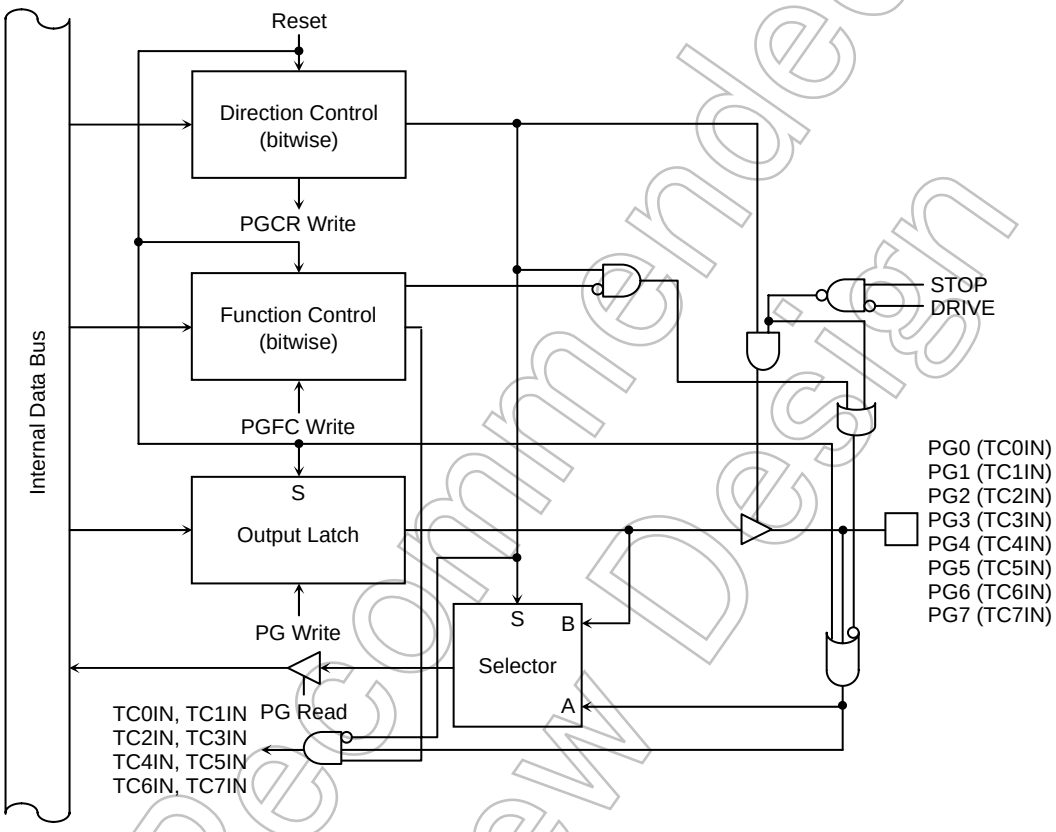


Figure 7.48 Port G (PG0 - PG7)

Port G Register

		7	6	5	4	3	2	1	0
PG	Bit Symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
0xFFFF_F062	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port G Control Register

		7	6	5	4	3	2	1	0
PGCR	Bit Symbol	PG7C	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
0xFFFF_F066	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port G Function Register

		7	6	5	4	3	2	1	0
PGFC	Bit Symbol	PG7F	PG6F	PG5F	PG4F	PG3F	PG2F	PG1F	PG0F
0xFFFF_F06A	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TC7IN	0: PORT 1: TC6IN	0: PORT 1: TC5IN	0: PORT 1: TC4IN	0: PORT 1: TC3IN	0: PORT 1: TC2IN	0: PORT 1: TC1IN	0: PORT 1: TC0IN

Figure 7.49 Port G Registers

7.16 Port H (PH0 - PH7)

Eight Port H pins can be individually programmed to function as discrete general-purpose I/O pins or 32-bit compare output pins. The PHCR register selects the direction of the Port H pins. Upon reset, the PHCR register bits are cleared to all 0s, configuring all Port H pins as input port pins. Setting the PHFC register bits configures the corresponding Port H pins as 32-bit compare outputs. A reset clears the PHCR and PHFC register bits, configuring all Port H pins as input port pins.

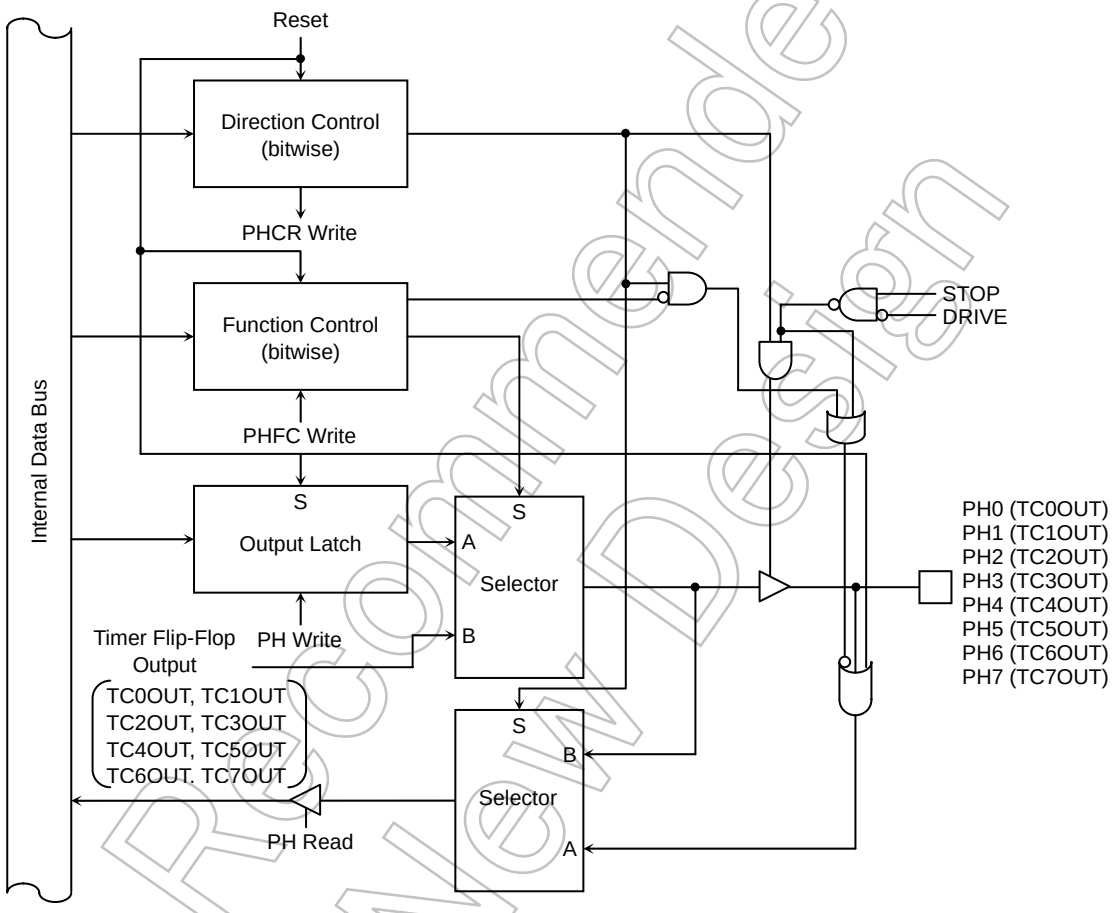


Figure 7.50 Port H (PH0 - PH7)

Port H Register

		7	6	5	4	3	2	1	0
PH	Bit Symbol	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
0xFFFF_F061	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port H Control Register

		7	6	5	4	3	2	1	0
PHCR	Bit Symbol	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
0xFFFF_F065	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port H Function Register

		7	6	5	4	3	2	1	0
PHFC	Bit Symbol	PH7F	PH6F	PH5F	PH4F	PH3F	PH2F	PH1F	PH0F
0xFFFF_F069	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TCOUT7	0: PORT 1: TCOUT6	0: PORT 1: TCOUT5	0: PORT 1: TCOUT4	0: PORT 1: TCOUT3	0: PORT 1: TCOUT2	0: PORT 1: TCOUT1	0: PORT 1: TCOUT0

Figure 7.51 Port H Registers

7.17 Port I (PI0 - PI7)

Eight Port I pins can be individually programmed to function as discrete general-purpose I/O pins or dedicated input pins. The PICR register selects the direction of the Port I pins. Upon reset, the PICR register bits are cleared to all 0s, configuring all Port I pins as input port pins. PI0 can be programmed as an trigger input for the A/D converter. PI1-PI6 can be programmed as external interrupt sources. Setting the PIFC register bits configures the corresponding Port I pins for dedicated functions. A reset clears the PICR and PIFC register bits, configuring all Port I pins as input port pins.

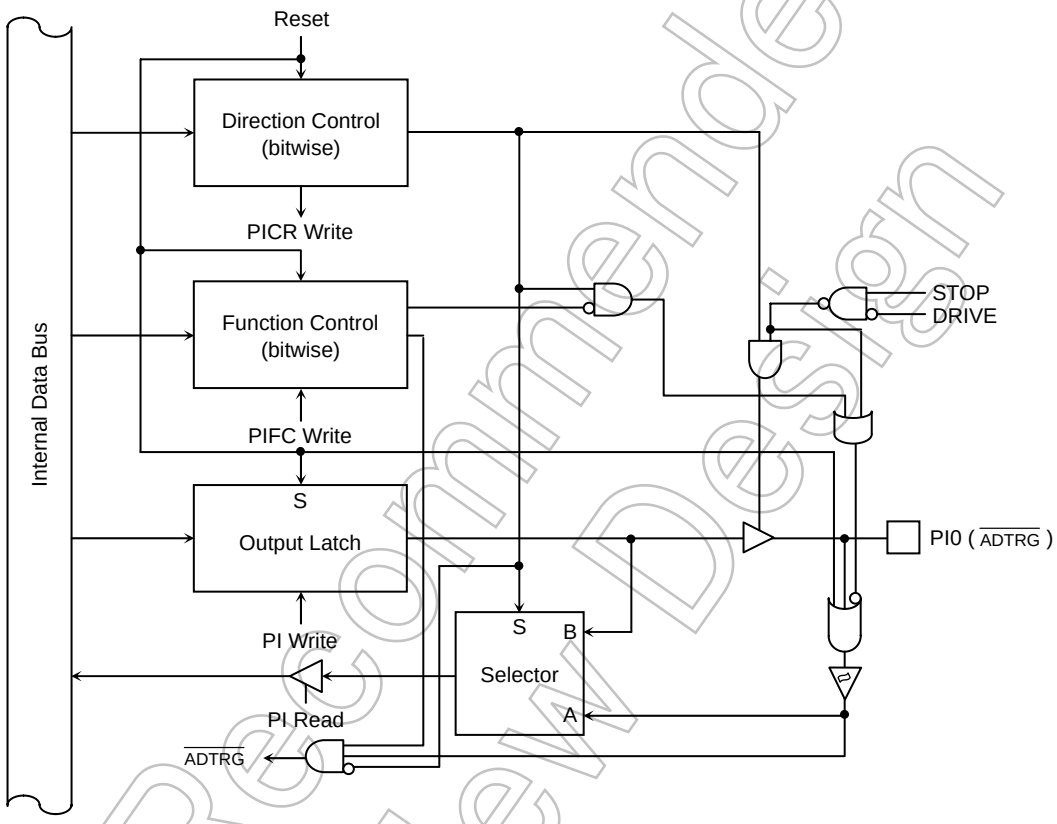


Figure 7.52 Port I (PI0)

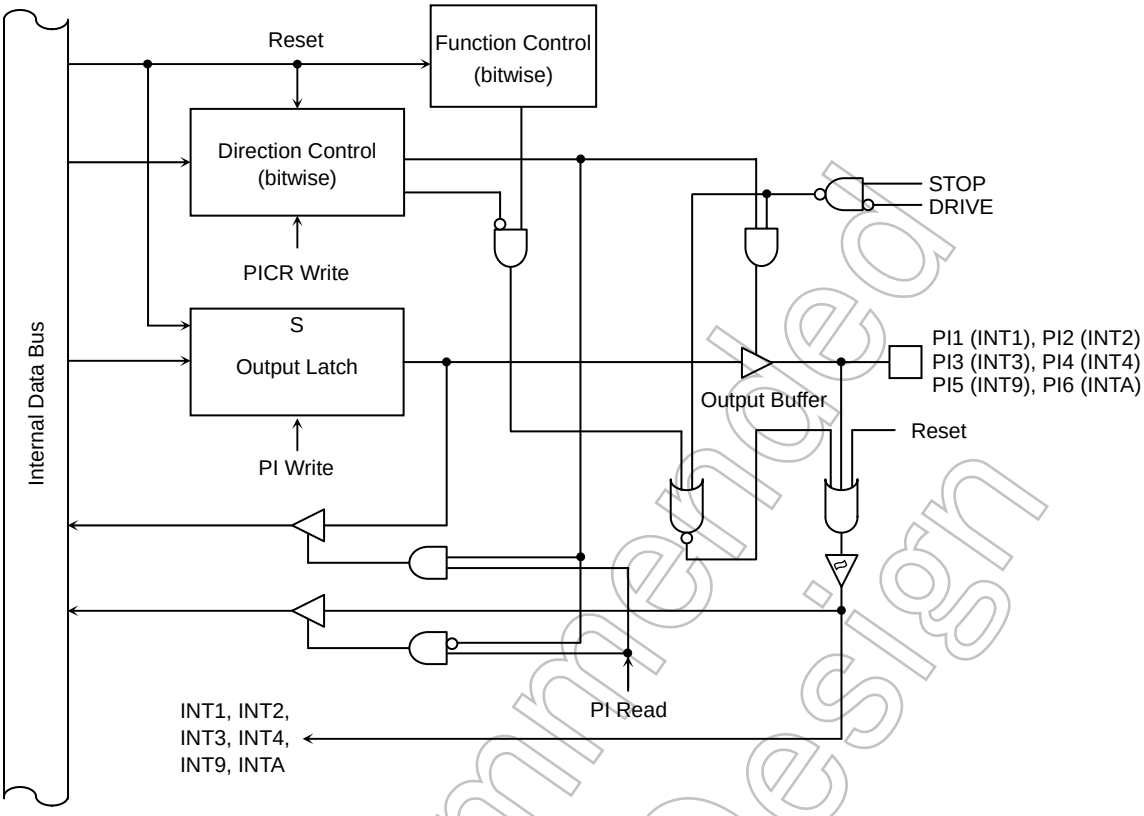


Figure 7.53 Port I (PI1 - PI6)

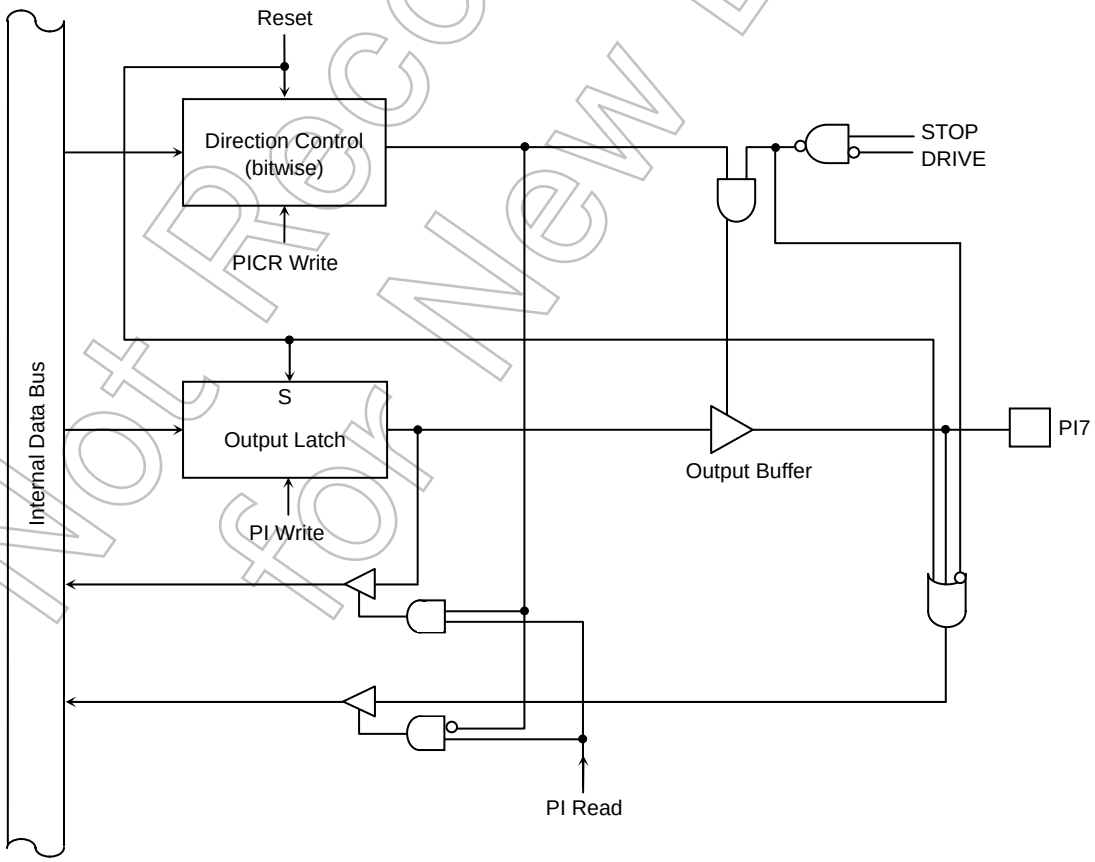


Figure 7.54 Port I (PI7)

Port I Register

		7	6	5	4	3	2	1	0
PI	Bit Symbol	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0
0xFFFF_F060	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port I Control Register

		7	6	5	4	3	2	1	0
PICR	Bit Symbol	PI7C	PI6C	PI5C	PI4C	PI3C	PI2C	PI1C	PI0C
0xFFFF_F064	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port I Function Register

		7	6	5	4	3	2	1	0
PIFC	Bit Symbol	—	PI6F	PI5F	PI4F	PI3F	PI2F	PI1F	PI0F
0xFFFF_F068	Read/Write		W						
	Reset Value		0	0	0	0	0	0	0
	Function		0: PORT 1: INTA	0: PORT 1: INT9	0: PORT 1: INT4	0: PORT 1: INT3	0: PORT 1: INT2	0: PORT 1: INT1	0: PORT 1: *ADTRG

Function	Corresponding Bit in PIFC	Corresponding Bit in PICR	Port Used
$\overline{\text{ADTRG}}$ output settings	1	0	PI0
INT1 input settings	1 (*1)	0	PI1
INT2 input settings	1 (*1)	0	PI2
INT3 input settings	1 (*1)	0	PI3
INT4 input settings	1 (*1)	0	PI4
INT9 input settings	1 (*1)	0	PI5
INTA input settings	1 (*1)	0	PI6

***1: This bit must be set when the corresponding interrupt source is used for STOP wake-up signaling with SYSCR.DRVE cleared to 0. Otherwise, the bit need not be set.**

Figure 7.55 Port I Registers

7.18 Port J (PJ0 - PJ4)

Five Port J pins can be individually programmed to function as discrete general-purpose I/O pins or dedicated input pins. The PJCR register selects the direction of the Port J pins. Upon reset, the PJCR register bits are cleared to all 0s, configuring all Port J pins as input port pins. PJ0 can be programmed as an external interrupt source. Setting the corresponding PJFC register bit configures PJ0 as an external interrupt source pin. A reset clears the PJCR and PJFC register bits, configuring all Port J pins as input port pins.

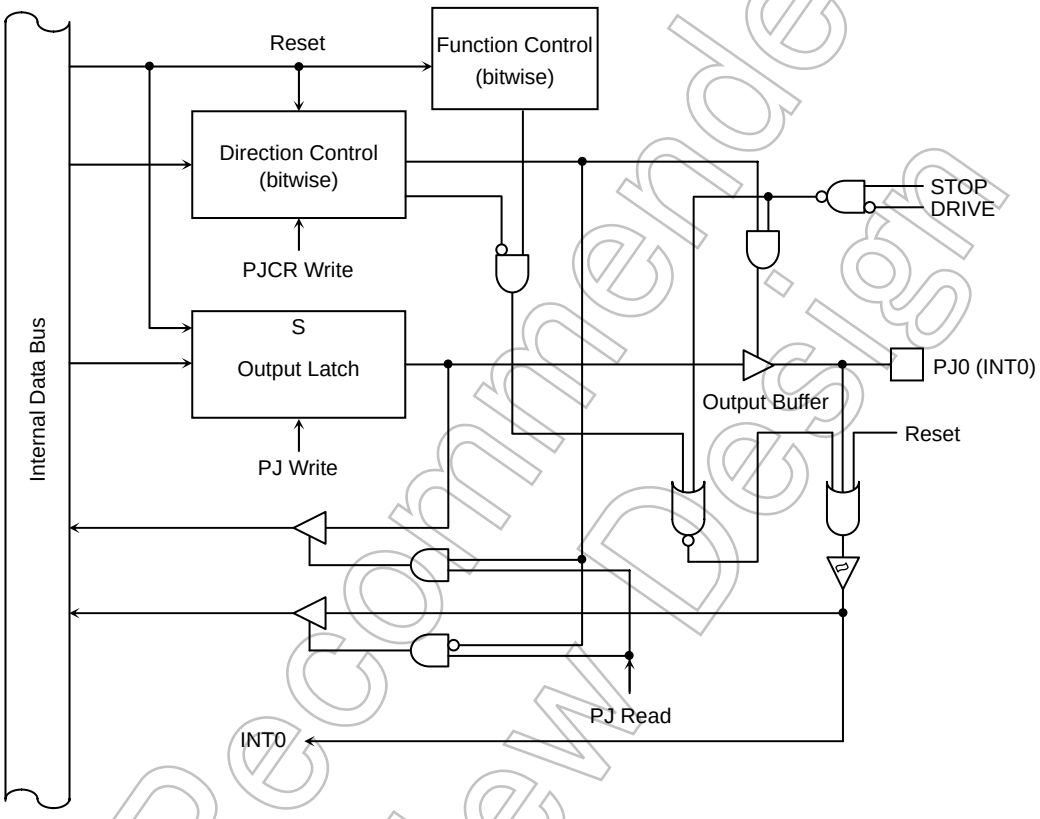


Figure 7.56 Port J (PJ0)

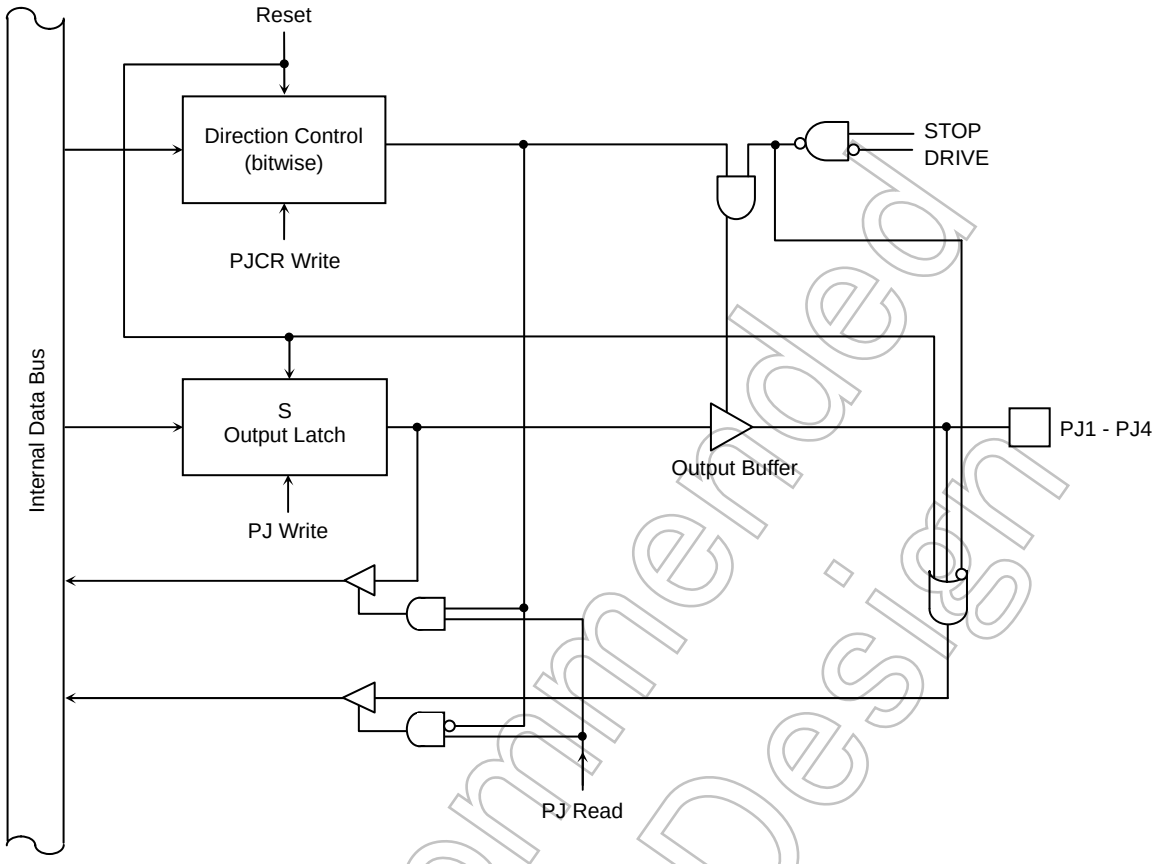


Figure 7.57 Port J (PJ1 - PJ4)

Port J Register

		7	6	5	4	3	2	1	0
PJ	Bit Symbol	—	—	—	PJ4	PJ3	PJ2	PJ1	PJ0
0xFFFF_F0C3	Read/Write				R/W				
	Reset Value				Input mode (The output latch is set to 1.)				

Port J Control Register

		7	6	5	4	3	2	1	0
PJCR	Bit Symbol	—	—	—	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
0xFFFF_F0C7	Read/Write				W				
	Reset Value				0	0	0	0	0
	Function				0: Input, 1: Output				

Port J Function Register

		7	6	5	4	3	2	1	0
PJFC	Bit Symbol	—	—	—	—	—	—	—	PJ0F
0xFFFF_F0CB	Read/Write								W
	Reset Value								0
	Function								0: PORT 1: INT0

Function	Corresponding Bit in PJFC	Corresponding Bit in PJCR	Port Used
INT0 input settings	1 (*1)	0	PJ0

***1: This bit must be set when the corresponding interrupt source is used for STOP wake-up signaling with SYSCR.DRVE cleared to 0. Otherwise, the bit need not be set.**

Figure 7.58 Port J Registers

7.19 Port K (PK0 - PK7)

Eight Port K pins can be individually programmed to function as discrete general-purpose I/O pins or key-pressed wake-up input pins. The PKCR register selects the direction of the Port K pins. Upon reset, the PKCR register bits are cleared to all 0s, configuring all Port K pins as input port pins. Setting the PKFC register bits configures the corresponding Port K pins as key-pressed wake-up inputs. A reset clears the PKCR and PKFC register bits, configuring all Port K pins as input port pins.

PK0-PK7 have internal pull-up resistors, which are enabled when key input is enabled through the programming of KWUPSTn with the KWUPCNT.KYPE bit set to 1 in the key-pressed wake-up circuit block. For details, refer to Chapter 19. The pull-up resistors are disabled when the PK0-PK7 pins are used as general-purpose I/O pins.

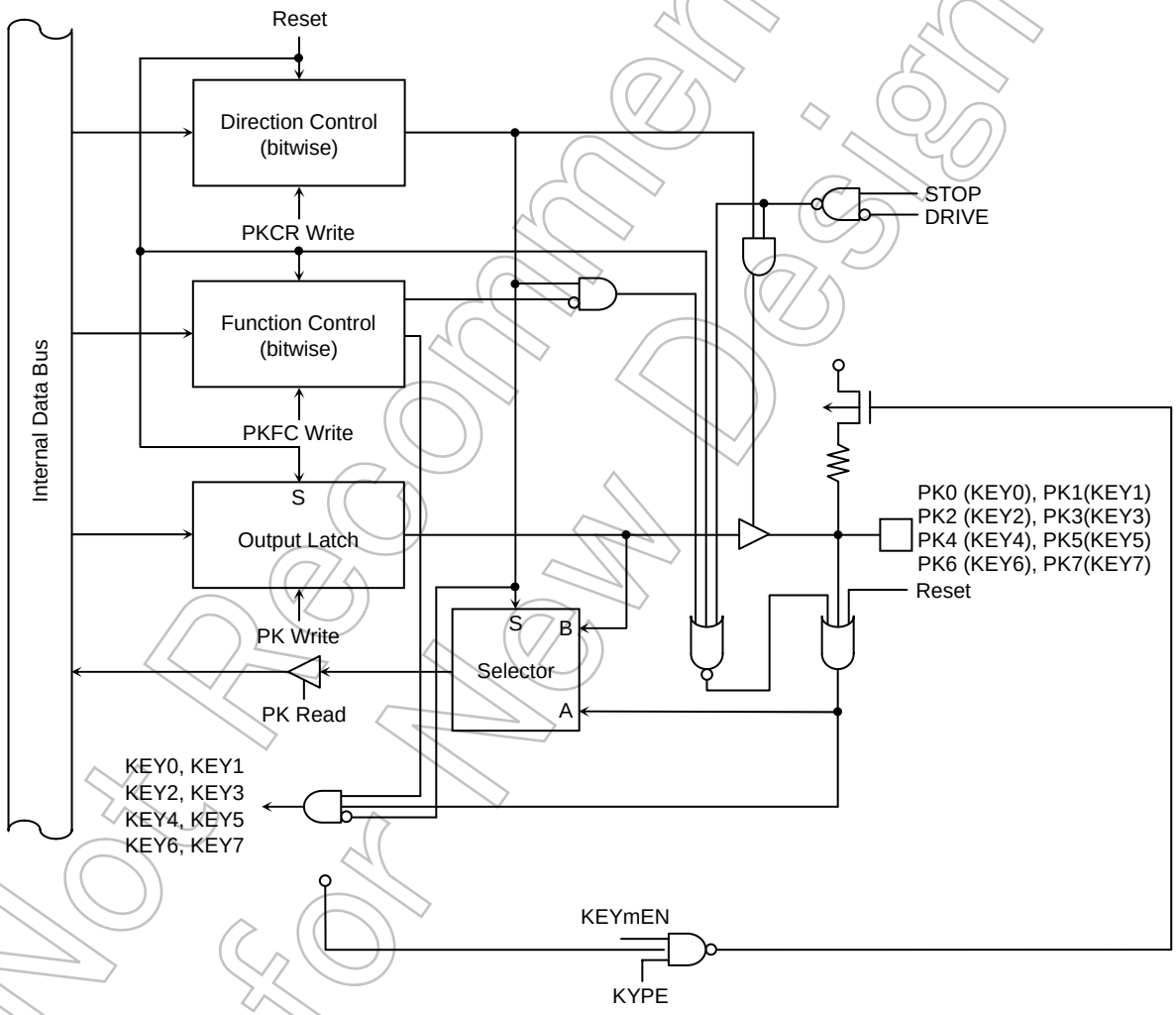


Figure 7.59 Port K (PK0 - PK7)

Port K Register

		7	6	5	4	3	2	1	0
PK	Bit Symbol	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
0xFFFF_F0C2	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port K Control Register

		7	6	5	4	3	2	1	0
PKCR	Bit Symbol	PK7C	PK6C	PK5C	PK4C	PK3C	PK2C	PK1C	PK0C
0xFFFF_F0C6	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port K Function Register

		7	6	5	4	3	2	1	0
PKFC	Bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
0xFFFF_F0CA	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: KEY7	0: PORT 1: KEY6	0: PORT 1: KEY5	0: PORT 1: KEY4	0: PORT 1: KEY3	0: PORT 1: KEY2	0: PORT 1: KEY1	0: PORT 1: KEY0

Figure 7.60 Port K Registers

7.20 Port L (PL0 - PL7)

Eight Port L pins can be individually programmed to function as discrete general-purpose I/O pins or timer input pins. The PLCR register selects the direction of the Port L pins. Upon reset, the PLCR register bits are cleared to all 0s, configuring all Port L pins as input port pins. PL0-PL3 can be programmed as inputs to 8-bit timers. PL4-PL7 can be programmed as inputs to 16-bit timers. Setting the PLFC register bits configures the corresponding Port L pins for timer functions. A reset clears the PLCR and PLFC register bits, configuring all Port L pins as input port pins.

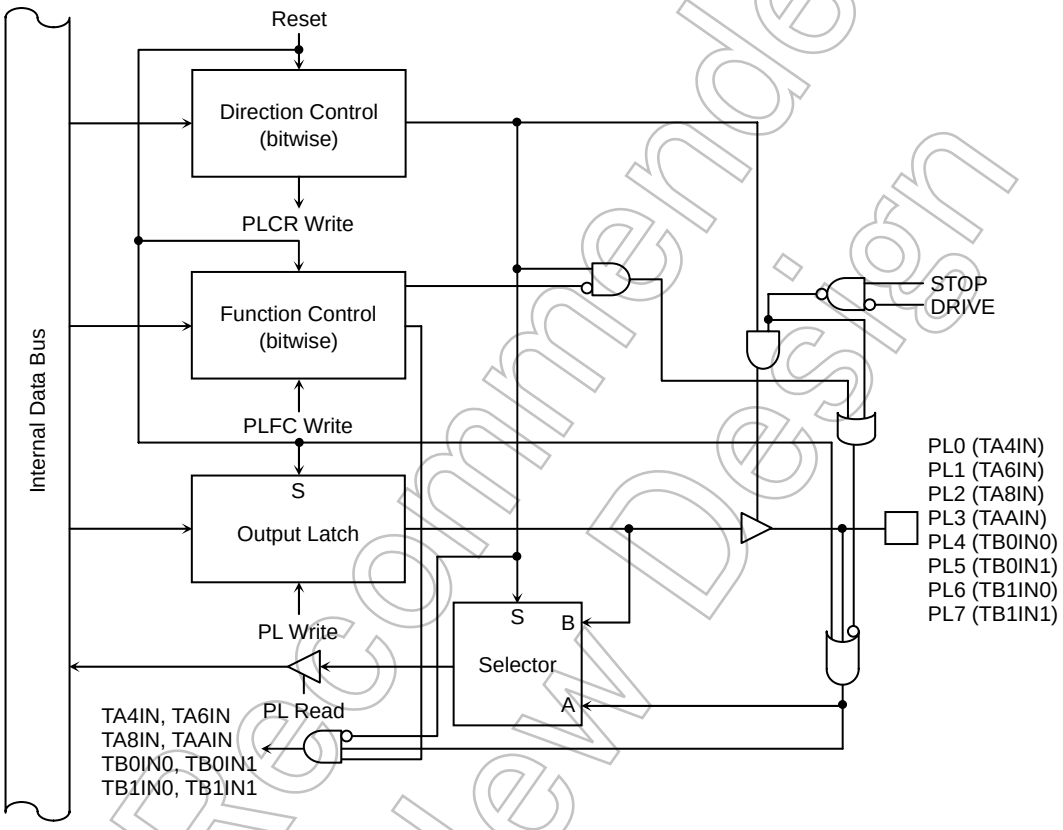


Figure 7.61 Port L (PL0 - PL7)

Port L Register

		7	6	5	4	3	2	1	0
PL	Bit Symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
0xFFFF_F0C1	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port L Control Register

		7	6	5	4	3	2	1	0
PLCR	Bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
0xFFFF_F0C5	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port L Function Register

		7	6	5	4	3	2	1	0
PLFC	Bit Symbol	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
0xFFFF_F0C9	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TB1IN1	0: PORT 1: TB1IN0	0: PORT 1: TB0IN1	0: PORT 1: TB0IN0	0: PORT 1: TAAIN	0: PORT 1: TA8IN	0: PORT 1: TA6IN	0: PORT 1: TA4IN

Figure 7.62 Port L Registers

7.21 Port M (PM0 - PM7)

Eight Port M pins can be individually programmed to function as discrete general-purpose I/O pins. The PMCR register selects the direction of the Port M pins. Upon reset, the PMCR register bits are cleared to all 0s, configuring all Port M pins as input port pins.

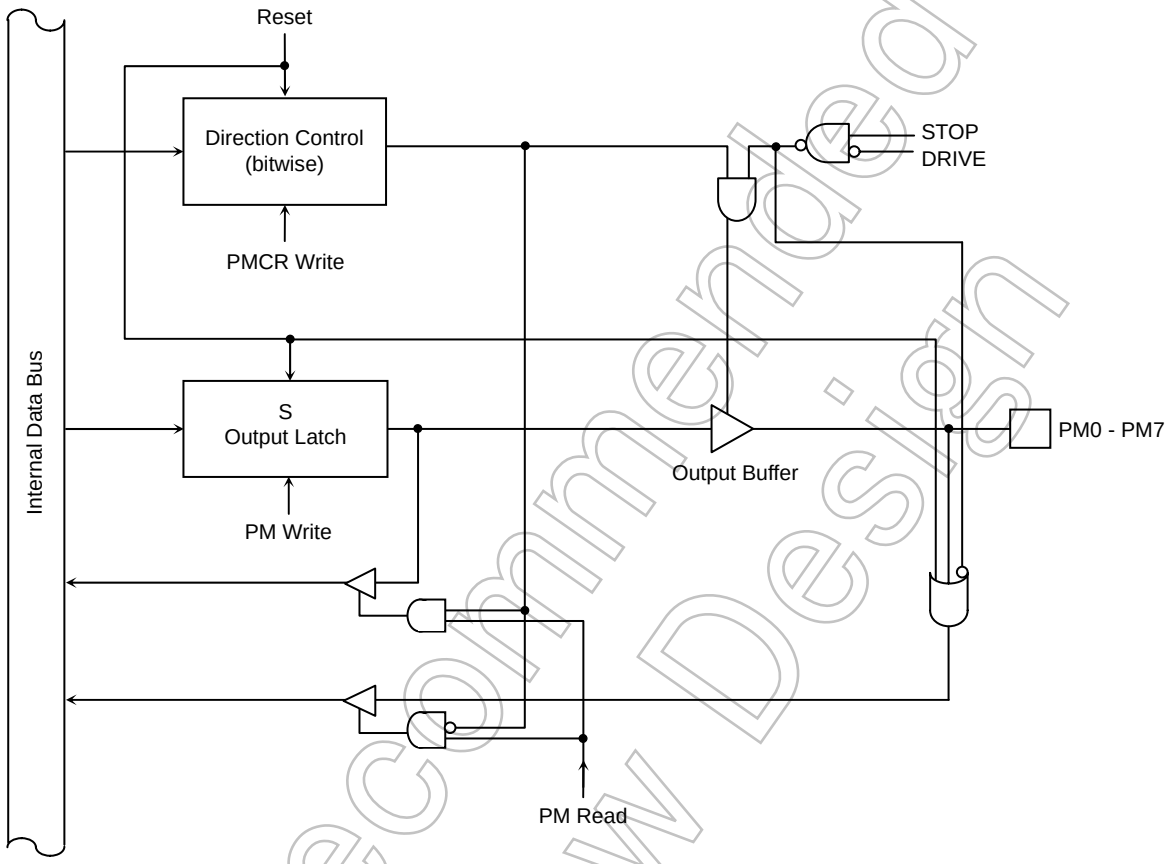


Figure 7.63 Port M (PM0 - PM7)

Port M Register

		7	6	5	4	3	2	1	0
PM	Bit Symbol	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
0xFFFF_F0C0	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port M Control Register

		7	6	5	4	3	2	1	0
PMCR	Bit Symbol	PM7C	PM6C	PM5C	PM4C	PM3C	PM2C	PM1C	PM0C
0xFFFF_F0C4	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Figure 7.64 Port M Registers

7.22 Port N (PN0 - PN7)

Eight Port N pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. The PNCR register selects the direction of the Port N pins. Upon reset, the PNCR register bits are cleared to all 0s, configuring all Port N pins as input port pins. PN0 can be programmed as an SIO data output. PN1 can be programmed as an SIO data input. PN2 can be programmed as an SIO clock input/output or CTS input. Setting the PNFC register bits configures the corresponding Port N pins for dedicated functions. A reset clears the PNCR and PNFC register bits, configuring all Port N pins as input port pins.

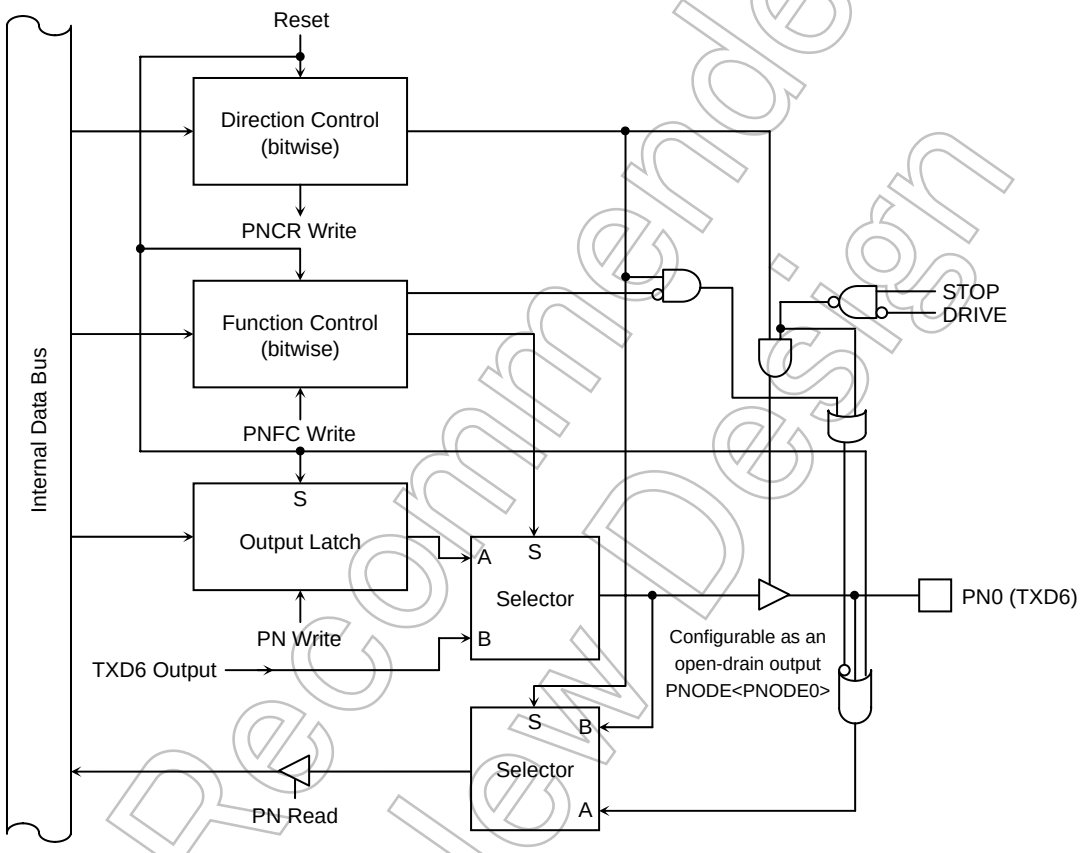


Figure 7.65 Port N (PN0)

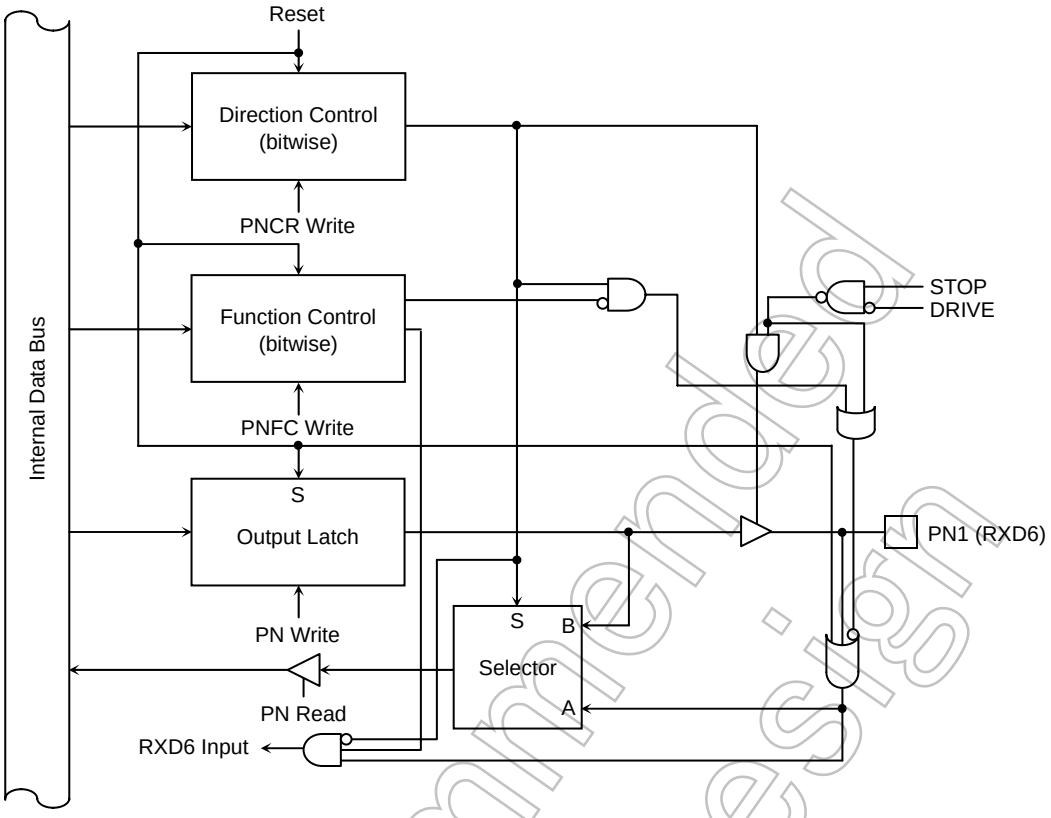


Figure 7.66 Port N (PN1)

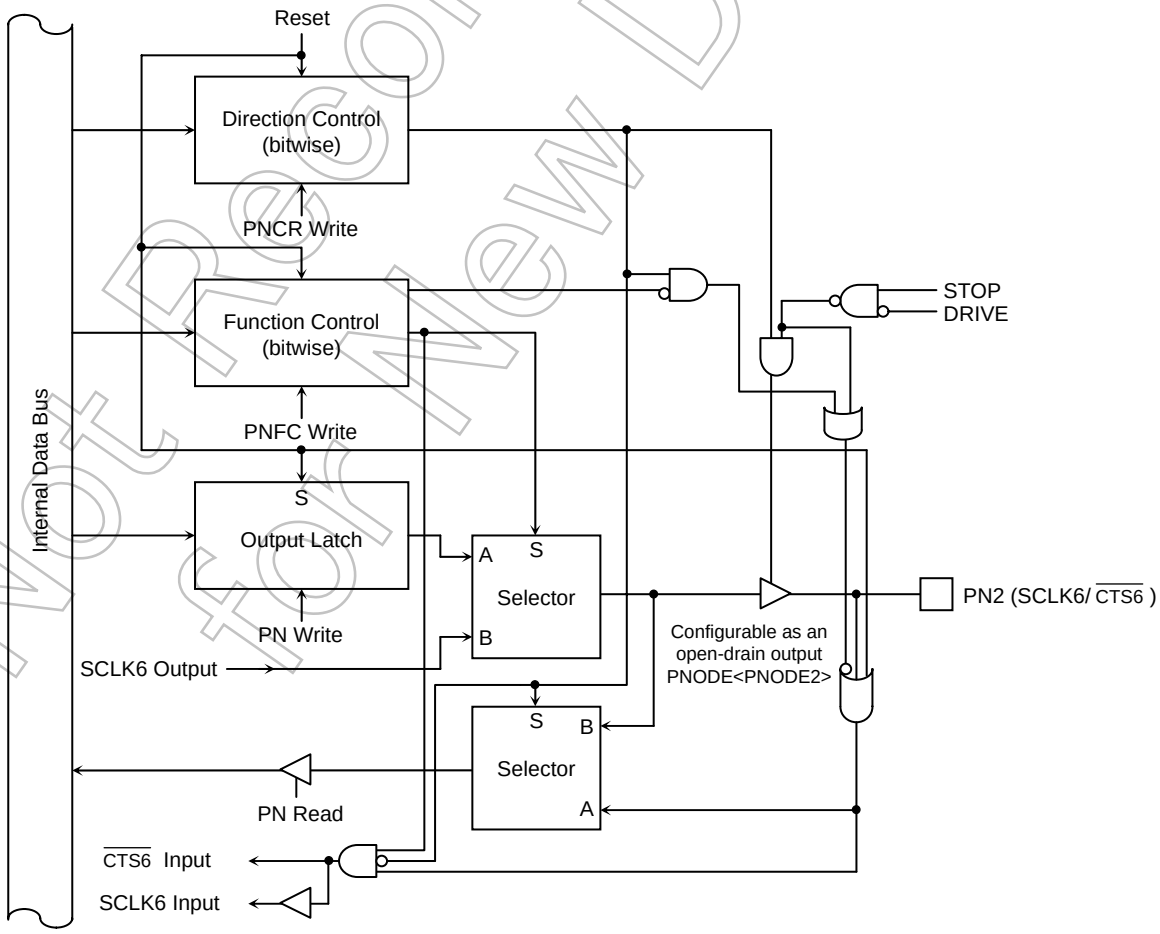


Figure 7.67 Port N (PN2)

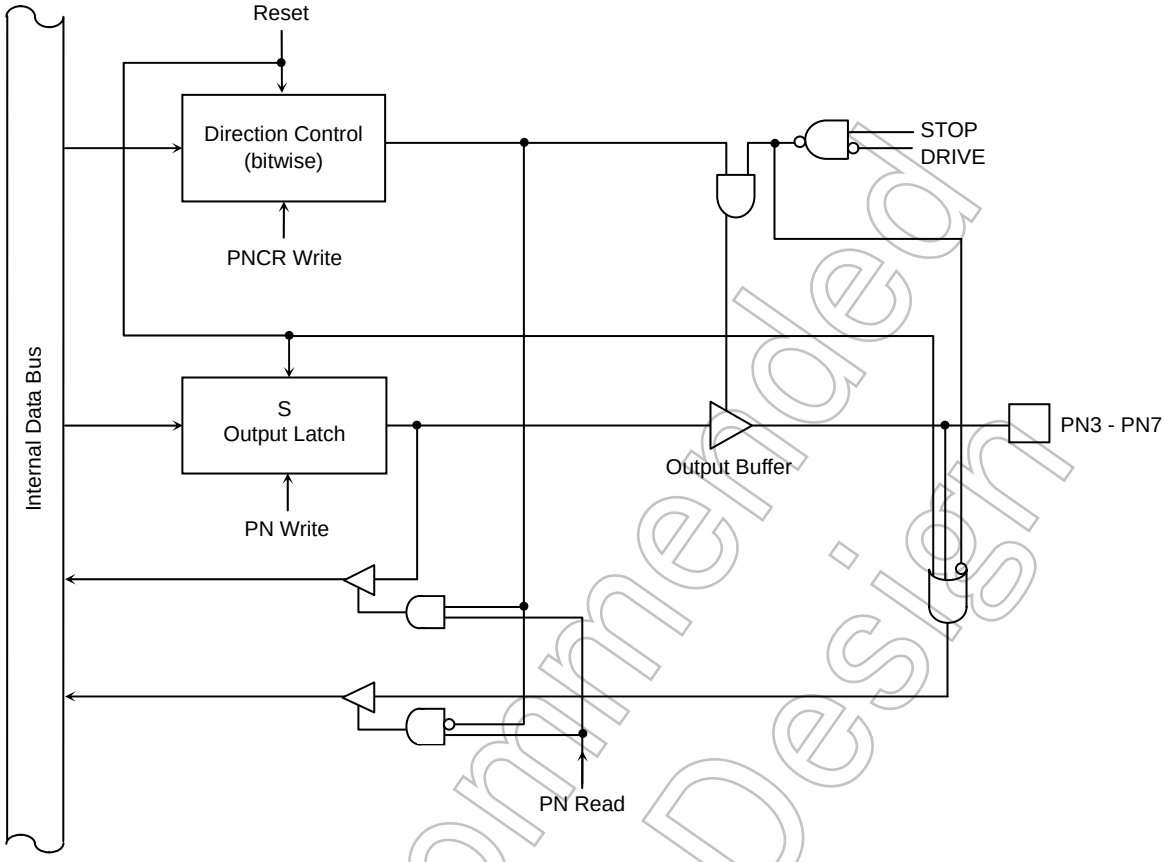


Figure 7.68 Port N (PN3 - PN7)

Port N Register

		7	6	5	4	3	2	1	0
PN	Bit Symbol	PN7	PN6	PN5	PN4	PN3	PN2	PN1	PN0
0xFFFF_F0D3	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port N Control Register

		7	6	5	4	3	2	1	0
PNCR	Bit Symbol	PN7C	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
0xFFFF_F0D7	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Port N Function Register

		7	6	5	4	3	2	1	0
PNFC	Bit Symbol	—	—	—	—	—	PN2F	PN1F	PN0F
0xFFFF_F0DB	Read/Write						W		
	Reset Value						0	0	0
	Function						0: PORT 1: SCLK6 *CTS6	0: PORT 1: RXD6	0: PORT 1: TXD6

Port N Open-Drain Enable Register

		7	6	5	4	3	2	1	0
PNODE	Bit Symbol	—	—	—	—	—	PNODE2	—	PNODE0
0xFFFF_F0DF	Read/Write						W		W
	Reset Value						0		0
	Function						0: CMOS 1: Open-drain		0: CMOS 1: Open-drain

Figure 7.69 Port N Registers

7.23 Port O and Port P (PO0 - PO7, PP0 - PP7)

Eight Port O pins and eight Port P pins can be individually programmed to function as discrete general-purpose I/O pins. The POCR and PPCR registers select the direction of the Port O and P pins, respectively. Upon reset, the POCR and PPCR register bits are cleared to all 0s, configuring all Port O and P pins as input port pins.

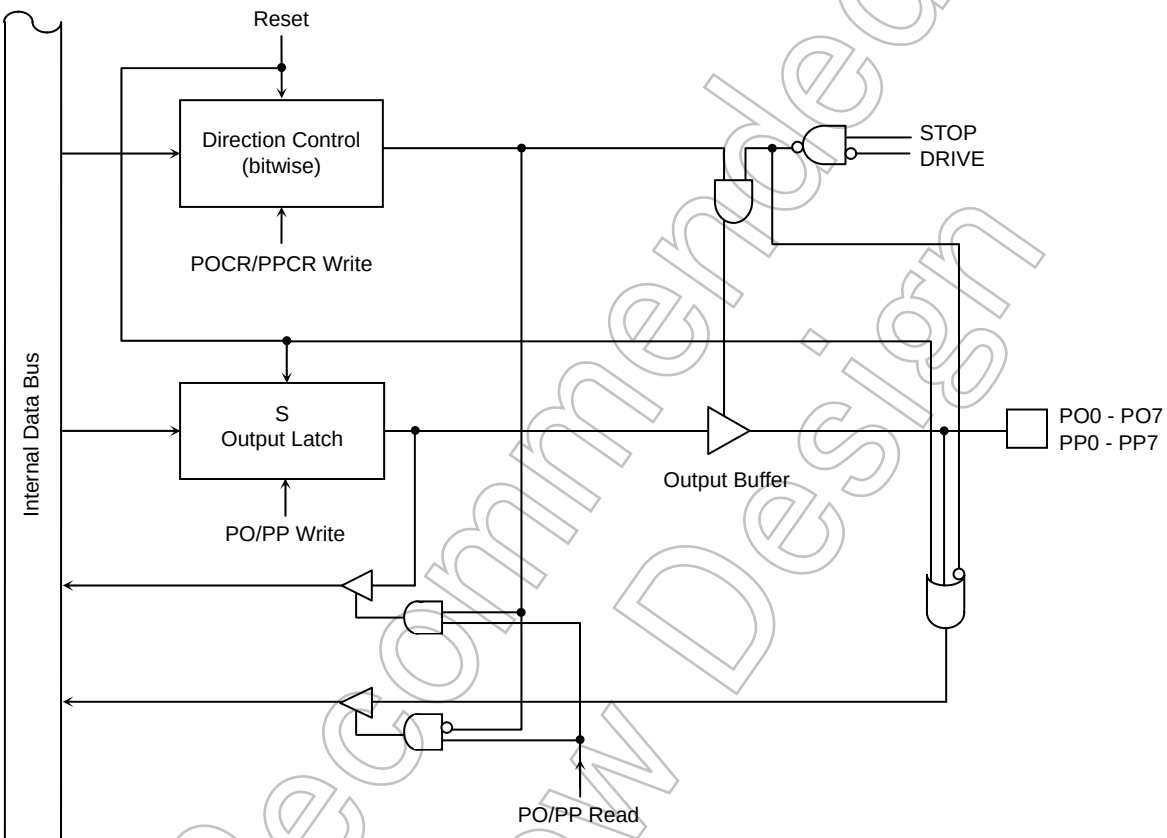


Figure 7.70 Port O and Port P (PO0 - PO7, PP0 - PP7)

Port O Register

		7	6	5	4	3	2	1	0
PO	Bit Symbol	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0
0xFFFF_F0D2	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port O Control Register

		7	6	5	4	3	2	1	0
POCR	Bit Symbol	PO7C	PO6C	PO5C	PO4C	PO3C	PO2C	PO1C	PO0C
0xFFFF_F0D6	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Figure 7.71 Port O Registers

Port P Register

		7	6	5	4	3	2	1	0
PP	Bit Symbol	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
0xFFFF_F0D1	Read/Write	R/W							
	Reset Value	Input mode (The output latch is set to 1.)							

Port P Control Register

		7	6	5	4	3	2	1	0
PPCR	Bit Symbol	PP7C	PP6C	PP5C	PP4C	PP3C	PP2C	PP1C	PP0C
0xFFFF_F0D5	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: Input, 1: Output							

Figure 7.72 Port P Registers

8. External Bus Interface

The TMP1962 contains external bus interface logic that handles the transfer of information between the internal buses and the memory or peripherals in the external address space. It consists of the External Bus Interface (EBIF) logic and the Chip Select/Wait Controller.

The CS/Wait Controller provides four programmable chip select signals, with variable block sizes. The chip select function supports automatic wait-state generation and data bus sizing (8-bit or 16-bit) for each of the four address blocks and the rest of the external address locations.

The EBIF logic controls the timing of the external bus, based on the settings of the CS/Wait Controller. The EBIF logic also performs dynamic bus sizing and bus arbitration.

- External bus mode

  Address/Data Separate Bus mode or Multiplexed Bus mode

- Wait-state generation

  Individually programmable for each address block

  - Automatic insertion of up to seven wait cycles
  - Insertion of wait cycles through the $\overline{\text{WAIT}}$ / $\overline{\text{RDY}}$ pin

- Data bus width

  Individually programmable (8-bit or 16-bit) for each address block

- Recovery cycles (read and write)

  Individually programmable (to up to 2 cycles) for each address block. Recovery cycles are dummy cycles inserted between two consecutive external bus cycles.

- Bus arbitration

8.1 Address and Data Buses

(1) Supported configurations

The TMP1962 supports the selection of either Separate Bus mode or Multiplexed Bus mode. If the BUSMD pin (Port J1) is driven low upon reset, Separate Bus mode is selected. If the BUSMD pin is driven high upon reset, Multiplexed Bus mode is selected. For external memory interface, Port 0, Port 1, Port 2, Port 5 and Port 6 pins can be configured as the address bus, data bus or address/data bus. Table 8.1 shows the usage of the port pins in Separate and Multiplexed Bus modes.

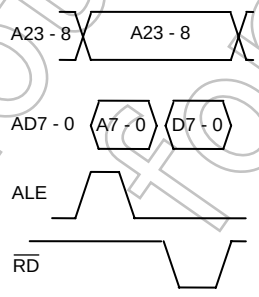
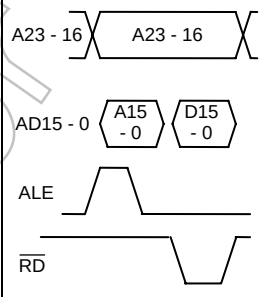
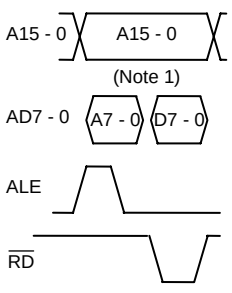
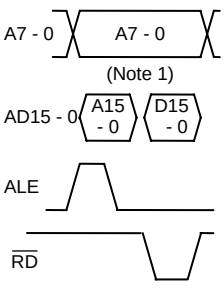
Table 8.1 Usage of Port Pins in Separate and Multiplexed Bus Modes

	Separate Bus Mode (BUSMD = L)	Multiplexed Bus Mode (BUSMD = H)
Port 0 (P00 - P07)	D0 - D7	AD0 - AD7
Port 1 (P10 - P17)	D8 - D15	AD8 - AD15/A8 - A15
Port 2 (P20 - P27)	A16 - A23	A0 - A7/A16 - A23
Port 5 (P50 - P57)	A0 - A7	General-purpose port
Port 6 (P60 - P67)	A8 - A15	General-purpose port
Port 3 (P37 only)	General-purpose port	ALE

Upon reset, all port pins are configured as general-purpose input port pins. For external memory accesses, port pins must be configured as the address or data bus through the programming of the corresponding Port Control Register (PnCR) and Port Function Register (PnFC).

In Multiplexed Bus mode, the TMP1962 supports the following four bus configurations, according to the settings in the PnCR and PnFC.

Table 8.2 Address and Data Pins in Multiplexed Bus Mode

		(1)	(2)	(3)	(4)
Address Lines		max.24 (- 16 MB)	max.24 (- 16 MB)	max.16 (- 64 KB)	max.8 (- 256 B)
Data Lines		8	16	8	16
Multiplexed Address/Data Lines		8	16	0	0
Pin Functions	Port 0	AD0 - AD7	AD0 - AD7	AD0 - AD7	AD0 - AD7
	Port 1	A8 - A15	AD8 - AD15	A8 - A15	AD8 - AD15
	Port 2	A16 - A23	A16 - A23	A0 - A7	A0 - A7
Timing Diagram					

Note 1: Because the data bus is multiplexed with the address bus, even in the (3) and (4) configurations, address bits also appear on the AD bus prior to the data being accepted or provided.

Note 2: Upon reset, all of Ports 0-2 are configured as general-purpose input ports; programming is required to use them as address or data bus pins.

Note 3: Address and data bus configurations are selectable through the programming of the P1CR, P1FC, P2CR and P2FC registers.

(2) States of the address bus during on-chip address accesses

While an on-chip address is being accessed, the address bus maintains the previous address externally presented. During this time, the data bus assumes the high-impedance state.

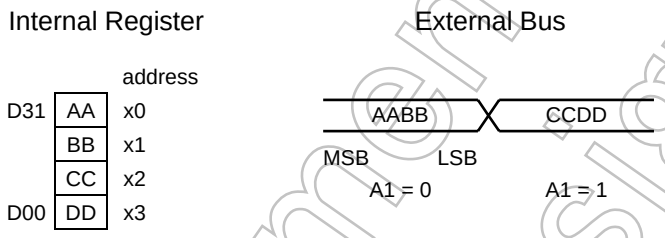
8.2 Data Formats

This section shows the relationship between the external bus interface and the TMP1962 internal register assignments.

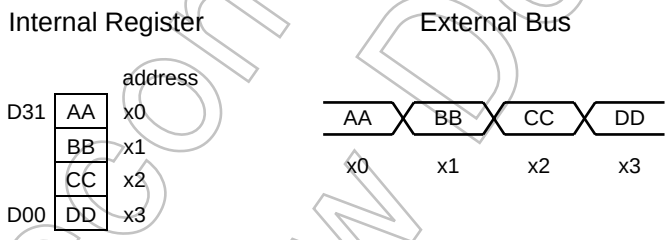
(1) Big-Endian mode

1) Word access

- 16-bit bus

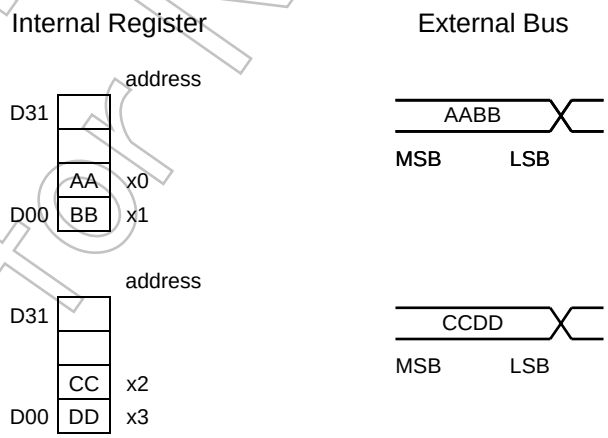


- 8-bit bus

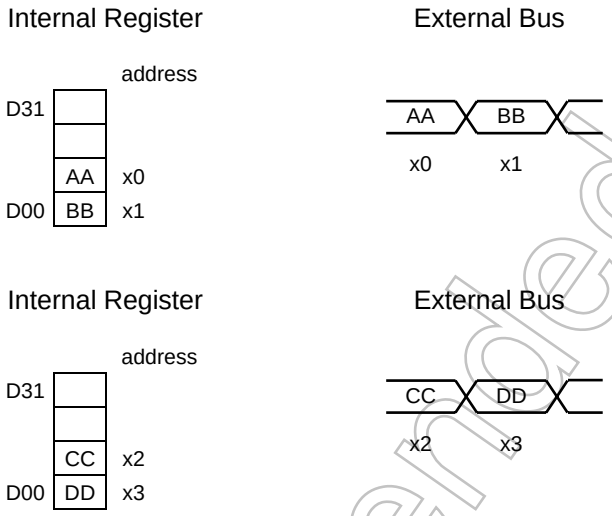


2) Halfword access

- 16-bit bus

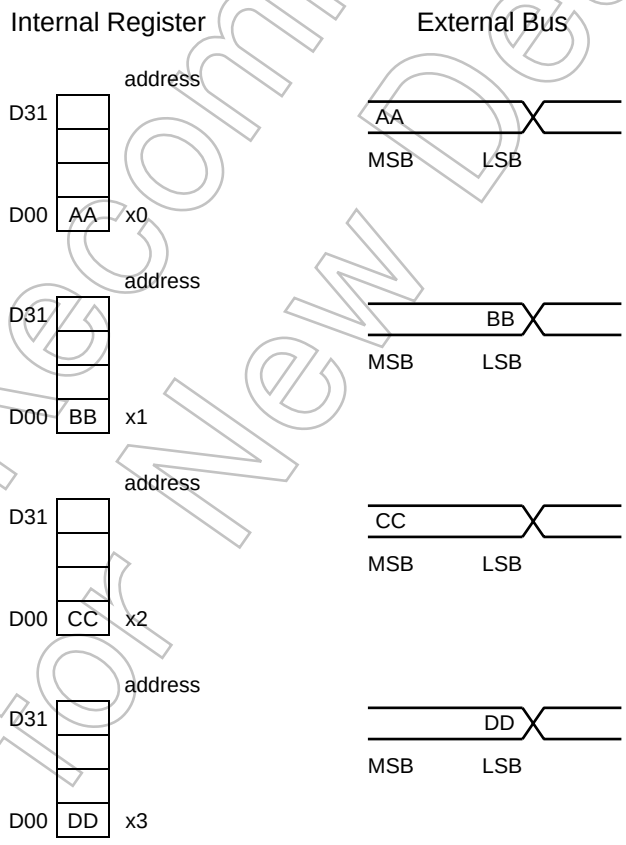


- 8-bit bus

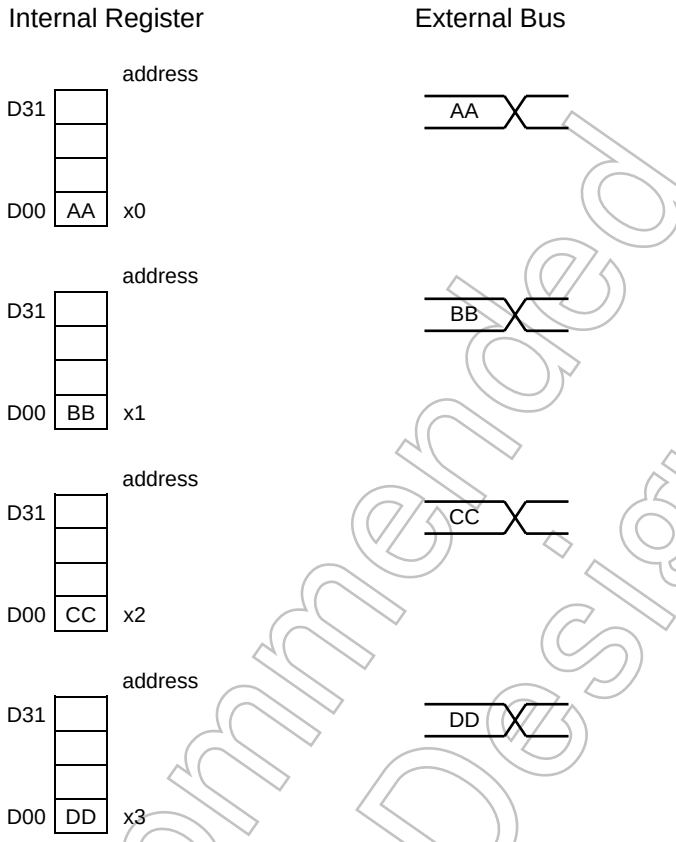


3) Byte access

- 16-bit bus



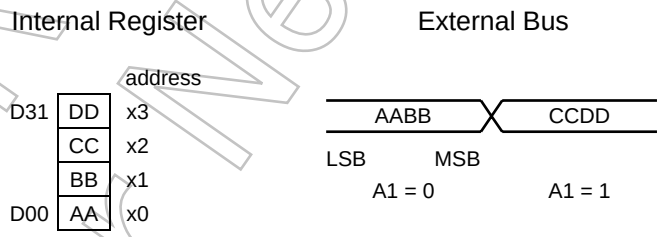
- 8-bit bus



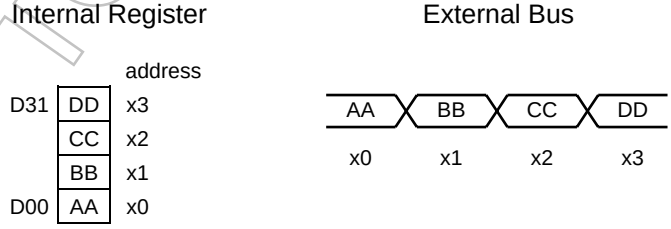
(2) Little-Endian mode

1) Word access

- 16-bit bus

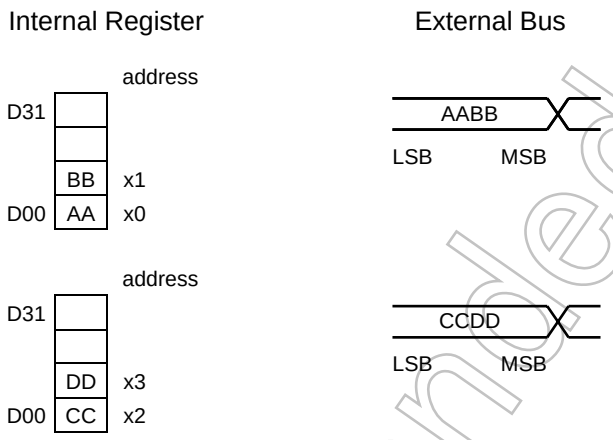


- 8-bit bus

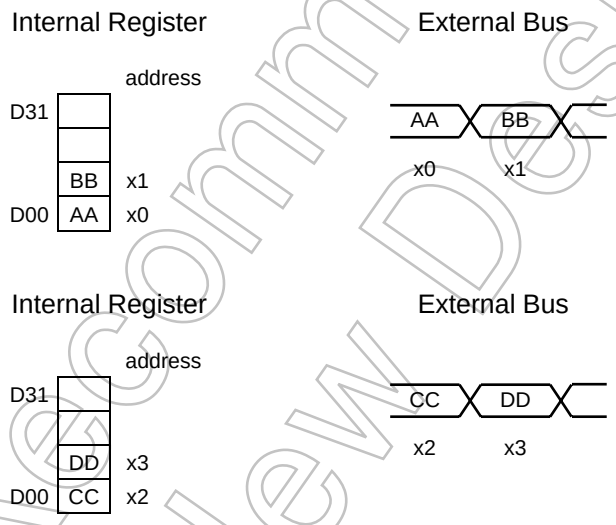


2) Halfword access

- 16-bit bus

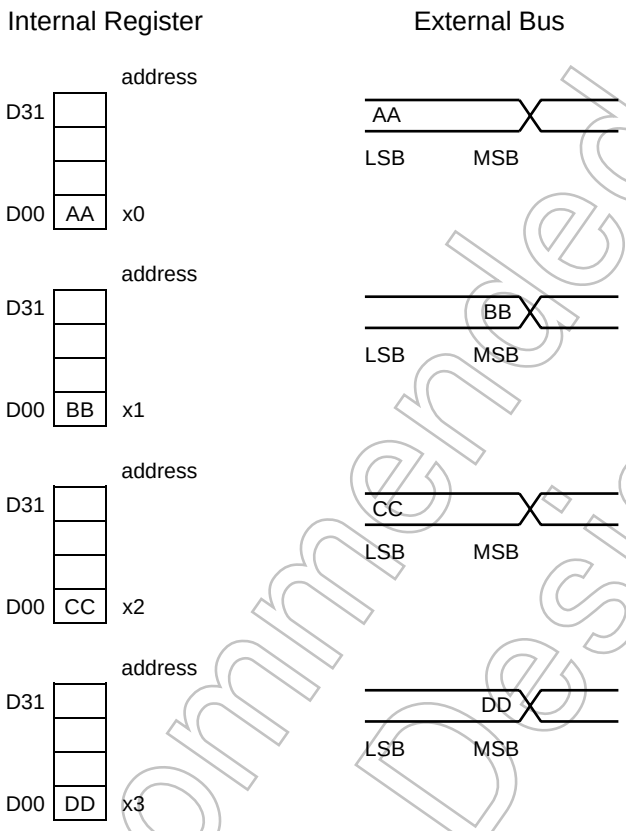


- 8-bit bus

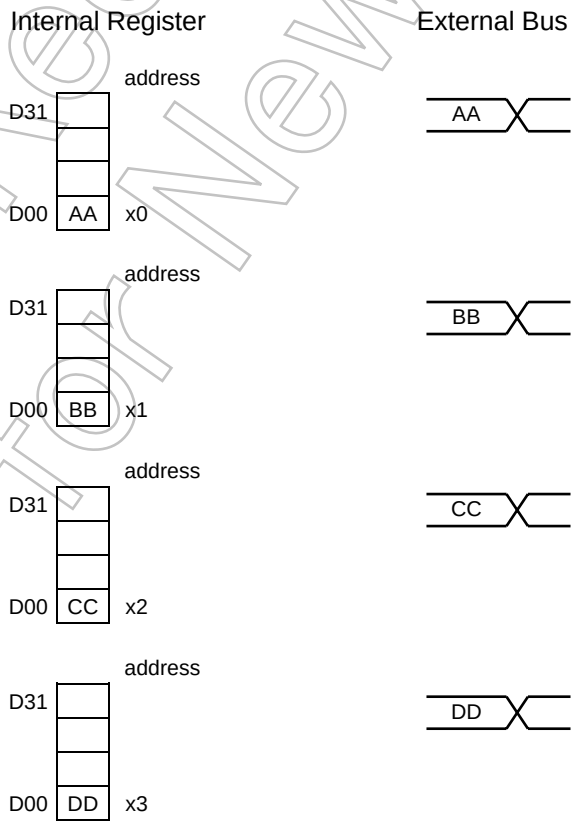


3) Byte access

- 16-bit bus



- 8-bit bus



8.3 External Bus Operation (Separate Bus Mode)

This section describes external bus operations. In the timing diagrams which follow, A23-A0 are used as the address bus, and D15-D0 are used as the data bus.

(1) Basic bus operation

While the TMP1962 provides a total of three clock cycles to perform a read or write, it also allows the bus cycle to be extended by inserting wait states. The internal system clock is also used as the basic clock for external bys cycles.

Figure 8.1 shows external bus read timing. Figure 8.2 shows external bus write timing. While an on-chip address is being accessed, the external address bus maintains the previous value. During this time, the data bus assumes the high-impedance state, and bus control signals such as $\overline{RD}$ and $\overline{WR}$ remain inactive.

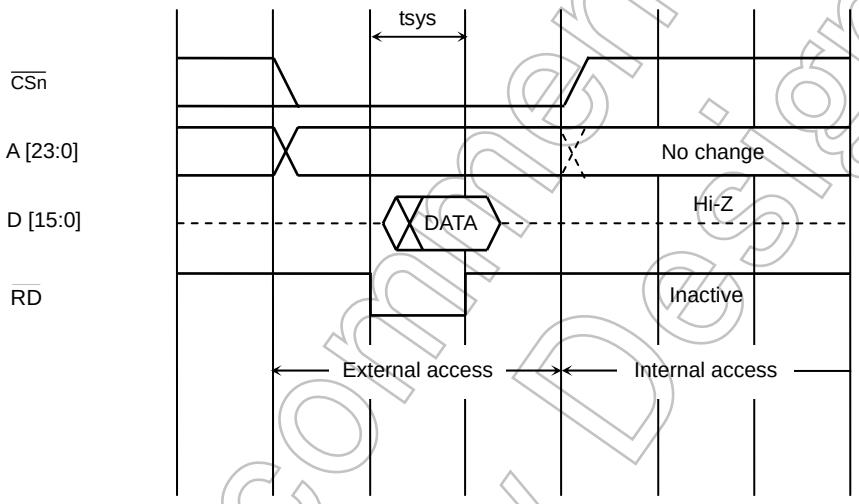


Figure 8.1 Read Cycle Timing

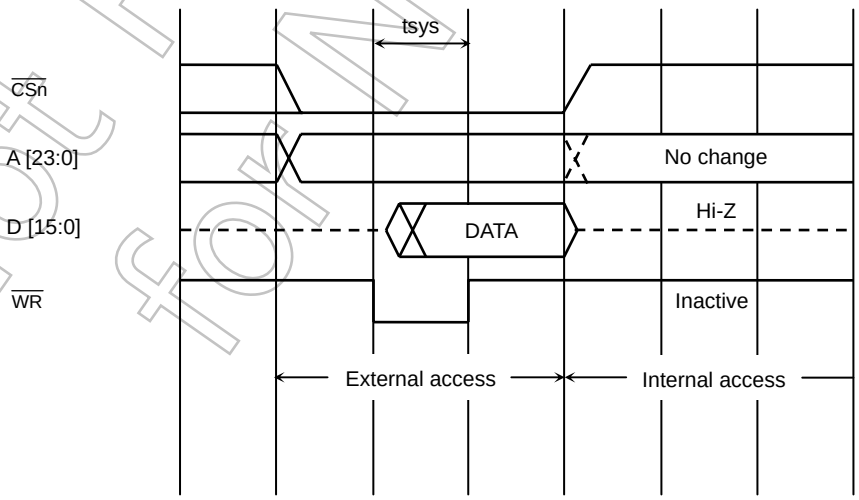


Figure 8.2 Write Cycle Timing

(2) Wait timing

The CS/Wait Controller provides three ways to insert wait states in a bus cycle for each address block:

1) Inserting required number of wait state cycles automatically (up to seven cycles)
2) Using the $\overline{\text{WAIT}}$ pin to insert wait states dynamically (1+N, 3+N, 5+N or 7+N, where N is the number of wait state cycles inserted)
3) Using the $\overline{\text{RDY}}$ pin to insert wait states dynamically (1+N, 3+N, 5+N or 7+N, where N is the number of wait state cycles inserted)

The BnW bit of the CS/Wait Control Register (BmnCS) defines the number of wait state cycles to be inserted automatically as well as external wait state input settings.

Figure 8.3 through 8.12 show bus cycle timings with wait states.

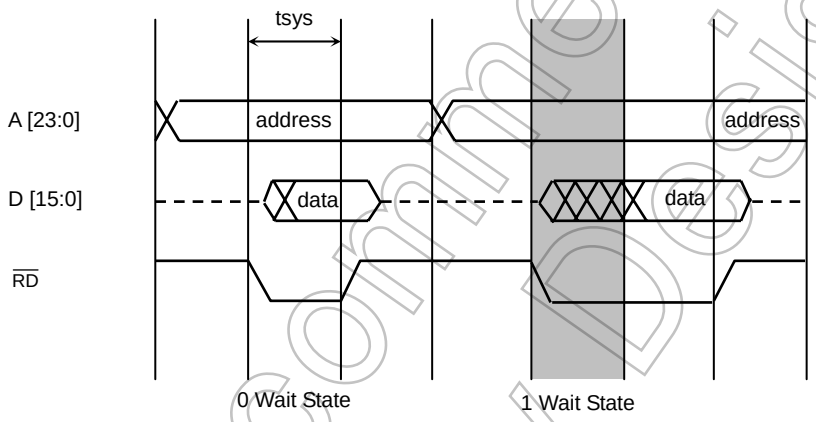


Figure 8.3 Read Cycle Timing (with Zero and Automatically Inserted One Wait State)

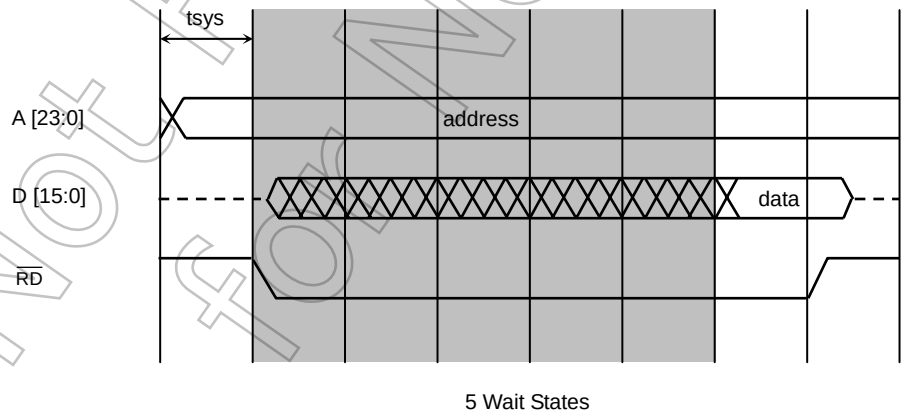


Figure 8.4 Read Cycle Timing (with Automatically Inserted Five Wait States)

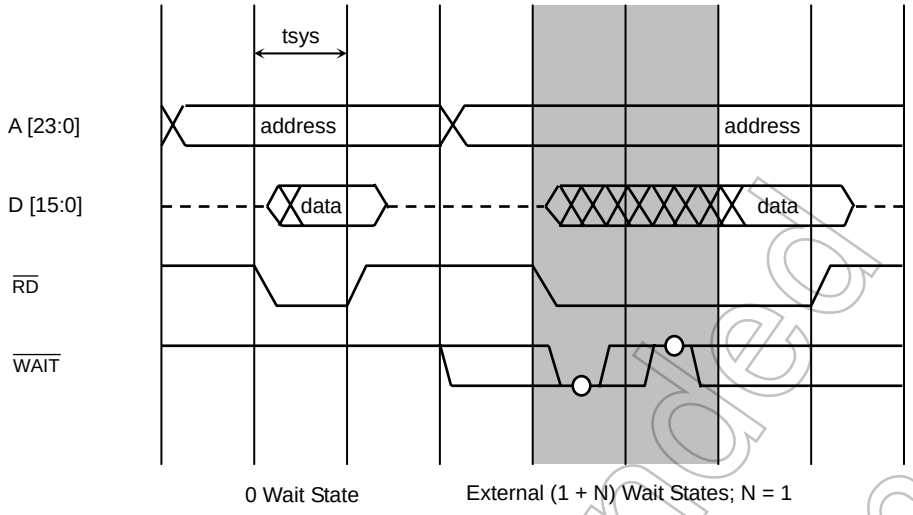


Figure 8.5 Read Cycle Timing (with Externally Inserted (1 + N) Wait States; N = 1)

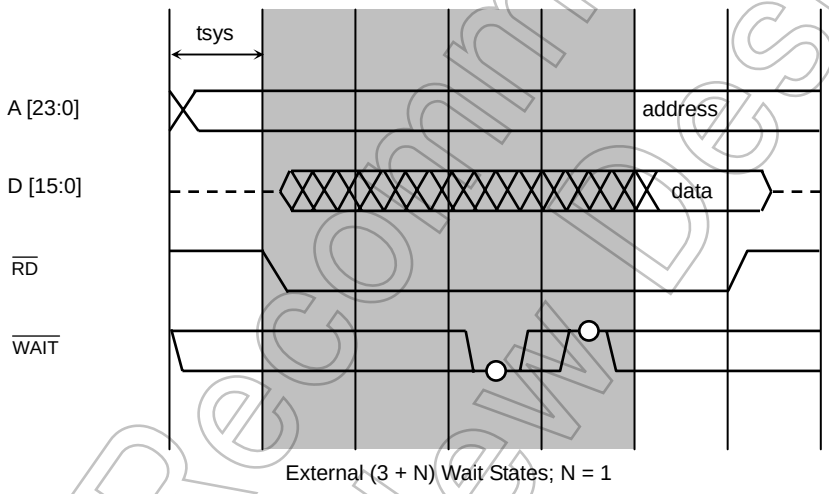


Figure 8.6 Read Cycle Timing (with Externally Inserted (3 + N) Wait States; N = 1)

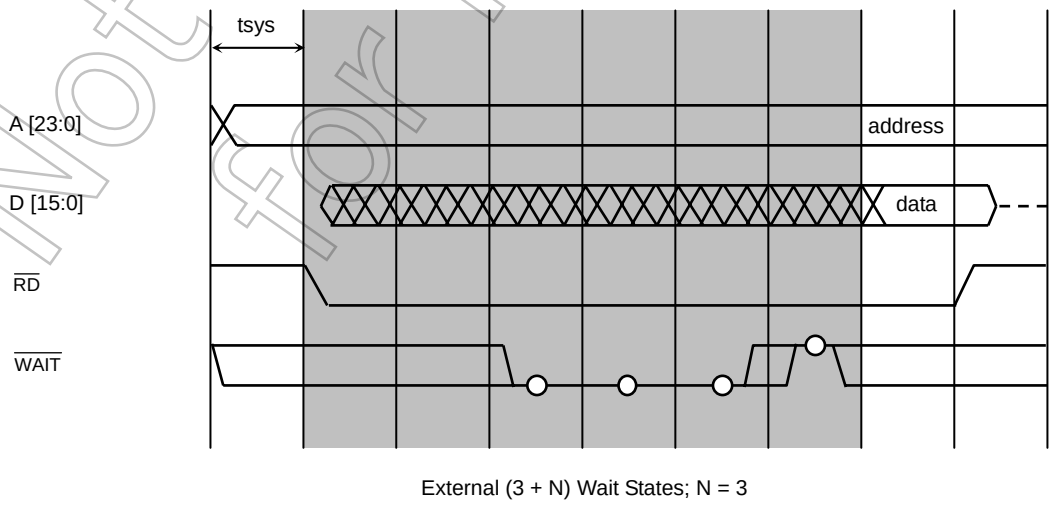


Figure 8.7 Read Cycle Timing (with Externally Inserted (3 + N) Wait States; N = 3)

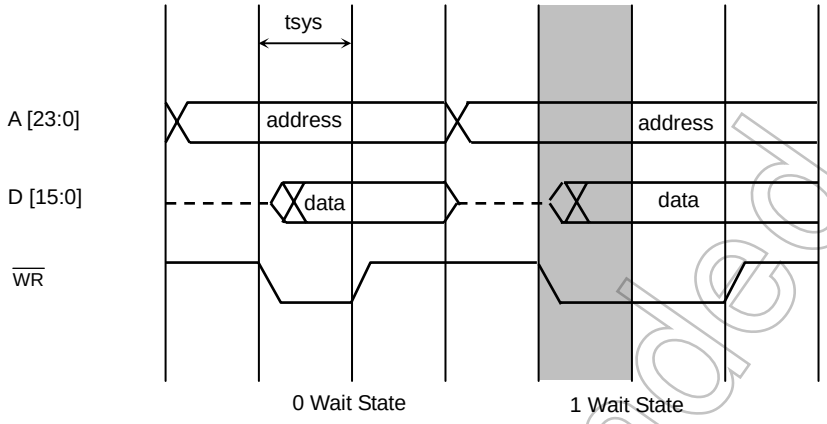


Figure 8.8 Write Cycle Timing (with Zero and Automatically Inserted One Wait State)

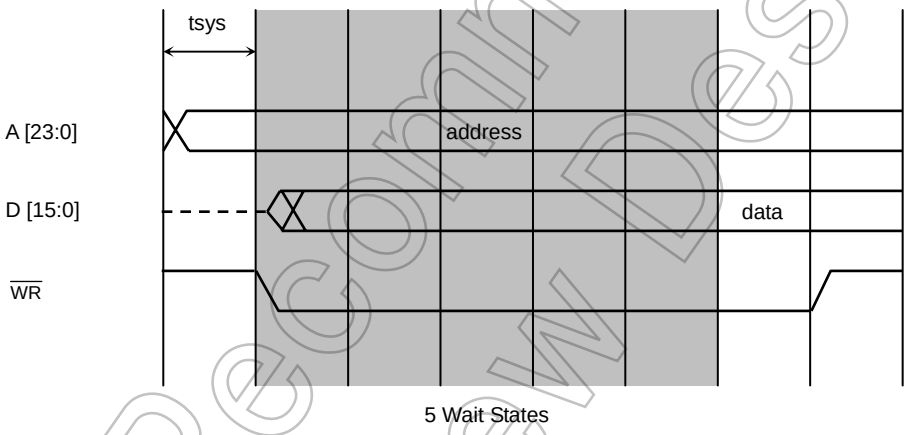


Figure 8.9 Write Cycle Timing (with Automatically Inserted Five Wait States)

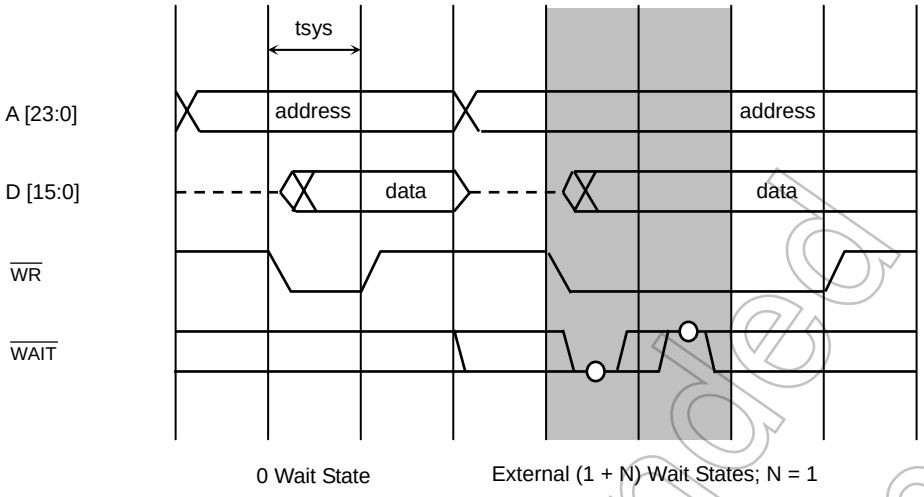


Figure 8.10 Write Cycle Timing (with Externally Inserted (1 + N) Wait States; N = 1)

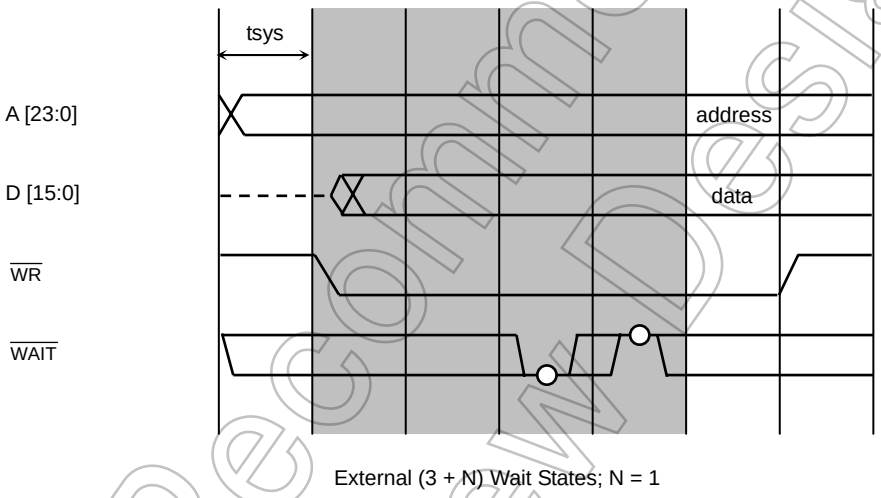


Figure 8.11 Write Cycle Timing (with Externally Inserted (3 + N) Wait States; N = 1)

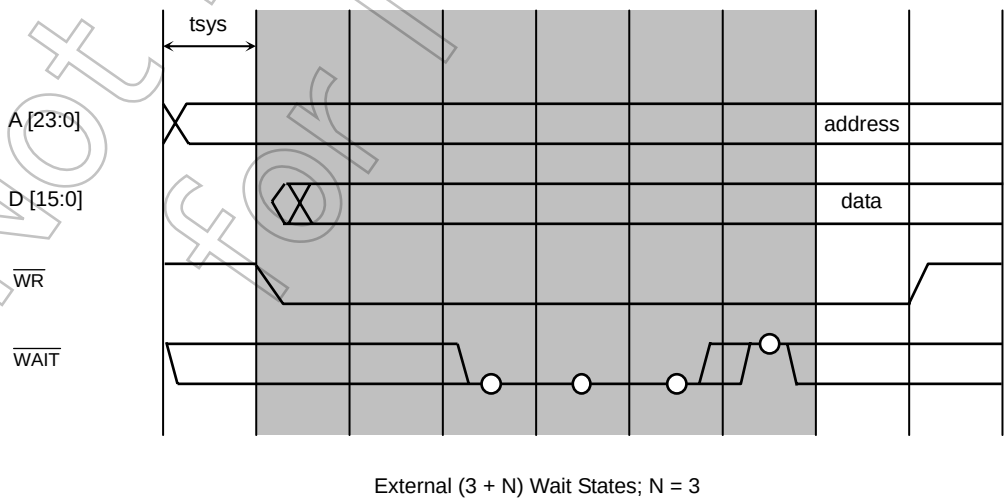


Figure 8.12 Write Cycle Timing (with Externally Inserted (3 + N) Wait States; N = 3)

Setting bit 3 (P33F) of the Port 3 Function Register (P3FC) to 1 configures the $\overline{\text{WAIT}}$ input pin (P33) as the $\overline{\text{RDY}}$ input pin.

The input supplied from the $\overline{\text{RDY}}$ pin to the external bus interface block is the logical negation of the $\overline{\text{WAIT}}$ input. The BnW bit of the CS/Wait Control Register (BmnCS) defines the number of wait state cycles to be inserted.

Figure 8.13 through 8.15 show wait states inserted with the $\overline{\text{RDY}}$ input.

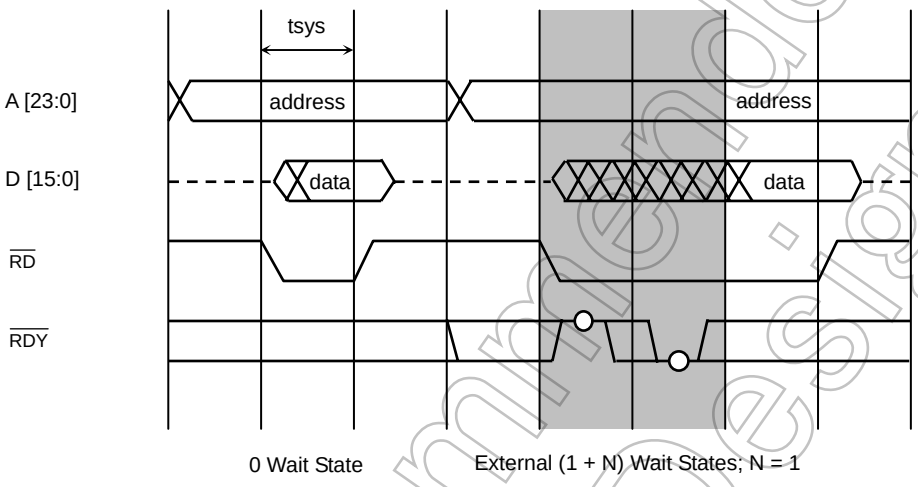


Figure 8.13 $\overline{\text{RDY}}$ Input Timing (with Externally Inserted (1 + N) Wait States; N = 1)

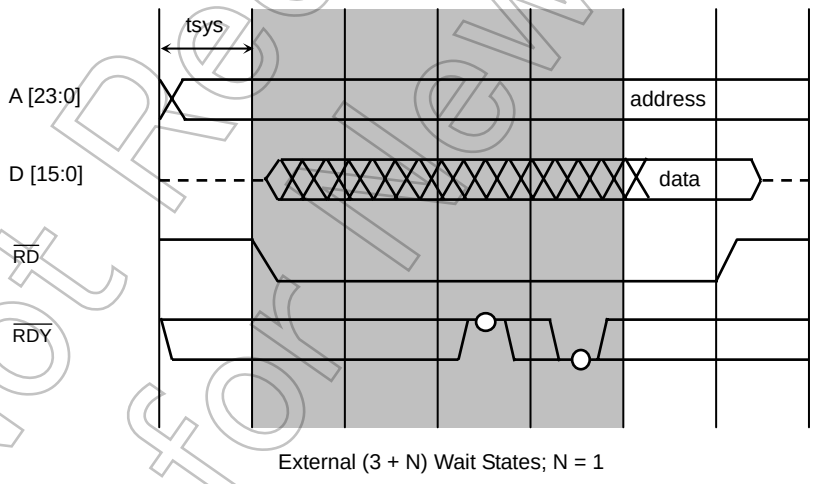


Figure 8.14 $\overline{\text{RDY}}$ Input Timing (with Externally Inserted (3 + N) Wait States; N = 1)

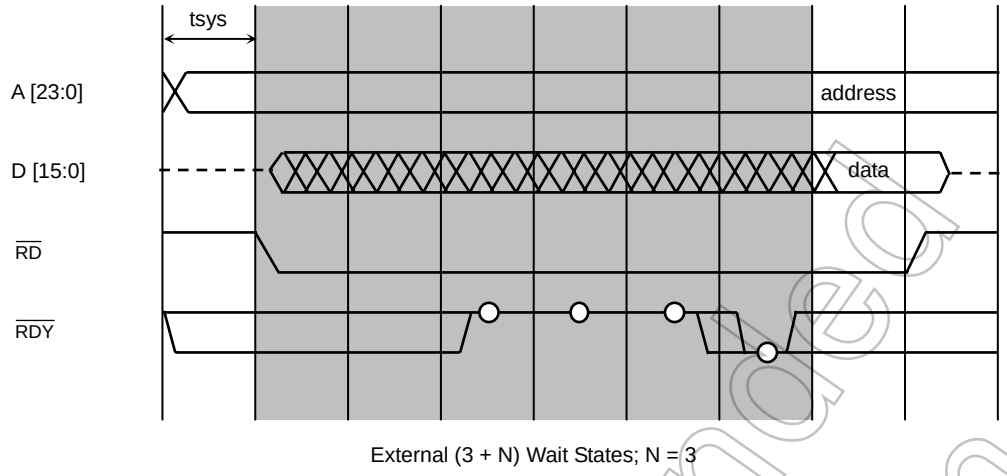


Figure 8.15 $\overline{RDY}$ Input Timing (with Externally Inserted (3 + N) Wait States; N = 3)

(3) ALE pulse width

When the TMP1962 external buses are used in Multiplexed Bus mode, the ALE pulse width can be programmed through the ALESEL bit of the SYSCR3 register within the CG. In Separate Bus mode, ALE is not asserted but the value of the SYSCR3.ALESEL bit determines the time between an address being established and $\overline{RD}$ or $\overline{WR}$ being asserted. Upon reset, ALESEL is set to 1, so that $\overline{RD}$ or $\overline{WR}$ is asserted two (internal) system clock cycles after an address is established. Clearing ALESEL to 0 causes RD or WR to be asserted one system clock cycle after an address is established. This setting applies to the whole external address space.

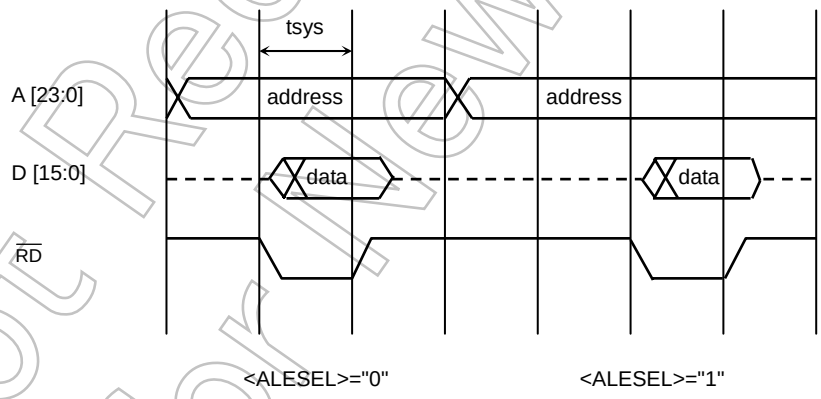


Figure 8.16 SYSCR3.ALESEL Setting and External Bus Operation

(4) Recovery time

Following an external bus cycle, a certain recovery time may be required before initiating the next external bus cycle. To allow for a recovery time, one or two dummy cycles can be inserted between back-to-back bus cycles.

Dummy cycles can be inserted either after a read cycle or a write cycle. Dummy cycle insertion is programmable with the BnWCV (write recovery cycle) and BnRCV (read recovery cycle) bits of the CS/Wait Control Register (BmnCS). The number of dummy cycles (one or two internal system clock cycles) can be specified for each block. Figure 8.17 shows timing with a recovery time inserted.

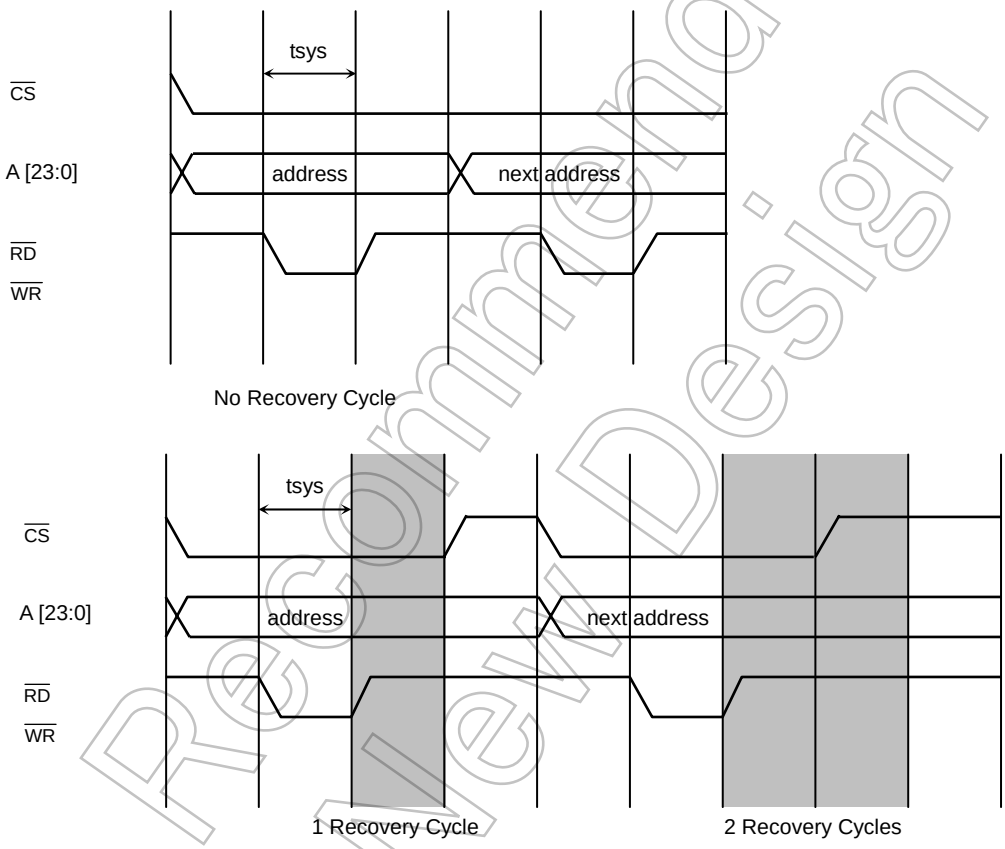


Figure 8.17 Timing with a Recovery Time Inserted

8.4 External Bus Operation (Multiplexed Bus Mode)

This section describes external bus operations. In the timing diagrams which follow, A23-A16 are used as the address bus, and AD15-AD0 are used as the address/data bus.

(1) Basic bus operation

While the TMP1962 provides a total of three clock cycles to perform a read or write, it also allows the bus cycle to be extended by inserting wait states. The internal system clock is also used as the basic clock for external bys cycles.

Figure 8.18 shows external bus read timing. Figure 8.19 shows external bus write timing. While an on-chip address is being accessed, the external address bus maintains the previous value with the ALE pin kept inactive. During this time, the address/data bus assumes the high-impedance state, and bus control signals such as $\overline{RD}$ and $\overline{WR}$ remain inactive.

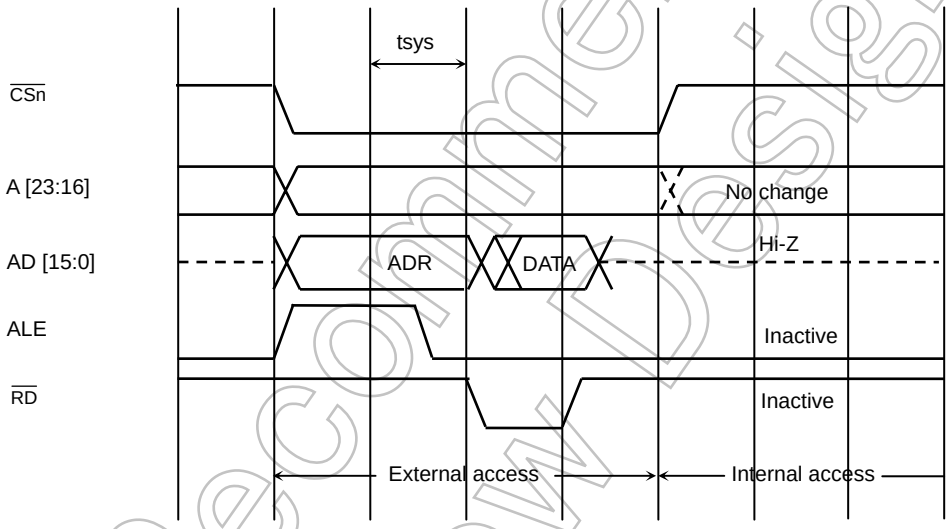


Figure 8.18 Read Cycle Timing

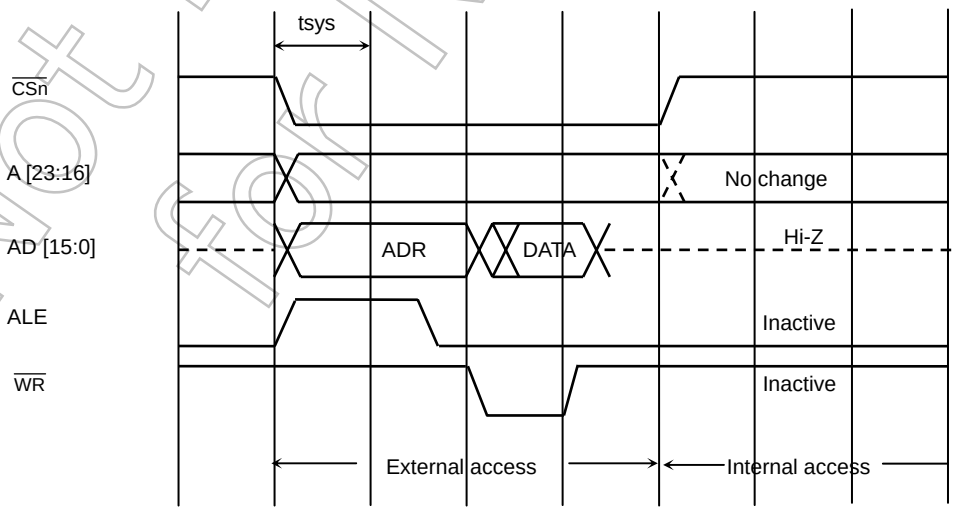


Figure 8.19 Write Cycle Timing

(2) Wait timing

The CS/Wait Controller provides three ways to insert wait states in a bus cycle for each address block:

1) Inserting required number of wait state cycles automatically (up to seven cycles)
2) Using the $\overline{\text{WAIT}}$ pin to insert wait states dynamically (1+N, 3+N, 5+N or 7+N, where N is the number of wait state cycles inserted)
3) Using the $\overline{\text{RDY}}$ pin to insert wait states dynamically (1+N, 3+N, 5+N or 7+N, where N is the number of wait state cycles inserted)

The BnW bit of the CS/Wait Control Register (BmnCS) defines the number of wait state cycles to be inserted automatically as well as external wait state input settings.

Figure 8.20 through 8.29 show bus cycle timings with wait states.

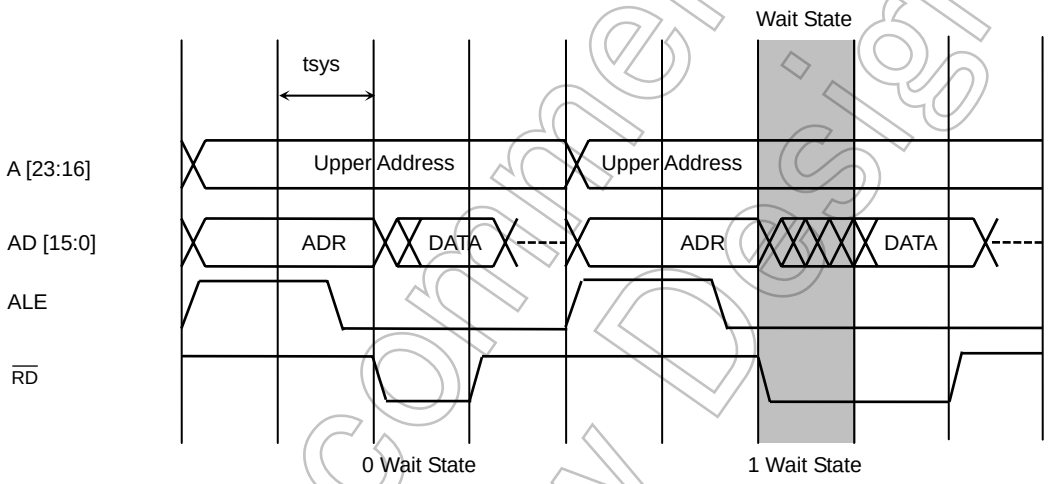


Figure 8.20 Read Cycle Timing (with Zero and One Wait State)

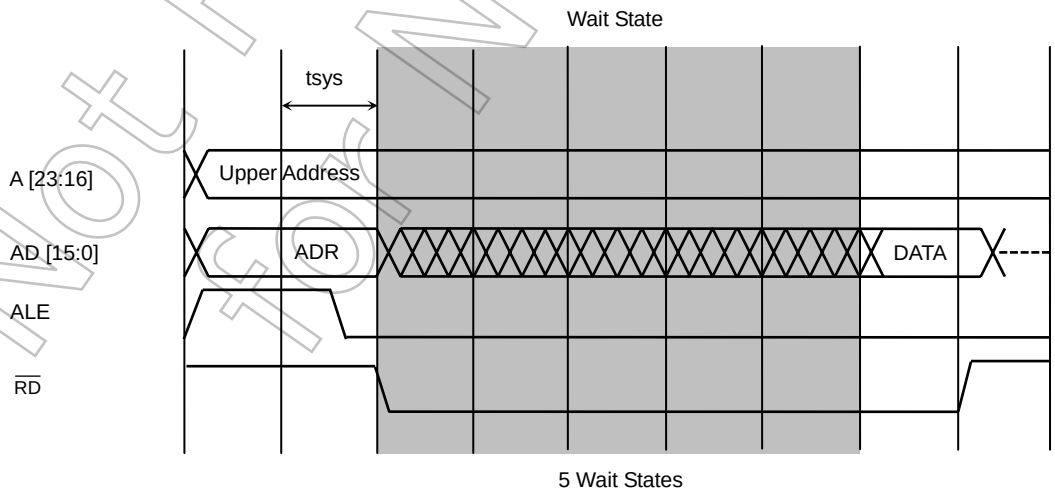


Figure 8.21 Read Cycle Timing (with Five Wait States)

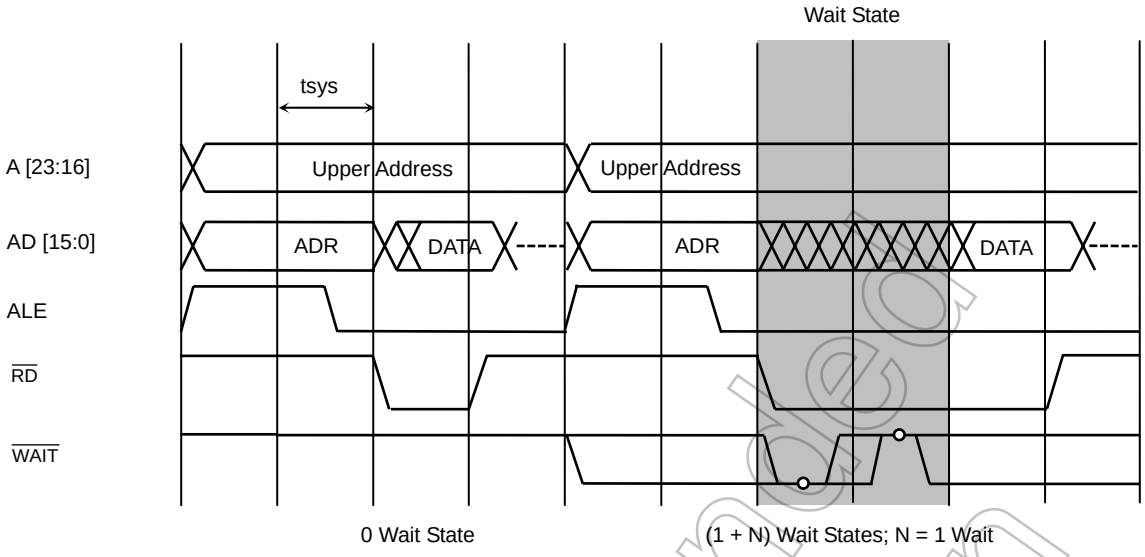


Figure 8.22 Read Cycle Timing (with (1 + N) Wait States; N = 1)

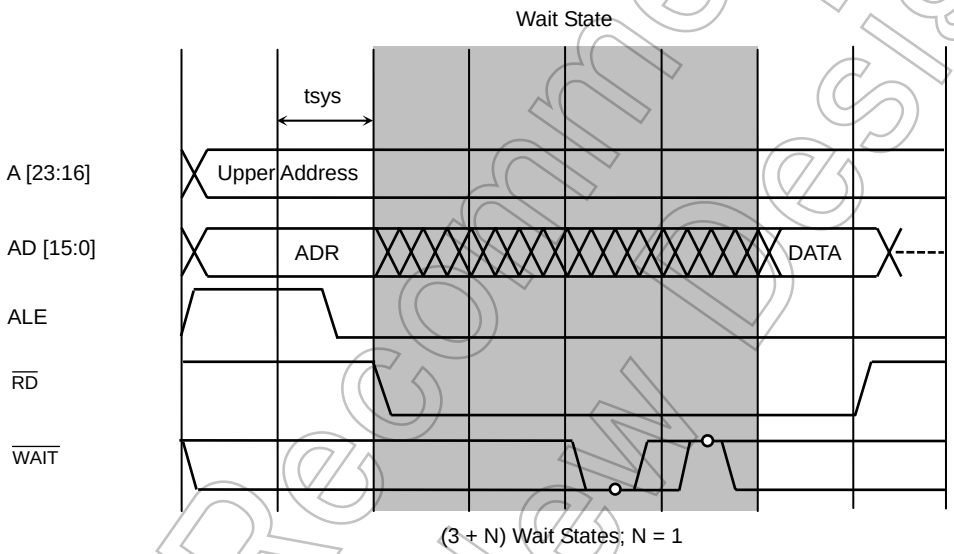


Figure 8.23 Read Cycle Timing (with (3 + N) Wait States; N = 1)

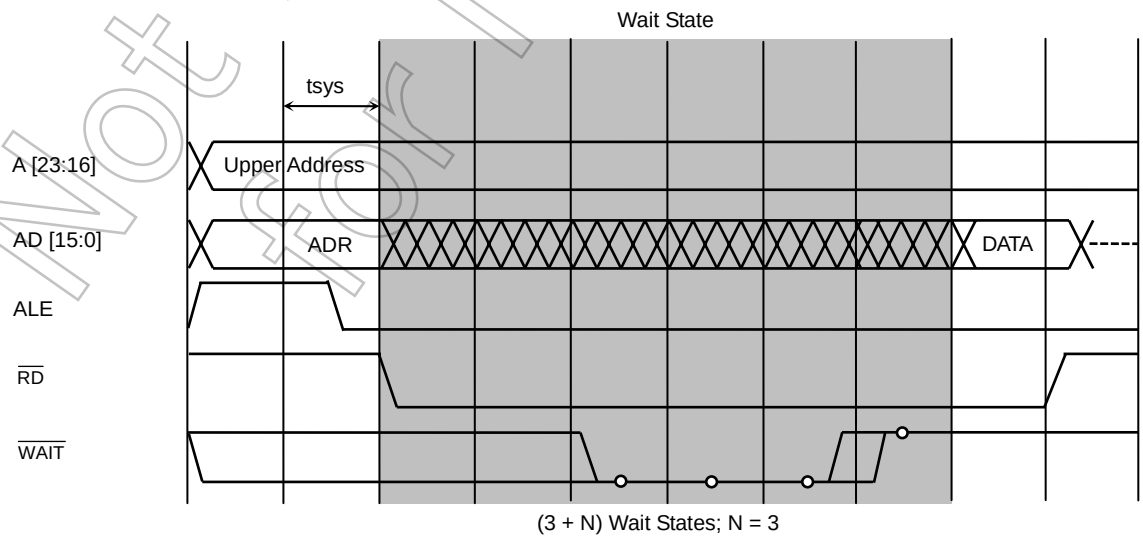


Figure 8.24 Read Cycle Timing (with (3 + N) Wait States; N = 3)

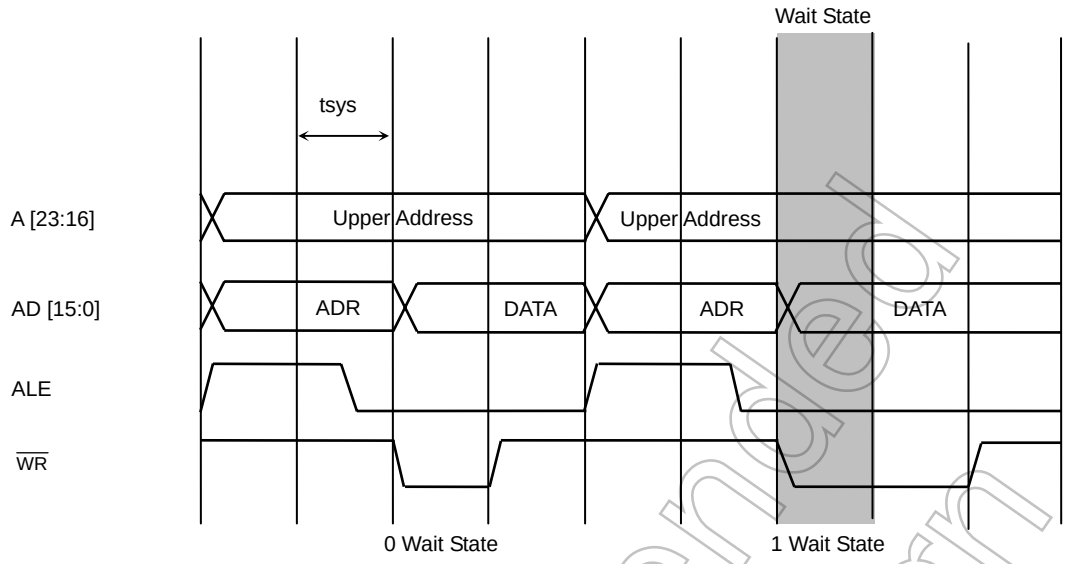


Figure 8.25 Write Cycle Timing (with Zero and One Wait State)

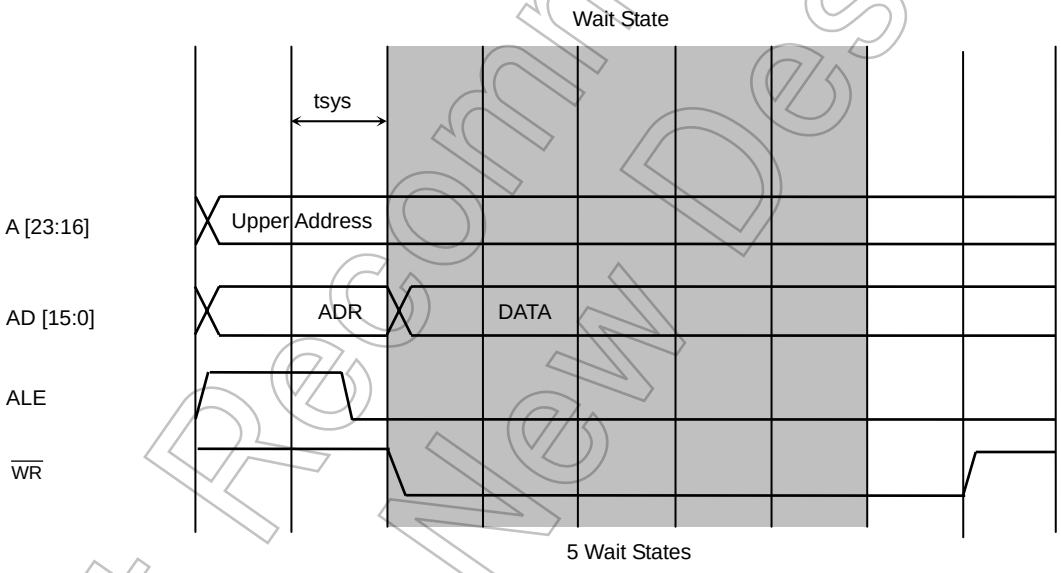


Figure 8.26 Write Cycle Timing (with Five Wait States)

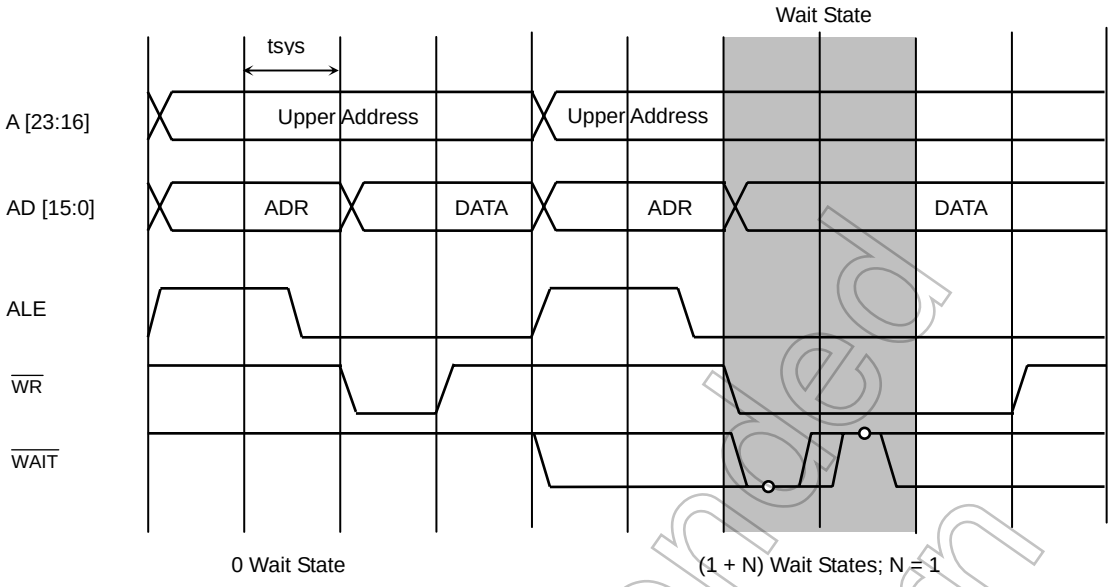


Figure 8.27 Write Cycle Timing (with (1 + N) Wait States; N = 1)

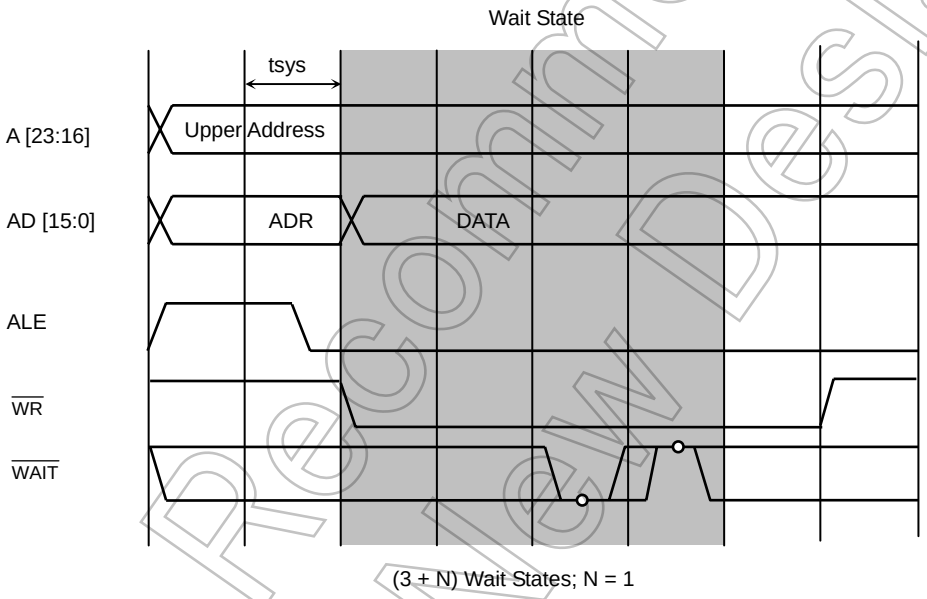


Figure 8.28 Write Cycle Timing (with (3 + N) Wait States; N = 1)

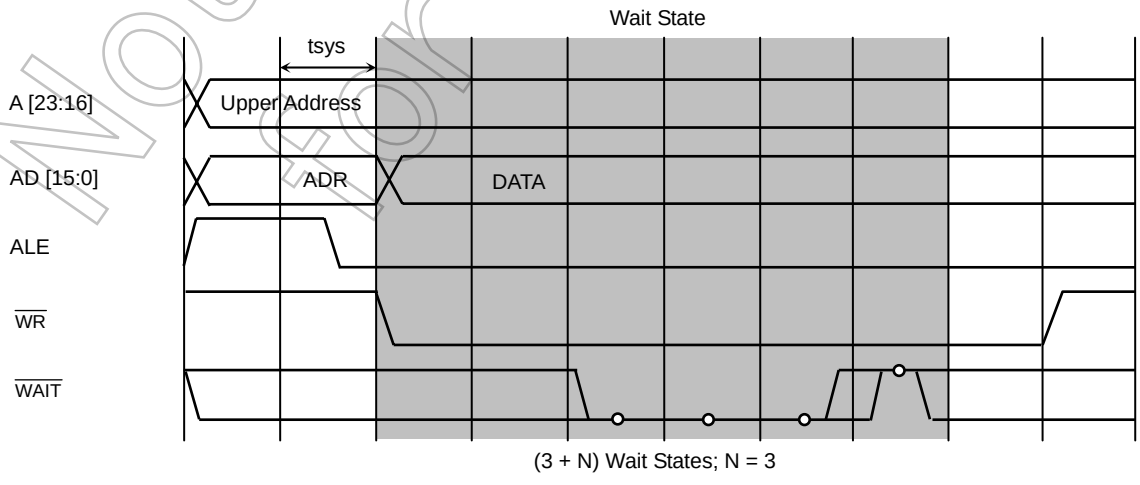


Figure 8.29 Write Cycle Timing (with (3 + N) Wait States; N = 3)

(3) ALE pulse width

The ALE pulse width is programmed to 0.5 or 1.5 clock cycles through the ALESEL bit of the SYSCR3 register within the CG. The default is 1.5 cycles. This setting applies to the whole external address space.

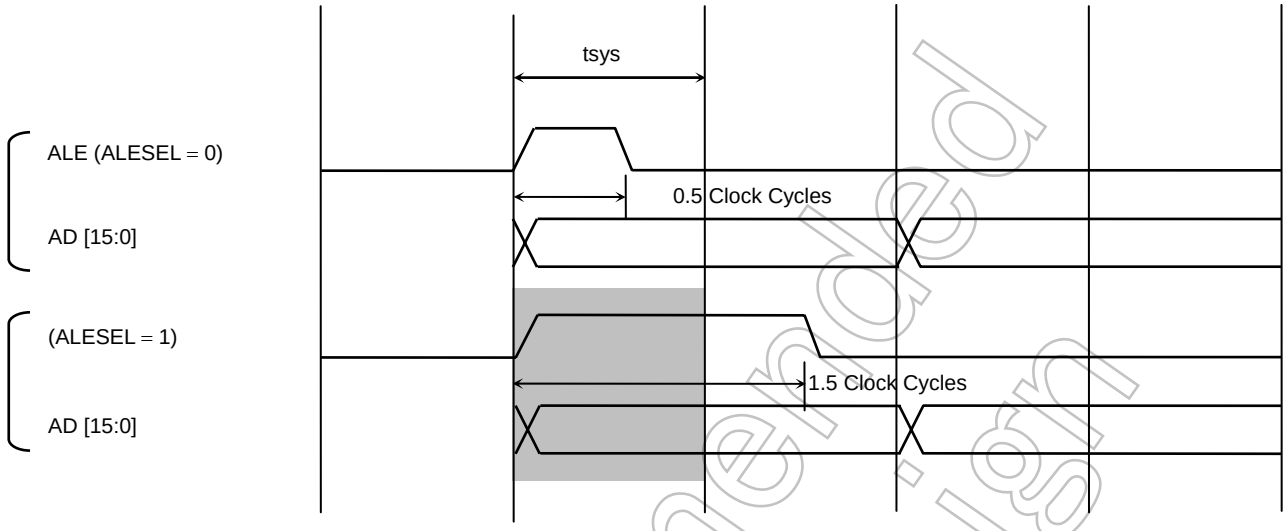


Figure 8.30 ALE Pulse Width

Figure 8.31 shows read cycle timing, with the ALE width programmed to 0.5 and 1.5 clock cycles.

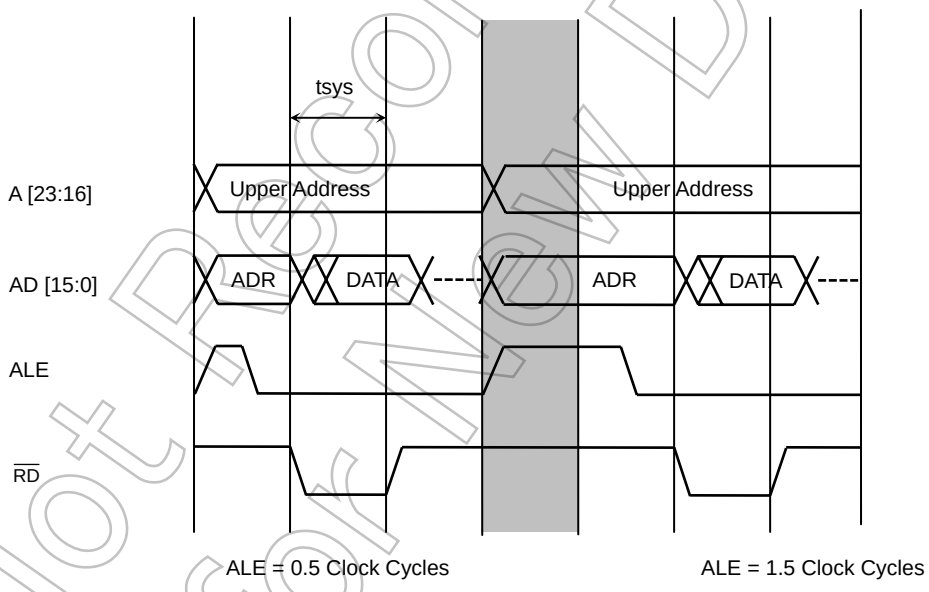


Figure 8.31 Read Cycle Timing (ALE = 0.5 and 1.5 Clock Cycles)

Note: In the TMP1962F10, configuring the ALE pulse width as 0.5 or 1.5 clock cycles results in the actual ALE pulse width being 1.5 or 2.5 clock cycles, respectively.

(4) Recovery time

Following an external bus cycle, a certain recovery time may be required before initiating the next external bus cycle. To allow for a recovery time, one or two dummy cycles can be inserted between back-to-back bus cycles.

Dummy cycles can be inserted either after a read cycle or a write cycle. Dummy cycle insertion is programmable with the BnWCV (write recovery cycle) and BnRCV (read recovery cycle) bits of the CS/Wait Control Register (BmnCS). The number of dummy cycles (one or two internal system clock cycles) can be specified for each block. Figure 8.32 shows timing with a recovery time inserted.

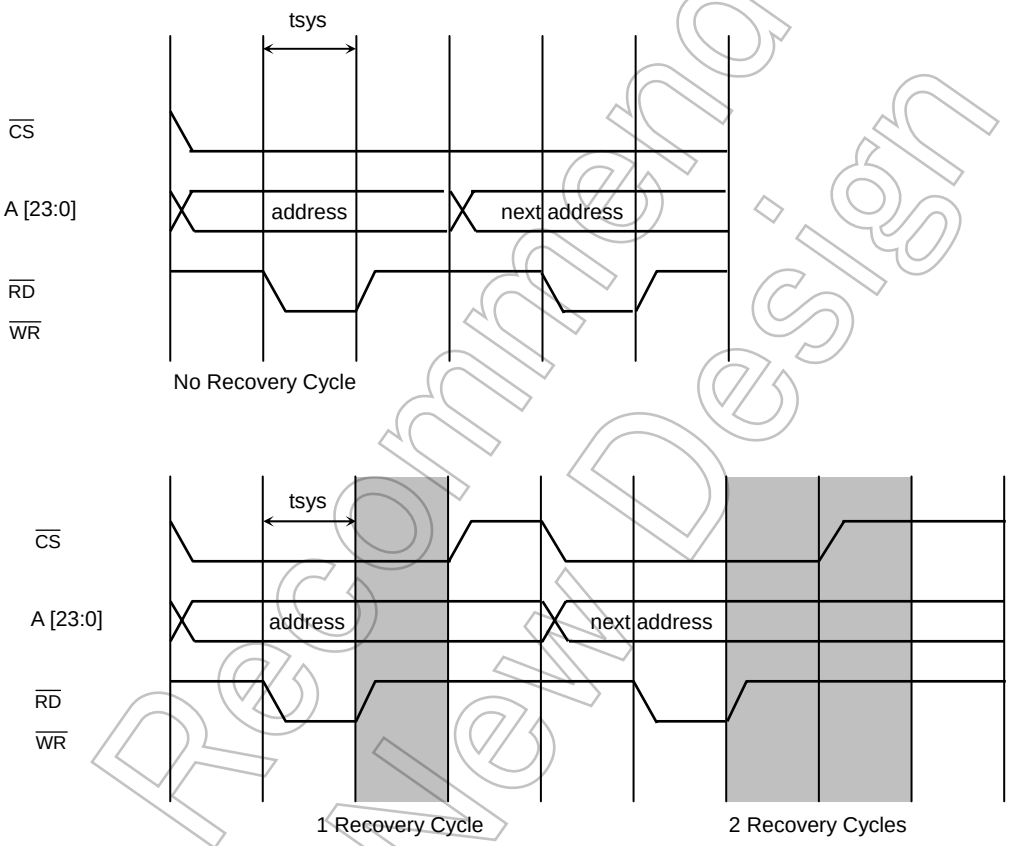


Figure 8.32 Timing with a Recovery Time Inserted

8.5 Bus Arbitration

The TMP1962 provides support for an external bus master to take control of the external bus. Two bus arbitration control signals, $\overline{\text{BUSRQ}}$ and $\overline{\text{BUSAK}}$, are used to determine the bus master. One or more of the external devices on the bus can have the capability of becoming bus master for the external bus, but not the TMP1962 internal bus.

(1) Bus access control

External bus masters can gain control of the external bus, but not the TMP1962 internal bus (G-Bus). Thus, external bus masters cannot access the TMP1962's on-chip memory and peripherals. The External Bus Interface (EBIF) logic in the TMP1962 manages the arbitration of the external bus; the CPU and on-chip DMAC do not participate in any way in this bus arbitration. During external bus mastership, the CPU and the on-chip DMAC can access the internal memory (RAM and ROM) and registers. Once an external device assumes bus mastership, the CPU or the on-chip DMAC has no way to regain the bus until the external bus master releases the bus. If the CPU or the on-chip DMAC issues an external memory access request, it is forced to wait until the TMP1962 regains the bus. Therefore, should $\overline{\text{BUSRQ}}$ be left asserted for a long time, the TMP1962 might suffer system lockups.

(2) Bus arbitration flow

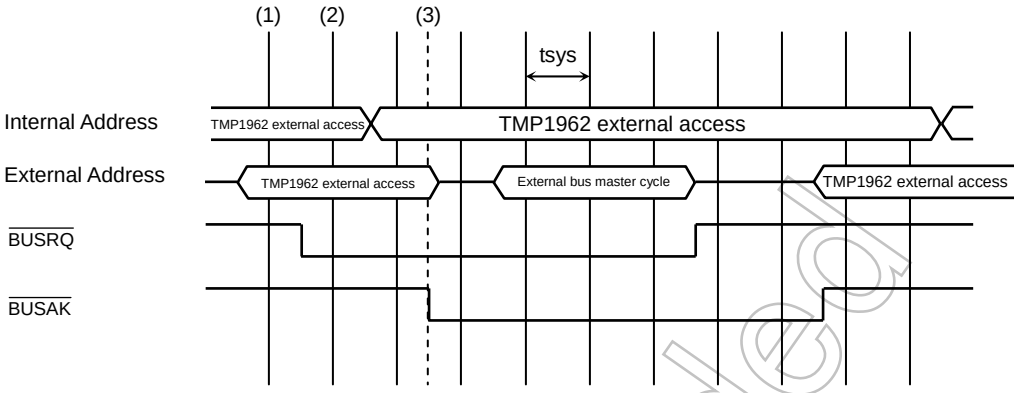
External devices capable of becoming bus masters assert $\overline{\text{BUSRQ}}$ to request the bus. The TMP1962 samples $\overline{\text{BUSRQ}}$ at the end of each external bus cycle, as seen on its internal bus (GBus). When the TMP1962 has made an internal decision to grant the bus, it asserts $\overline{\text{BUSAK}}$ to indicate to the requesting device that the bus is available. At the same time, the TMP1962 puts the address bus, the data bus and bus control signals in the high-impedance state.

A load or store may require multiple bus cycles, depending on the port size of the addressed device (dynamic bus sizing). In that case, the TMP1962 does not grant the bus until the entire transfer is complete.

The TMP1962, if so programmed, automatically inserts dummy cycles between back-to-back bus cycles to allow for sufficient read recovery time. In dummy cycles, the TMP1962 has already internally initiated a bus cycle on the G-Bus for the next external access. The TMP1962 can only accept an external bus request at the boundary of an internal G-Bus bus cycle. Therefore, if $\overline{\text{BUSRQ}}$ is asserted during a dummy cycle, the TMP1962 grants the bus after it completes the next external bus cycle.

An external bus master must keep $\overline{\text{BUSRQ}}$ asserted until it is granted the bus.

A timing diagram of the bus arbitration sequence is shown in Figure 8.33.



1) $\overline{\text{BUSRQ}}$ is sampled high.
2) The TMP1962 recognizes the assertion of $\overline{\text{BUSRQ}}$.
3) The TMP1962 asserts $\overline{\text{BUSAK}}$ at the completion of the current bus cycle. The external bus master recognizes $\overline{\text{BUSAK}}$ and assumes bus mastership to start a bus transfer.

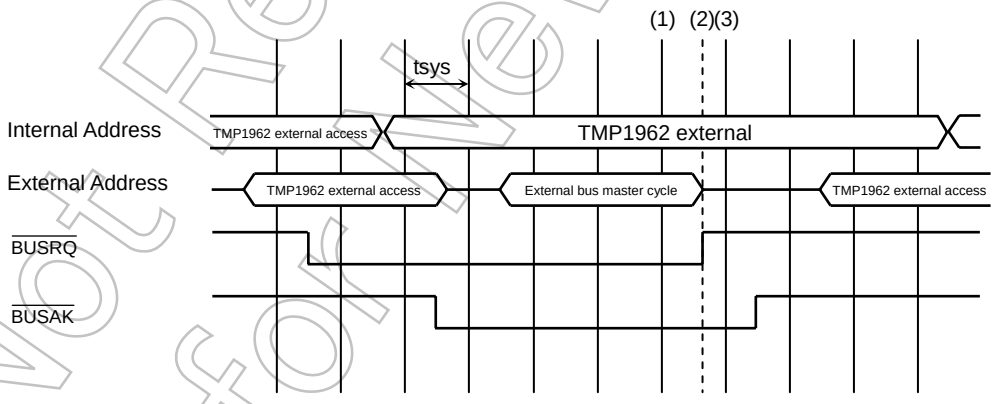
Figure 8.33 Bus Arbitration Timing Diagram

(3) Relinquishing the bus

The external bus master relinquishes the bus when it has completed its bus transactions and no longer requires the bus.

When the external bus master has completed its bus transactions, it de-asserts $\overline{\text{BUSRQ}}$ to relinquish the bus to the TMP1962.

Figure 8.34 shows the timing for an external bus master to relinquish the bus.



1) The external bus master has control of the bus.
2) When the external bus master no longer needs the bus, it de-asserts $\overline{\text{BUSRQ}}$.
3) In response to the de-assertion of $\overline{\text{BUSRQ}}$, the TMP1962 de-asserts $\overline{\text{BUSAK}}$.

Figure 8.34 External Bus Master Relinquishing the Bus

9. Chip Select/Wait Controller

The TMP1962 supports direct connections to external devices (I/O devices, ROM and SRAM).

The TMP1962 provides four programmable chip select signals. Programmable features include variable block sizes, data bus width, wait state insertion, and dummy cycle insertion for back-to-back bus cycles.

$\overline{CS0}$ - $\overline{CS3}$ (multiplexed with P40-P43) are the chip select output pins for the CS0-CS3 address ranges. These chip select signals are generated when the CPU or on-chip DMAC issues an address within the programmed ranges. The P40-P43 pins must be configured as CS0-CS3 by programming the Port 4 Control (P4CR) register and the Port 4 Function (P4FC) register.

Chip select address ranges are defined in terms of a base address and an address mask. There is a Base/Mask Address (BMAn) register for each of the four chip select signals, where n is a number from 0 to 3.

There is also a set of three Chip Select/Wait Control registers, B01CS, B23CS and BEXCS, each of which consists of a master enable bit, a data bus width bit, a wait state field and a dummy cycle field.

External memory devices can also use the $\overline{WAIT}$ pin to insert wait states and consequently prolong read and write bus cycles.

9.1 Programming Chip Select Ranges

Each of the four chip select address ranges is defined in the BMAn register. The basic chip select model allows one of the chip select output signals ($\overline{CS0}$ - $\overline{CS3}$) to assert when an address on the address bus falls within a particular programmed range. The B01CS register defines specific operations for CS0 and CS1, and the B23CS register defines specific operations for CS2 and CS3 (see Section 9.2).

9.1.1 Base/Mask Address Registers

The organizations of the BMAn registers are shown in Figure 9.1 and Figure 9.2. The base address (BAn) field specifies the starting address for a chip select. Any set bit in the address mask field (MAn) masks the corresponding base address bit. The address mask field determines the block size of a particular chip select line. The address is compared on every bus cycle.

(1) Base address

The base address (BAn) field specifies the upper 16 bits (A31-A16) of the starting address for a chip select. The lower 16 bits (A15-A0) are assumed to be zero. Thus, the base address is any multiple of 64 Kbytes starting at 0x0000_0000. Figure 9.3 shows the relationships between starting addresses and the BMAn values.

(2) Address mask

The address mask (MAn) field defines whether any particular bits of the address should be compared or masked. Any set bit masks the corresponding base address bit. The address compare logic uses only the address bits that are not masked (i.e., mask bit cleared to 0) to detect an address match.

Address bits that can be masked (i.e., supported block sizes) differ for the four chip select spaces as follows:

CS0 and CS1 spaces: A29-A14

CS2 and CS3 spaces: A30-A15

Note: Use physical addresses in the BMAn registers.

Base/Mask Address Registers: BMA0 (0xFFFF_E400) to BMA3 (0xFFFF_E40C)

BMA0 (0xFFFF_E400)

	31	30	29	28	27	26	25	24
Bit Symbol	BA0							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31-A24 of the starting address							
	23	22	21	20	19	18	17	16
Bit Symbol	BA0							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23-A16 of the starting address							
	15	14	13	12	11	10	9	8
Bit Symbol	MA0							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	1	1
Function	Must be written as 0.							
	7	6	5	4	3	2	1	0
Bit Symbol	MA0							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS0 block size 0: The address compare logic uses this address bit.							

BMA1 (0xFFFF_E404)

	31	30	29	28	27	26	25	24
Bit Symbol	BA1							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31-A24 of the starting address							
	23	22	21	20	19	18	17	16
Bit Symbol	BA1							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23-A16 of the starting address							
	15	14	13	12	11	10	9	8
Bit Symbol	MA1							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	1	1
Function	Must be written as 0.							
	7	6	5	4	3	2	1	0
Bit Symbol	MA1							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS1 block size 0: The address compare logic uses this address bit.							

Note: Bits 10-15 in the BMA0 and BMA1 must be written as zeros. The CS0 and CS1 block sizes can vary from 16 Kbytes to 1 Gbyte. However, the TMP1962 supports only 16 Mbytes of external address space. Therefore, bits 10-15 in the BMA0 and BMA1 must be cleared so that A24-A29 of an address will not be masked.

Figure 9.1 Base/Mask Address Registers (BMA0 and BMA1)

BMA2 (0xFFFF_E408)

	31	30	29	28	27	26	25	24
Bit Symbol	BA2							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31-A24 of the starting address							
	23	22	21	20	19	18	17	16
Bit Symbol	BA2							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23-A16 of the starting address							
	15	14	13	12	11	10	9	8
Bit Symbol	MA2							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	1
Function	Must be written as 0.							
	7	6	5	4	3	2	1	0
Bit Symbol	MA2							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS2 block size 0: The address compare logic uses this address bit.							

BMA3 (0xFFFF_E40C)

	31	30	29	28	27	26	25	24
Bit Symbol	BA3							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31-A24 of the starting address							
	23	22	21	20	19	18	17	16
Bit Symbol	BA3							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23-A16 of the starting address							
	15	14	13	12	11	10	9	8
Bit Symbol	MA3							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	1
Function	Must be written as 0.							
	7	6	5	4	3	2	1	0
Bit Symbol	MA3							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS3 block size 0: The address compare logic uses this address bit.							

Note: Bits 9-15 in the BMA2 and BMA3 must be written as zeros. The CS2 and CS3 block sizes can vary from 32 Kbytes to 2 Gbytes. However, the TMP1962 supports only 16 Mbytes of external address space. Therefore, bits 9-15 in the BMA2 and BMA3 must be cleared so that A24-A30 of an address will not be masked.

Figure 9.2 Base/Mask Address Registers (BMA2 and BMA3)

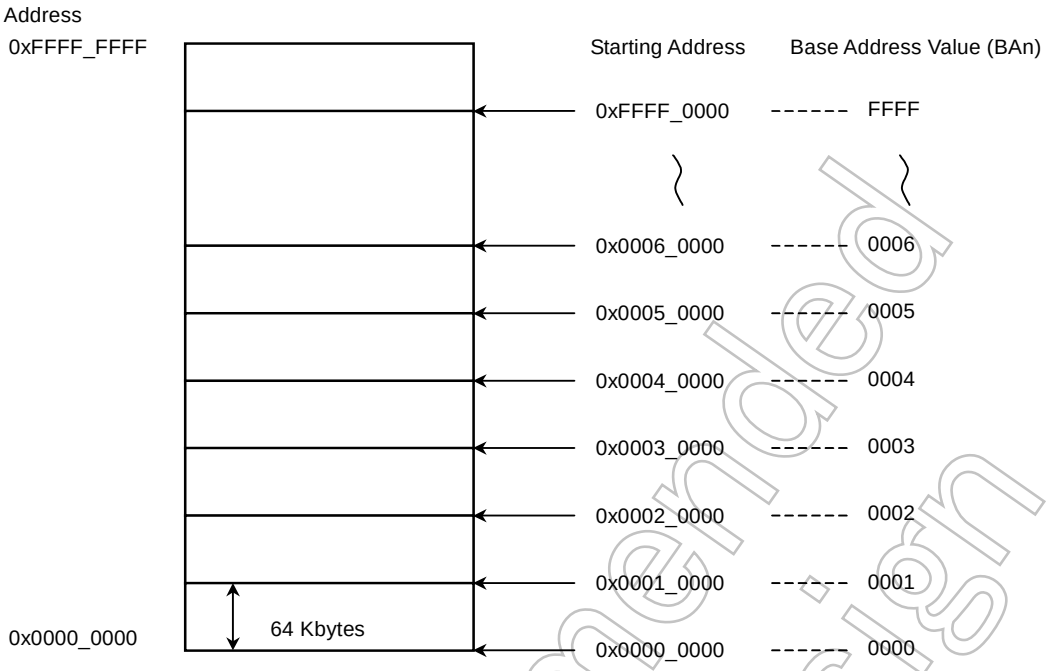
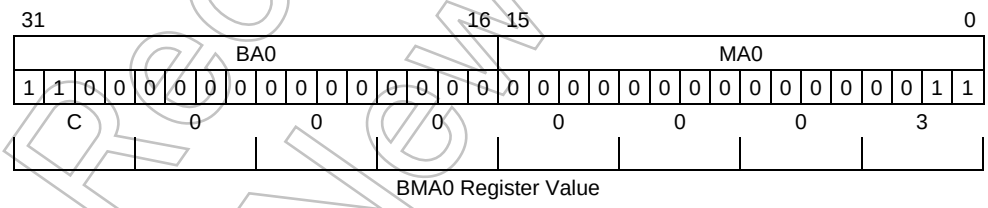


Figure 9.3 Relationships Between Starting Addresses and Base Address Register Values

9.1.2 Base Address and Address Mask Value Calculations

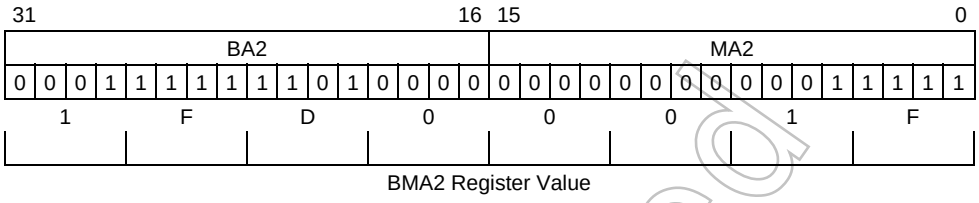
- Program the BMA0 register as follows to cause CS0 to be asserted in the 64 Kbytes of address space starting at 0xC000_0000.



The BA0 field specifies the upper 16 bits of the starting address, or 0xC000. The MA0 field determines whether the A29-A14 bits of the address should be compared or masked. The A31 and A30 bits are always compared. Bits 15-10 of the MA0 field must be cleared so that the A29-A24 bits are always compared.

When the BMA0 register is programmed as shown above, the A31-A16 bits of the address are compared to the value of the BA0 field. Consequently, the 64-Kbyte address range between 0xC000_0000 and 0xC000_FFFF is defined as the $\overline{CS0}$ space.

- Program the BMA2 register as follows to cause CS2 to be asserted in the 1 Mbyte of address space starting at 0x1FD0_0000.



BMA2 Register Value

The BA2 field specifies the upper 16 bits of the starting address, or 0x1FD0. The MA2 field determines whether the A30-A15 bits of the address should be compared or masked. The A31 bit is always compared. Bits 15-5 of the MA2 field must be cleared so that the A30-A20 bits are always compared.

When the BMA2 register is programmed as shown above, the A31-A20 bits of the address are compared to the value of the BA2 field. Consequently, the 1-Mbyte address range between 0x1FD0_0000 and 0x1FDF_FFFF is defined as the CS2 space.

Note: The TMP1962 does not assert any $\overline{CSn}$ signal in the following address ranges:
0x1FC_0000 through 0x1FCF_FFFF
0x4000_0000 through 0x400F_FFFF
0xFFFD_6000 through 0xFFFD_FFFF
0xFFFF_6000 through 0xFFFF_DFFF

Upon reset, the CS0, CS1 and CS3 spaces are disabled while the CS2 space is enabled and spans the entire 4-GB address space.

Table 9.1 shows the programmable block sizes for CS0 to CS3. Even if the user has accidentally programmed more than one chip select line to the same area, only one chip select line is driven because of internal line priorities. CS0 has the highest priority, and CS3 the lowest.

Example: The starting address of the CS0 space is programmed as 0xC000_0000 with a size of 16 Kbytes.
The starting address of the CS1 space is programmed as 0xC000_0000 with a size of 64 Kbytes.

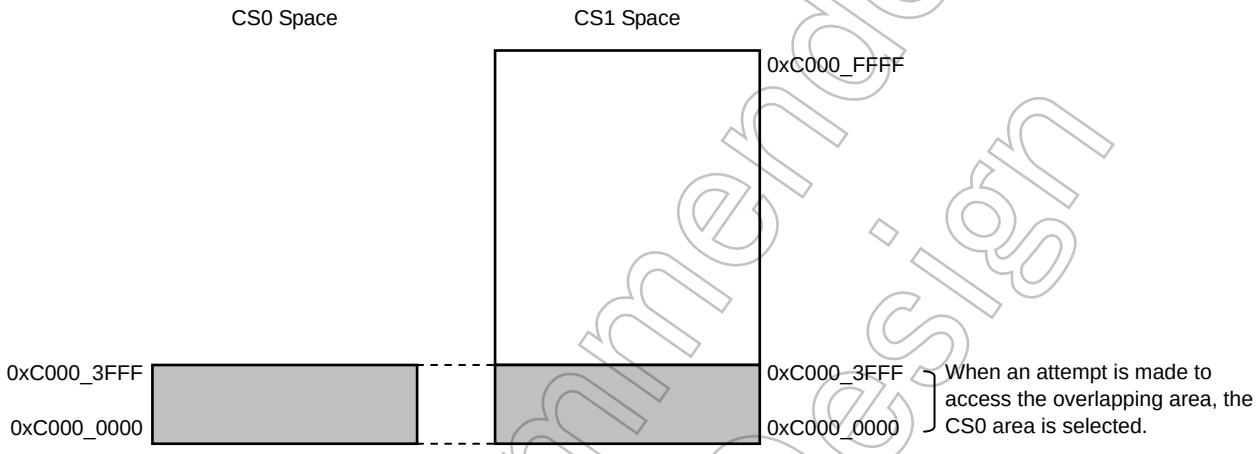


Table 9.1 Supported Block Sizes

Size (bytes) / CS Space	16 K	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M	16 M
CS0	○	○	○	○	○	○	○	○	○	○	○
CS1	○	○	○	○	○	○	○	○	○	○	○
CS2		○	○	○	○	○	○	○	○	○	○
CS3		○	○	○	○	○	○	○	○	○	○

9.2 Chip Select/Wait Control Registers

The organization of the Chip Select/Wait Control registers is shown in Figure 9.4. Each of these registers consist of a master enable bit, a data bus width bit, a wait state field and a dummy cycle field. The B01CS register defines the CS0 and CS1 lines; the B23CS register defines the CS2 and CS3 lines; and the BEXCS register defines the access characteristics for the rest of the address locations.

If the user has accidentally programmed more than one chip select line to the same area, only one chip select line is driven because of internal line priorities (CS0 > CS1 > CS2 > CS3 > EXCS).

B01CS (0xFFFF_E480), B23CS (0xFFFF_E484), BEXCS (0xFFFF_E488)

B01CS (0xFFFFE480)

	31	30	29	28	27	26	25	24
Bit Symbol			B1WCV		B1E		B1RCV	
Read/Write			R/W		R/W		R/W	
Reset Value			0	0	0		0	0
Function			Number of dummy cycles (Write recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited		CS1 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited	

	23	22	21	20	19	18	17	16
Bit Symbol	B1OM			B1BUS	B1W			
Read/Write	R/W		R/W					
Reset Value	0	0		0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Do not use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles (Automatically inserted wait states) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (Externally inserted wait states) 1001: (1+N) WAIT 1011: (3+N) WAIT 1101: (5+N) WAIT 1111: (7+N) WAIT 1000,1010,1100,1110: reserved			

	15	14	13	12	11	10	9	8
Bit Symbol			B0WCV		B0E		B0RCV	
Read/Write			R/W		R/W		R/W	
Reset Value			0	0	0		0	0
Function			Number of dummy cycles (Write recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited		CS0 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited	

	7	6	5	4	3	2	1	0
Bit Symbol	B0OM			B0BUS	B0W			
Read/Write	R/W			R/W				
Reset Value	0	0		0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Do not use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles (Automatically inserted wait states) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (Externally inserted wait states) 1001: (1+N) WAIT 1011: (3+N) WAIT 1101: (5+N) WAIT 1111: (7+N) WAIT 1000,1010,1100,1110: reserved			

Figure 9.4 Chip Select/Wait Control Registers (1/3)

Note:"Please set the number of wait as "+1" when you use = long and BUSRQ the ALE width. "

B23CS (0xFFFF_E484)

	31	30	29	28	27	26	25	24
Bit Symbol			B3WCV		B3E		B3RCV	
Read/Write			R/W		R/W		R/W	
Reset Value			0	0	0		0	0
Function			Number of dummy cycles (Write recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited		CS3 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited	

	23	22	21	20	19	18	17	16
Bit Symbol	B3OM			B3BUS	B3W			
Read/Write	R/W			R/W				
Reset Value	0	0		0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Do not use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles (Automatically inserted wait states) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (Externally inserted wait states) 1001: (1+N) WAIT 1011: (3+N) WAIT 1101: (5+N) WAIT 1111: (7+N) WAIT 1000,1010,1100,1110: reserved			

	15	14	13	12	11	10	9	8
Bit Symbol			B2WCV		B2E	B2M	B2RCV	
Read/Write			R/W		R/W			
Reset Value			0	0	1	0	0	0
Function			Number of dummy cycles (Write recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited		CS2 enable 0: Disable 1: Enable	CS2 space select 0: Whole 4-Gbyte space 1: CS space	Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited	

	7	6	5	4	3	2	1	0
Bit Symbol	B2OM			B2BUS	B2W			
Read/Write	R/W			R/W				
Reset Value	0	0		0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Do not use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles (Automatically inserted wait states) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (Externally inserted wait states) 1001: (1+N) WAIT 1011: (3+N) WAIT 1101: (5+N) WAIT 1111: (7+N) WAIT 1000,1010,1100,1110: reserved			

Figure 9.4 Chip Select/Wait Control Registers (2/3)

Note:"Please set the number of wait as "+1" when you use = long and BUSRQ the ALE width. "

BEXCS (0xFFFF_E488)

	15	14	13	12	11	10	9	8
Bit Symbol			BEXWCV				BEXRCV	
Read/Write			R/W		R/W		R/W	
Reset Value			0	0	0		0	0
Function			Number of dummy cycles (Write recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited				Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Setting prohibited	

	7	6	5	4	3	2	1	0
Bit Symbol	BEXOM			BEXBUS	BEXW			
Read/Write	R/W			R/W				
Reset Value	0	0		0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Do not use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles (Automatically inserted wait states) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (Externally inserted wait states) 1001: (1+N) WAIT 1011: (3+N) WAIT 1101: (5+N) WAIT 1111: (7+N) WAIT 1000,1010,1100,1110: reserved			

Figure 9.4 Chip Select/Wait Control Registers (3/3)

Note:"Please set the number of wait as "+1" when you use = long and BUSRQ the ALE width. "

Both CS1 and CS2 are shared with Port 4 pins. Upon reset, all Port 4 pins are configured as input port pins. To use them as chip select pins, set appropriate bits in the Port 4 Control (P4CR) register and the Port 4 Function (P4FC) register to 1.

10. DMA Controller (DMAC)

The TMP1962 contains an eight-channel DMA controller.

10.1 Features

The TMP1962 DMAC has the following features:

(1) Eight independent DMA channels

(2) Two types of bus requests, with and without bus snooping

(3) Transfer requests:

Internal transfer requests: Software initiated

External transfer requests: Interrupt signals from on-chip I/O peripherals and external interrupt pins, or request signals from $\overline{\text{DREQ}}$ pins

Request signals from $\overline{\text{DREQ}}$: Level-sensitive mode (memory-to-memory)

Edge-triggered mode (memory-to-I/O, I/O-to-memory)

(4) Dual-address mode

(5) Memory-to-memory, memory-to-I/O, and I/O-to-memory transfers

(6) Transfer width:

- Memory: 32-bit (8-bit and 16-bit memory devices are supported through the programming of the CS/Wait Controller.)
- I/O peripherals: 8-, 16-, and 32-bit

(7) Address pointers can increment, decrement or remain constant. The user can program the bit positions at which address increment or decrement occurs.

(8) Fixed channel priority

(9) Selectable endian mode

10.2 Implementation

10.2.1 On-Chip DMAC Interface

Figure 10.1 shows how the DMAC is internally connected with the TX19 core processor and the Interrupt Controller (INTC).

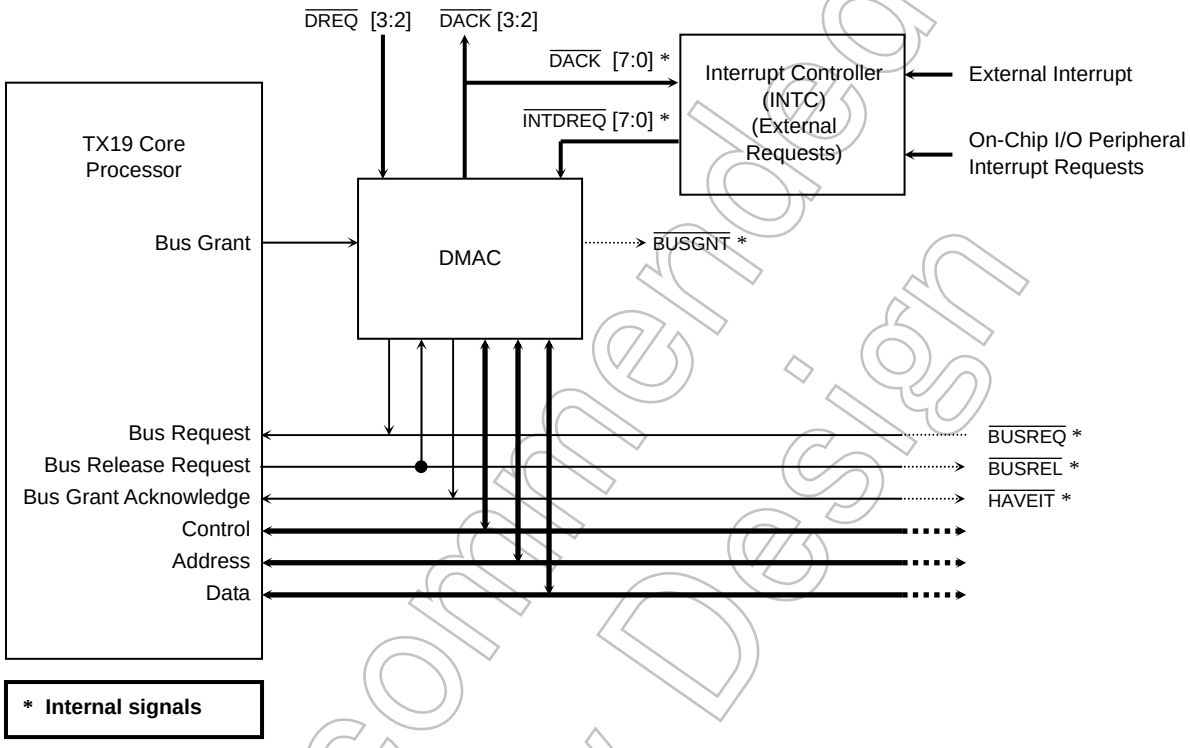


Figure 10.1 DMAC Connections within the TMP1962

The DMAC provides eight independently programmable channels. With each DMA channel, there are two associated signals: a DMA request ($\overline{\text{INTDREQn}}$) and a DMA acknowledge ($\overline{\text{DACKn}}$), where n is a channel number from 0 to 7. $\overline{\text{INTDREQn}}$ is an input to the DMAC coming from the INTC, and DACKn is an output signal from the DMAC going to the INTC. Channels 2 and 3 also accept external DMA requests from the $\overline{\text{DREQ2}}$ and $\overline{\text{DREQ3}}$ pins and send acknowledge signals through the $\overline{\text{DACK2}}$ and $\overline{\text{DACK3}}$ pins. Channel priority is fixed. Channel 0 has the highest priority, and Channel 7 has the lowest priority.

The TX19 core processor supports bus snooping. When snooping is enabled, the TX19 core processor grants the processor data bus to the DMAC, so that the DMAC can access the on-chip RAM and ROM connected to the processor. Snooping can be enabled and disabled under software control. The DMAC bus snooping is discussed in Section 10.2.3 in more detail.

There are two bus request signals from the DMAC going to the TX19 core processor, SREQ and GREQ. GREQ is a bus request without snooping. SREQ is a bus request with snooping. SREQ always takes precedence over GREQ.

10.2.2 DMAC Block

The DMAC block diagram is shown in Figure 10.2.

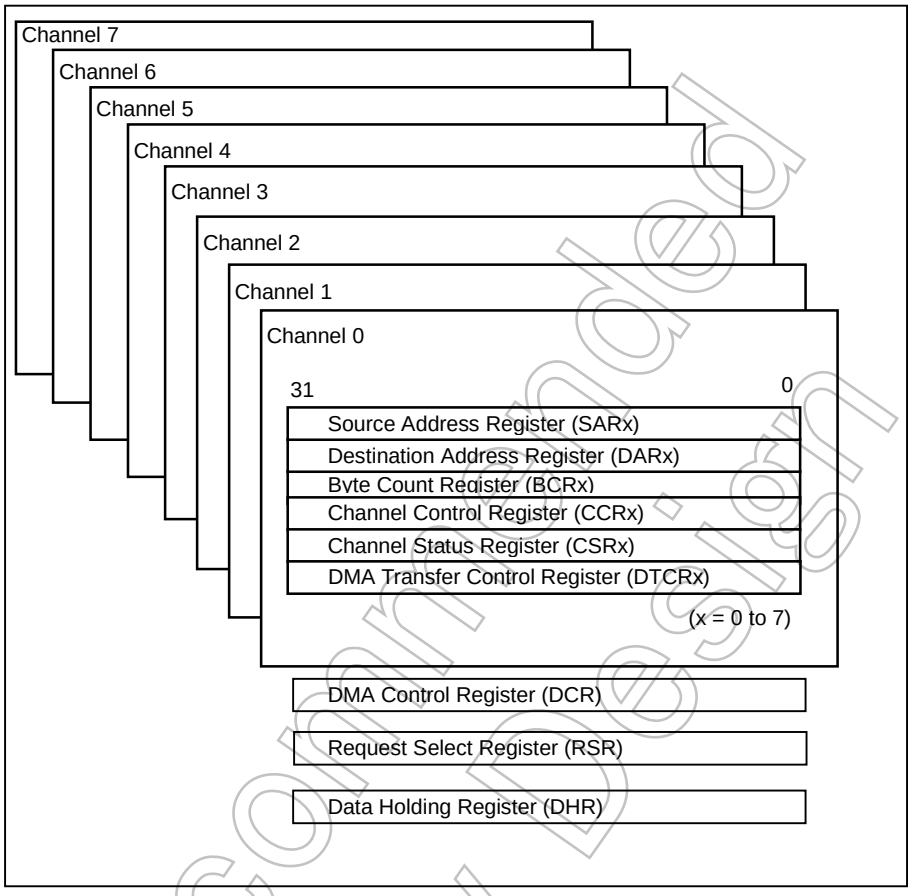


Figure 10.2 DMAC Block Diagram

10.2.3 Bus Snooping

The TX19 core processor supports snoop operations.

If snooping is enabled, the TX19 core processor grants the processor data bus to the DMAC. Because the DMAC takes control of the processor data bus, the TX19 stops operating during snoop operations until the DMAC relinquishes the bus to the processor. Snooping allows the DMAC to access the on-chip RAM and ROM, and thus to use them as a DMA source or destination device.

If snooping is disabled, the DMAC cannot access the on-chip RAM and ROM. However, regardless of whether snooping is enabled or disabled, the DMAC assumes mastership of the TMP1962 on-chip bus (G-Bus) during DMA transfers. Therefore, as long as DMA transfers are in progress, the TX19 core processor cannot access memory or I/O peripherals via the G-Bus; any attempt to do so causes the processor pipeline to stall.

Note: If snooping is disabled, the TX19 core processor does not grant mastership of the processor data bus to the DMAC. Therefore, if the on-chip RAM or ROM is specified as a source or destination for DMA transfers, a DMA acknowledge signal will never be returned, causing bus lockup.

10.3 Register Description

The DMAC has fifty-one 32-bit registers. The DMAC register map is shown in Table 10.1.

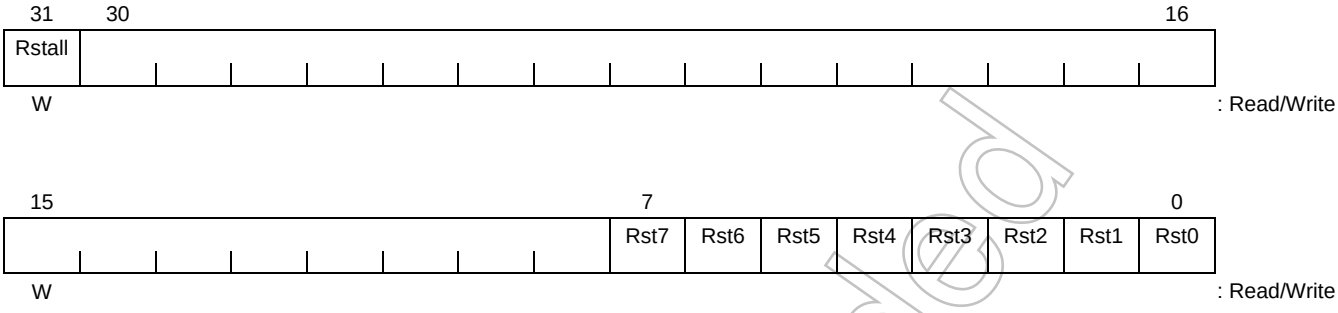
Table 10.1 DMAC Registers (1/2)

Address	Symbol	Register Name
0xFFFF_E200	CCR0	Channel Control Register (Ch. 0)
0xFFFF_E204	CSR0	Channel Status Register (Ch. 0)
0xFFFF_E208	SAR0	Source Address Register (Ch. 0)
0xFFFF_E20C	DAR0	Destination Address Register (Ch. 0)
0xFFFF_E210	BCR0	Byte Count Register (Ch. 0)
0xFFFF_E218	DTCR0	DMA Transfer Control Register (Ch. 0)
0xFFFF_E220	CCR1	Channel Control Register (Ch. 1)
0xFFFF_E224	CSR1	Channel Status Register (Ch. 1)
0xFFFF_E228	SAR1	Source Address Register (Ch. 1)
0xFFFF_E22C	DAR1	Destination Address Register (Ch. 1)
0xFFFF_E230	BCR1	Byte Count Register (Ch. 1)
0xFFFF_E238	DTCR1	DMA Transfer Control Register (Ch. 1)
0xFFFF_E240	CCR2	Channel Control Register (Ch. 2)
0xFFFF_E244	CSR2	Channel Status Register (Ch. 2)
0xFFFF_E248	SAR2	Source Address Register (Ch. 2)
0xFFFF_E24C	DAR2	Destination Address Register (Ch. 2)
0xFFFF_E250	BCR2	Byte Count Register (Ch. 2)
0xFFFF_E258	DTCR2	DMA Transfer Control Register (Ch. 2)
0xFFFF_E260	CCR3	Channel Control Register (Ch. 3)
0xFFFF_E264	CSR3	Channel Status Register (Ch. 3)
0xFFFF_E268	SAR3	Source Address Register (Ch. 3)
0xFFFF_E26C	DAR3	Destination Address Register (Ch. 3)
0xFFFF_E270	BCR3	Byte Count Register (Ch. 3)
0xFFFF_E278	DTCR3	DMA Transfer Control Register (Ch. 3)
0xFFFF_E280	CCR4	Channel Control Register (Ch. 4)
0xFFFF_E284	CSR4	Channel Status Register (Ch. 4)
0xFFFF_E288	SAR4	Source Address Register (Ch. 4)
0xFFFF_E28C	DAR4	Destination Address Register (Ch. 4)
0xFFFF_E290	BCR4	Byte Count Register (Ch. 4)
0xFFFF_E298	DTCR4	DMA Transfer Control Register (Ch. 4)
0xFFFF_E2A0	CCR5	Channel Control Register (Ch. 5)
0xFFFF_E2A4	CSR5	Channel Status Register (Ch. 5)
0xFFFF_E2A8	SAR5	Source Address Register (Ch. 5)
0xFFFF_E2AC	DAR5	Destination Address Register (Ch. 5)
0xFFFF_E2B0	BCR5	Byte Count Register (Ch. 5)
0xFFFF_E2B8	DTCR5	DMA Transfer Control Register (Ch. 5)
0xFFFF_E2C0	CCR6	Channel Control Register (Ch. 6)
0xFFFF_E2C4	CSR6	Channel Status Register (Ch. 6)
0xFFFF_E2C8	SAR6	Source Address Register (Ch. 6)
0xFFFF_E2CC	DAR6	Destination Address Register (Ch. 6)
0xFFFF_E2D0	BCR6	Byte Count Register (Ch. 6)
0xFFFF_E2D8	DTCR6	DMA Transfer Control Register (Ch. 6)

Figure 10.2 DMAC Registers (2/2)

0xFFFF_E2E0	CCR7	Channel Control Register (Ch. 7)
0xFFFF_E2E4	CSR7	Channel Status Register (Ch. 7)
0xFFFF_E2E8	SAR7	Source Address Register (Ch. 7)
0xFFFF_E2EC	DAR7	Destination Address Register (Ch. 7)
0xFFFF_E2F0	BCR7	Byte Count Register (Ch. 7)
0xFFFF_E2F8	DTCR7	DMA Transfer Control Register (Ch. 7)
0xFFFF_E300	DCR	DMA Control Register (DMAC)
0xFFFF_E304	RSR	Request Select Register (DMAC)
0xFFFF_E30C	DHR	Data Holding Register (DMAC)

10.3.1 DMA Control Register (DCR)



Bits	Mnemonic	Field Name	Description
31	Rstall	Reset All	Performs a software reset of the DMAC. When the Rstall bit is set to 1, all the DMAC internal registers are initialized to their reset values. Any transfer requests are removed and all the eight DMA channels are put in Idle state. 0: Don't care 1: Resets the DMAC.
7	Rst7	Reset 7	Performs a software reset of DMAC Channel 7. When the Rst7 bit is set to 1, all the DMAC Channel 7 internal registers and the RSR Channel 7 bit are initialized to their reset values. Any transfer requests for Channel 7 are removed and Channel 7 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 7.
6	Rst6	Reset 6	Performs a software reset of DMAC Channel 6. When the Rst6 bit is set to 1, all the DMAC Channel 6 internal registers and the RSR Channel 6 bit are initialized to their reset values. Any transfer requests for Channel 6 are removed and Channel 6 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 6.
5	Rst5	Reset 5	Performs a software reset of DMAC Channel 5. When the Rst5 bit is set to 1, all the DMAC Channel 5 internal registers and the RSR Channel 5 bit are initialized to their reset values. Any transfer requests for Channel 5 are removed and Channel 5 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 5.
4	Rst4	Reset 4	Performs a software reset of DMAC Channel 4. When the Rst4 bit is set to 1, all the DMAC Channel 4 internal registers and the RSR Channel 4 bit are initialized to their reset values. Any transfer requests for Channel 4 are removed and Channel 4 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 4.
3	Rst3	Reset 3	Performs a software reset of DMAC Channel 3. When the Rst3 bit is set to 1, all the DMAC Channel 3 internal registers and the RSR Channel 3 bit are initialized to their reset values. Any transfer requests for Channel 3 are removed and Channel 3 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 3.

Bits	Mnemonic	Field Name	Description
2	Rst2	Reset 2	Performs a software reset of DMAC Channel 2. When the Rst2 bit is set to 1, all the DMAC Channel 2 internal registers and the RSR Channel 2 bit are initialized to their reset values. Any transfer requests for Channel 2 are removed and Channel 2 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 2.
1	Rst1	Reset 1	Performs a software reset of DMAC Channel 1. When the Rst1 bit is set to 1, all the DMAC Channel 1 internal registers and the RSR Channel 1 bit are initialized to their reset values. Any transfer requests for Channel 1 are removed and Channel 1 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 1.
0	Rst0	Reset 0	Performs a software reset of DMAC Channel 0. When the Rst0 bit is set to 1, all the DMAC Channel 0 internal registers and the RSR Channel 0 bit are initialized to their reset values. Any transfer requests for Channel 0 are removed and Channel 0 is put in Idle state. 0: Don't care 1: Resets DMAC Channel 0.

Figure 10.3 DMA Control Register (DCR)

Note 1: If the software reset command is written to the DCR register immediately after the completion of the last transfer cycle of a DMA transaction, the DMA-done interrupt will not be cleared. In this case, the software reset only initializes channel registers and other settings.

Note 2: Do not issue a software reset command to the DCR register via a DMA transfer.

10.3.2 Channel Control Registers (CCRn)

31	30					25	24	23	22	21	20	19	18	17	16	
Str	0						—	NIEn	AbIEn	—	—	—	—	Big	—	
W							W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: Read/Write
								1	1	1	0	0	0	1	0	: Reset Value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
—	ExR	PosE	Lev	SReq	RelEn	SIO	SAC		DIO	DAC		TrSiz		DPS		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W		R/W		R/W		: Read/Write
0	0	0	0	0	0	0	00		0	00		00		00		: Reset Value

Bits	Mnemonic	Field Name	Description
31	Str	Channel Start	Start (Reset value: -) Enables a DMA channel. Setting this bit puts the DMA channel in Ready state. DMA transfer starts as soon as a transfer request is received. Only a write of 1 is valid, and a write of 0 has no effect on this bit. A 0 is returned on read. 1: Enables a DMA channel.
24	—	(Reserved)	This bit is reserved and must be written as 0.
23	NIEn	Normal Completion Interrupt Enable	Normal Completion Interrupt Enable (Reset value: 1) 1: Enables an interrupt when the channel finishes a transfer without an error condition. 0: Does not enable an interrupt when the channel finishes a transfer without an error condition.
22	AbIEn	Abnormal Termination Interrupt Enable	Abnormal Completion Interrupt Enable (Reset value: 1) 1: Enables an interrupt when the channel encounters a transfer error. 0: Does not enable an interrupt when the channel encounters a transfer error.
21	—	(Reserved)	This bit is reserved and reset to 1, but it must be written as 0.
20	—	(Reserved)	This bit is reserved and must be written as 0.
19	—	(Reserved)	This bit is reserved and must be written as 0.
18	—	(Reserved)	This bit is reserved and must be written as 0.
17	Big	Big-Endian	Big Endian (Reset value: 1) 1: The DMA channel operates in big-endian mode. 0: The DMA channel operates in little-endian mode.
16	—	(Reserved)	This bit is reserved and must be written as 0.
15	—	(Reserved)	This bit is reserved and must be written as 0.
14	ExR	External Request Mode	External Request Mode (Reset value: 0) Selects a transfer request mode. 1: External transfer requests (interrupt-driven or DREQ-driven) 0: Internal transfer requests (software-initiated)
13	PosE	Positive Edge	Positive Edge (Reset value: 0) Defines the polarity of the internal DMA request signal (INTDREQn or DREQn) for the channel. This bit is valid for external transfer requests (i.e., when ExR = 1), and has no effect on internal transfer requests (i.e., when ExR = 0). The PosE bit must always be cleared because INTDREQn and DREQn are low-active signals. 1: Setting prohibited 0: INTDREQn and DREQn are configured as falling-edge triggered or low-level sensitive. DACKn is low active.

Figure 10.4 Channel Control Registers (CCRn) (1/3)

Bits	Mnemonic	Field Name	Description
12	Lev	Level Mode	Level Mode (Reset value: 0) Specifies whether external transfer requests are level-sensitive or edge-triggered. This bit is valid for external transfer requests (i.e., when ExR = 1), and has no effect on internal transfer requests (i.e., when ExR = 0). The Lev bit must always be set to 1 because INTDREQn is low-active signals. The Lev bit determines how DREQn is recognized, as follows: 1: Level mode. A specified level (if PosE = 0, low level) of DREQn is recognized as a data transfer request. 0: Edge mode. A specified transition (if PosE = 0, falling edge) of DREQn is recognized as a data transfer request.
11	SReq	Snoop Request	Snoop Request (Reset value: 0) Controls whether or not to request bus mastership with snooping. If set, the TX19 core processor's snoop function becomes valid, allowing the DMAC to use the processor's data bus. If cleared, the snoop function is disabled. 1: The snoop function is enabled (i.e., SREQ is used as a bus request signal). 0: The snoop function is disabled (i.e., GREQ is used as a bus request signal).
10	RelEn	Bus Release Request Enable	Release Request Enable (Reset value: 0) Controls whether or not to respond to the bus release request signal from the TX19 core processor. This bit is valid when the DMAC uses GREQ as a bus request signal. This bit has no meaning or effect when the DMAC uses SREQ as a bus request signal because, in that case, the TX19 core processor does not have the capability to generate a bus release request signal. 1: The DMAC will respond to the bus release request signal from the TX19 core processor, if it has control of the bus. The DMAC will relinquish the bus when the current DMA bus cycle completes. 0: The DMAC will ignore the bus release request signal from the TX19 core processor.
9	SIO	I/O Source	Source Type: I/O (Reset value: 0) Specifies the type of the source device. 1: I/O device 0: Memory
8 : 7	SAC	Source Address Count	Source Address Count (Reset value: 00) Selects the manner in which the source address changes after each cycle. 1x: Fixed (remains unchanged) 01: Decremented 00: Incremented
6	DIO	I/O Destination	Destination Type: I/O (Reset value: 0) Specifies the type of the destination device. 1: I/O device 0: Memory
5 : 4	DAC	Destination Address Count	Destination Address Count (Reset value: 00) Selects the manner in which the destination address changes after each cycle. 1x: Fixed (remains unchanged) 01: Decremented 00: Incremented

Figure 10.4 Channel Control Registers (CCRn) (2/3)

Bits	Mnemonic	Field Name	Description
3 : 2	TrSiz	Transfer Size	Transfer Size (Reset value: 00) Specifies the amount of data to be transferred in response to a DMA request. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)
1 : 0	DPS	Device Port Size	Device Port Size (Reset value: 00) Specifies the port size of a source or destination I/O device. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)

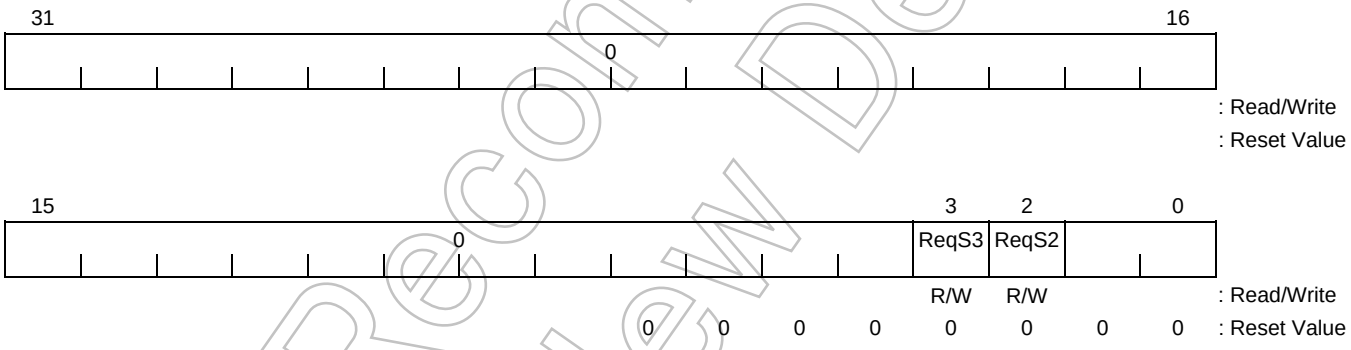
Figure 10.4 Channel Control Registers (CCRn) (3/3)

Note 1: The CCRn register must be programmed before placing the DMAC in Ready state.

Note 2: To access on-chip peripherals or to perform a DMA transfer in response to a request issued through a DREQ pin, the transfer size (TrSiz) must be equal to the device port size (DPS).

Note 3: The DPS field has no meaning or effect on memory-to-memory transfers.

10.3.3 Request Select Register (RSR)

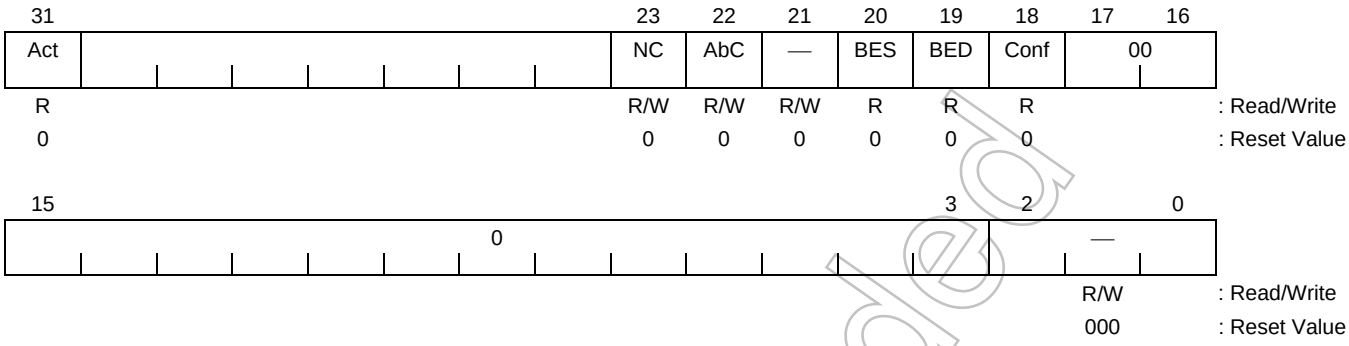


Bits	Mnemonic	Field Name	Description
3	ReqS3	Request Select (Ch. 3)	Request Select (Reset value: 0) Selects the type of an external transfer request enabled for DMA Channel 3. 1: Request from the DREQ3 pin. 0: Request from the Interrupt Controller (INTC).
2	ReqS2	Request Select (Ch. 2)	Request Select (Reset value: 0) Selects the type of an external transfer request enabled for DMA Channel 2. 1: Request from the DREQ2 pin. 0: Request from the Interrupt Controller (INTC).

Note: Bits 0, 1, and 4 to 7 of the RSR must be set to 0.

Figure 10.5 Request Select Register (RSR)

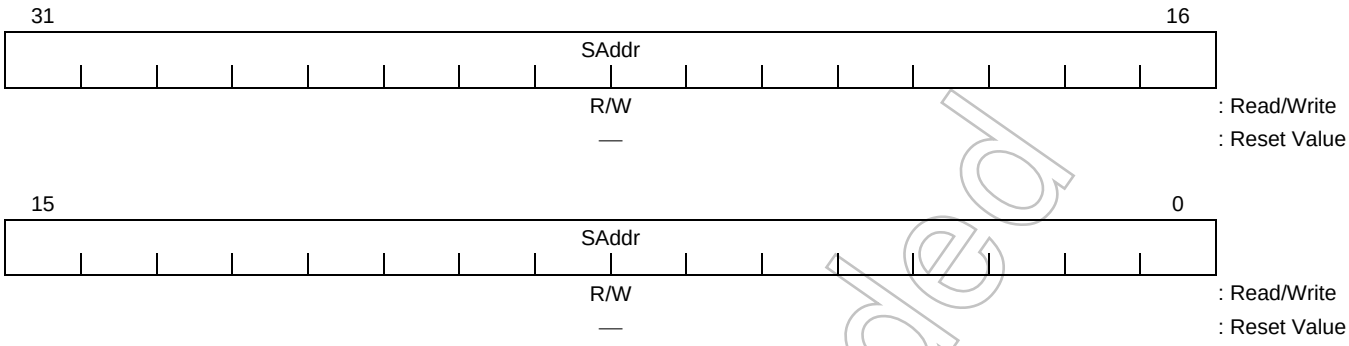
10.3.4 Channel Status Registers (CSRn)



Bits	Mnemonic	Field Name	Description
31	Act	Channel Active	Channel Active (Reset value: 0) Indicates whether or not the DMA channel is in Ready state. 1: The DMA channel is in Ready state. 0: The DMA channel is not in Ready state.
23	NC	Normal Completion	Normal Completion (Reset value: 0) If set, the DMA channel has terminated by normal completion. If the NIEn bit in the CCRn is set, an interrupt is generated. The NC bit is cleared by writing a 0 to it. Clearing the NC bit causes the interrupt to be cleared. The NC bit must be cleared prior to starting the next transfer. An attempt to set the Str bit in the CCRn when NC = 1 will cause an error. A write of 1 has no effect on this bit. 1: The DMA channel has terminated by normal completion. 0: The DMA channel has not terminated by normal completion.
22	AbC	Abnormal Completion	Abnormal Completion (Reset value: 0) If set, the DMA channel has terminated with an error. If the AbIEn bit in the CCRn is set, an interrupt is generated. The AbC bit is cleared by writing a 0 to it. Clearing the AbC bit causes the interrupt to be cleared and the BES, BED and Conf bits to be also cleared. The AbC bit must be cleared prior to starting the next transfer. An attempt to set the Str bit in the CCRn when AbC = 1 will cause an error. A write of 1 has no effect on this bit. 1: The DMA channel has terminated with an error. 0: The DMA channel has not terminated with an error.
21	—	(Reserved)	This bit is reserved and must be written as 0.
20	BES	Source Bus Error	Source Bus Error (Reset value: 0) 1: A bus error has occurred during the source read cycle. 0: A bus error has not occurred during the source read cycle.
19	BED	Destination Bus Error	Destination Bus Error (Reset value: 0) 1: A bus error has occurred during the destination write cycle. 0: A bus error has not occurred during the destination write cycle.
18	Conf	Configuration Error	Configuration Error (Reset value: 0) 1: A configuration error is present. 0: No configuration error is present.
2 : 0	—	(Reserved)	These bits are reserved and must be written as 0s.

Figure 10.6 Channel Status Registers (CSRn)

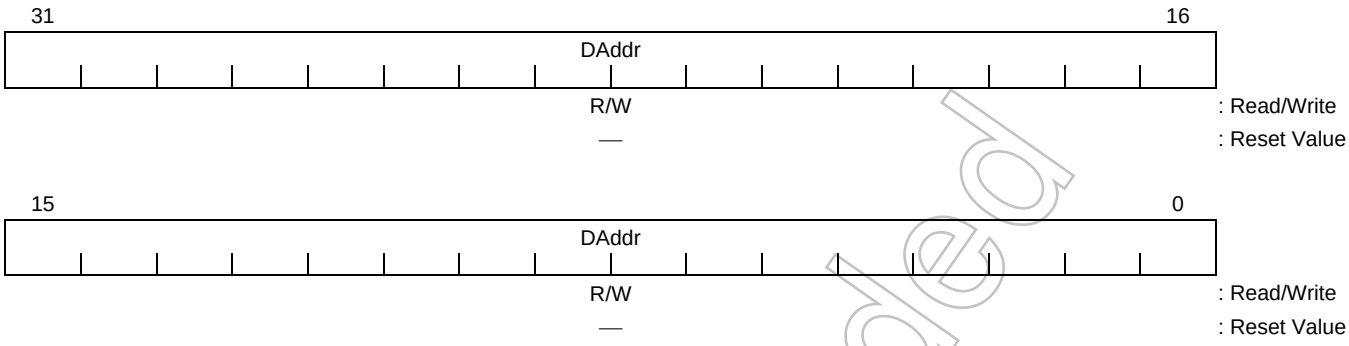
10.3.5 Source Address Registers (SARn)



Bits	Mnemonic	Field Name	Description
31 : 0	SAddr	Source Address	Source Address (Reset value: -) Contains the physical address of the source device. The address changes as programmed in the SAC and TrSiz fields in the CCRn and the SACM field in the DTCRn.

Figure 10.7 Source Address Registers (SARn)

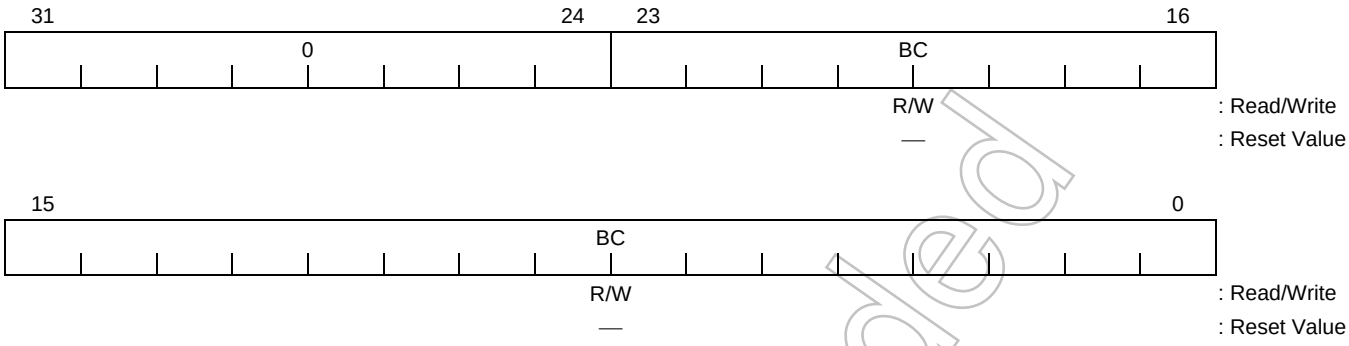
10.3.6 Destination Address Registers (DARn)



Bits	Mnemonic	Field Name	Description
31 : 0	DAddr	Destination Address	Destination Address (Reset value: -) Contains the physical address of the destination device. The address changes as programmed in the DAC and TrSiz fields in the CCRn and the DACM field in the DTCRn.

Figure 10.8 Destination Address Registers (DARn)

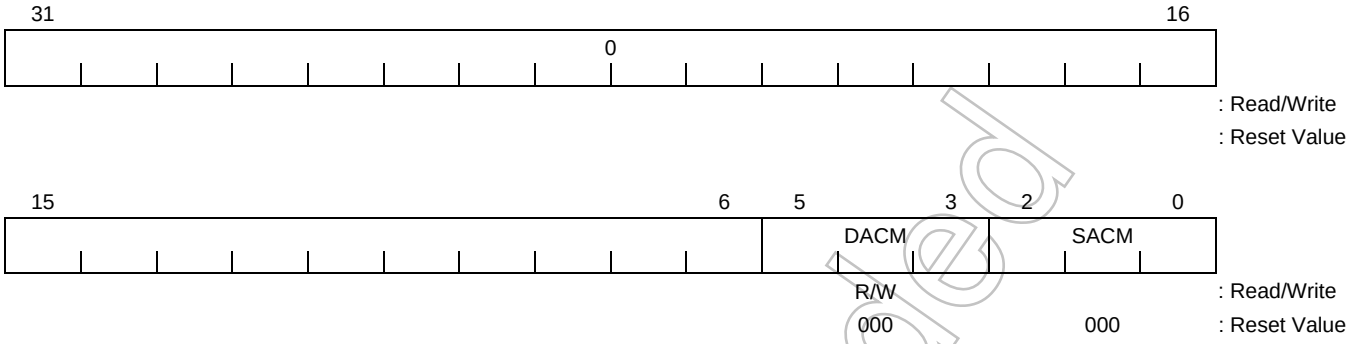
10.3.7 Byte Count Registers (BCRn)



Bits	Mnemonic	Field Name	Description
23 : 0	BC	Byte Count	Byte Count (Reset value: -) Contains the number of bytes left to transfer on a DMA channel. The count is decremented by 1, 2 or 4 (as determined by the TrSiz field in the CCRn register) for each successful transfer.

Figure 10.9 Byte Count Registers (BCRn)

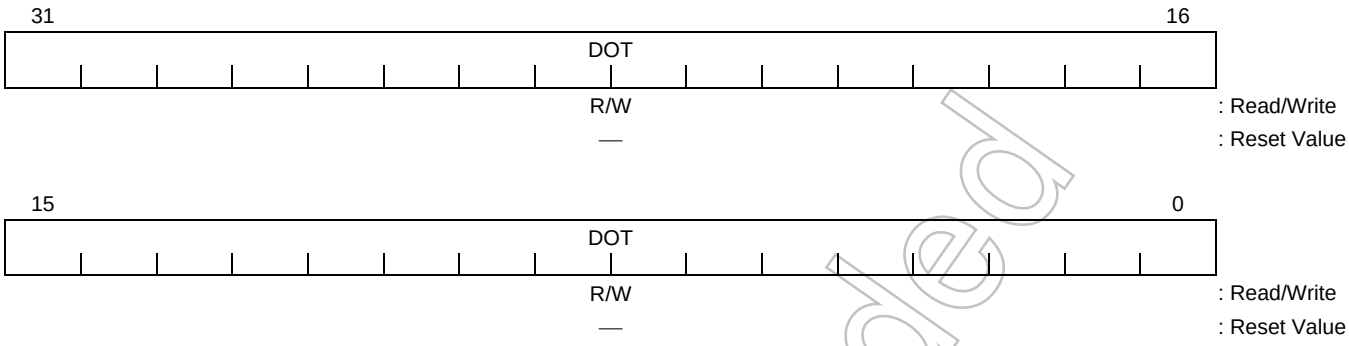
10.3.8 DMA Transfer Control Registers (DTCRn)



Bits	Mnemonic	Field Name	Description
5 : 3	DACM	Destination Address Count Mode	Destination Address Count Mode Selects the manner in which the destination address is incremented or decremented. 000: Counting begins with bit 0 of the DARn. 001: Counting begins with bit 4 of the DARn. 010: Counting begins with bit 8 of the DARn. 011: Counting begins with bit 12 of the DARn. 100: Counting begins with bit 16 of the DARn. 101: Reserved 110: Reserved 111: Reserved
2 : 0	SACM	Source Address Count Mode	Source Address Count Mode Selects the manner in which the source address is incremented or decremented. 000: Counting begins with bit 0 of the SARn. 001: Counting begins with bit 4 of the SARn. 010: Counting begins with bit 8 of the SARn. 011: Counting begins with bit 12 of the SARn. 100: Counting begins with bit 16 of the SARn. 101: Reserved 110: Reserved 111: Reserved

Figure 10.10 DMA Transfer Control Registers (DTCRn)

10.3.9 Data Holding Register (DHR)



Bits	Mnemonic	Field Name	Description
31 : 0	DOT	Data on Transfer	Data on Transfer (Reset value: -) Contains data read from the source address during a dual-address operation.

Figure 10.11 Data Holding Register (DHR)

10.4 Operation

This section describes the operation of the DMAC.

10.4.1 Overview

The DMAC is a high-speed 32-bit DMA controller used to move quickly large blocks of data between I/O peripherals and memory without intervention of the TX19 core processor.

(1) Devices supported for the source and destination

The DMAC handles data transfers from memory to memory, and between memory and I/O peripherals. The device from which data is transferred is referred to as a source device, and the device to which data is transferred is referred to as a destination device. Both memory and I/O peripherals can be a source or destination device. The DMAC supports data transfers from memory to I/O peripherals, from I/O peripherals to memory, and from memory to memory, but not from I/O peripherals to I/O peripherals.

DMA protocols for memory and I/O peripherals differ in that when accessing an I/O peripheral, the DMAC asserts the $\overline{\text{DACKn}}$ (n = channel number) signal to indicate that data is being transferred in response to a previous transfer request. Because each DMA channel has only one $\overline{\text{DACKn}}$ signal, the DMAC cannot handle data transfers between two I/O peripherals.

Interrupt requests can be programmed to be a trigger to initiate a DMA process instead of requesting an interrupt to the TX19 core processor. If so programmed, the Interrupt Controller (INTC) forwards a DMA request to the DMAC (see "Interrupts"). The DMA request coming from the INTC is cleared when the INTC receives a $\overline{\text{DACKn}}$ from the DMAC. Consequently, a DMA request for a transfer to/from an I/O peripheral is cleared after each DMA bus cycle (i.e., every time the number of bytes programmed into the CCRn.TrSiz field is transferred). On the other hand, during memory-to-memory transfer, the $\overline{\text{DACKn}}$ signal is not asserted until the byte count register (BCRn) reaches zero. Therefore, memory-to-memory transfer can continuously move large blocks of data in response to a single DMA request.

For example, data transfers between the TMP1962 on-chip peripheral and on- or off-chip memory is discontinued after every DMA bus cycle. Nonetheless, until the BCRn register reaches zero, the DMAC remains in Ready state to wait for the next transfer request.

(2) Exchanging bus mastership (bus arbitration)

In response to a DMA request, the DMAC issues a bus request to the TX19 core processor. When the DMAC receives a bus grant signal from the TX19 core processor, it assumes bus mastership to service the DMA request.

There are two bus request signals from the DMAC going to the TX19 core processor. One is a bus request without snooping (GREQ), and the other is a bus request with snooping (SREQ). The SReq bit in the CCRn register is used to select a bus request signal to use for each DMA channel.

While the DMAC has control of the bus, the TX19 core processor may issue a bus release request to the DMAC. The RelEn bit of the CCRn register controls whether to honor this request on a channel-by-channel basis. This setting has a meaning only when a DMA channel uses GREQ (i.e., a bus request without snooping). It has no meaning or effect when a DMA channel uses SREQ (i.e., a bus request with snooping) because, in this case, the TX19 core processor does not have the capability to generate a bus release request.

The DMAC relinquishes the bus to the TX19 core processor when there is no pending DMA request to be serviced.

Note 1: The $\overline{NMI}$ interrupt is left pending while the DMAC has control of the bus.
Note 2: Do not place the TMP1962 in Halt power-down mode while the DMAC is operating.

(3) Transfer request generation

Each DMA channel supports two types of request generation method: internal and external. Internal requests are those generated within the DMAC. The DMA channel is started as soon as the Str bit in the CCRn register is set. The channel immediately requests the bus and begins transferring data.

If a channel is programmed for external request and the Str bit is set, the $\overline{INTDREQn}$ signal asserted by the Interrupt Controller or the $\overline{DREQn}$ signal asserted by the external device causes the channel to request the bus and begin a transfer. The DMAC can be programmed to recognize a transfer request either with the low level of the $\overline{INTDREQn}$ signal or with the falling edge or low level of the $\overline{DREQn}$ signal.

(4) Data transfer modes

The TMP1962 DMAC supports dual-address transfers, but not single-address transfers.

The dual-address mode allows data to be transferred from memory to memory and between memory and an I/O peripheral. In this mode, the DMAC explicitly addresses both the source and destination devices. The DMAC also generates a $\overline{DACKn}$ signal when accessing an I/O peripheral. In dual-address mode, a transfer takes place in two DMA bus cycles: a source read cycle and a destination write cycle. In the source read cycle, the data being transferred is read from the source address and put into the DMAC internal Data Holding Register (DHR). In the destination write cycle, the DMAC writes data in the DHR to a destination address.

(5) DMA channel operation

The DMAC has eight independent DMA channels, 0 to 7. Setting the Start (Str) bit in the CCRn (n = 0-7) enables a particular channel and puts it in Ready state.

When a DMA request is detected in any of the channels in Ready state, the DMAC arbitrates for the bus and begins a transfer. When no DMA request is pending, the DMAC relinquishes the bus to the TX19 core processor and returns to Ready state. The channel can terminate by normal completion or from an error of a bus cycle. When a channel terminates, that channel is put in Idle state. Interrupts can be generated by error termination or by normal channel termination.

Figure 10.12 shows general state transitions of a DMA channel.

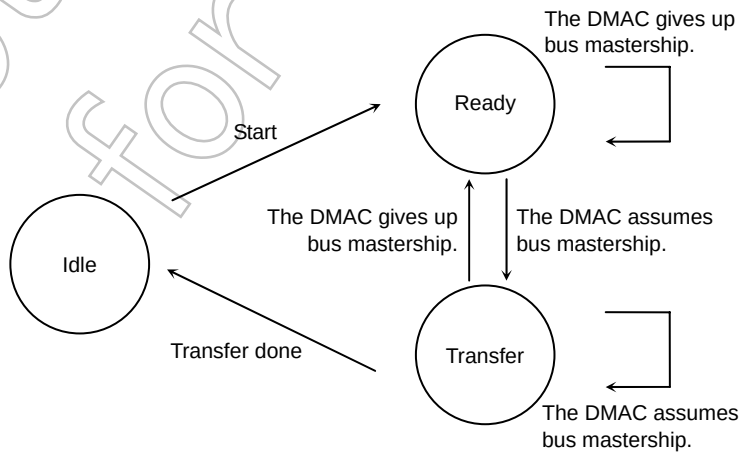


Figure 10.12 DMA Channel State Transitions

(6) Summary of transfer modes

The DMAC can perform data transfers as follows according to the combination of mode settings.

Transfer Request	Edge/Level	Address Mode	Data Flow
Internal	—	Dual	Memory-to-memory
External	Low Level ($\overline{\text{INTDREQn}}$)		Memory-to-memory
			Memory-to-I/O
			I/O-to-memory
External	Low Level ($\overline{\text{DREQn}}$)		Memory-to-memory
	Falling Edge ($\overline{\text{DREQn}}$)		Memory-to-I/O
			I/O-to-memory

(7) Address change options

Address pointers can increment, decrement or remain constant. The SAC and DAC fields in the CCRn respectively select address change directions for the Source Address Register (SARn) and the Destination Address Register (DARn). While memory addresses can be programmed to increment, decrement or remain constant, I/O addresses must be programmed to remain constant. When an I/O peripheral is selected as the source or destination device, the SAC or DAC field in the CCRn must be set to 1x (address fixed).

The SACM and DACM fields in the DTCRn provide options to program bit positions at which the source and destination addresses are incremented or decremented after each transfer. The bit position can be bit 0, 4, 8, 12 or 16. Use of bit 0 is the regular increment/decrement mode in which the address changes by 1, 2 or 4, according to the source or destination size. Two examples of how other increment/decrement modes affect address changes are shown below.

Example 1: When address bit 0 is selected in the SACM field and address bit 4 is selected in the DACM field

SAC: Programmed to increment the source address
DAC: Programmed to increment the destination address
TrSiz: Programmed to a transfer size of 32 bits
Source address: 0xA000_1000
Destination address: 0xB000_0000
SACM: 000 → Bit 0 is the source address bit at which address increment occurs.
DACM: 001 → Bit 4 is the destination address bit at which address increment occurs.

	Source	Destination
1st transfer	0xA000_1000	0xB000_0000
2nd transfer	0xA000_1001	0xB000_0010
3rd transfer	0xA000_1002	0xB000_0020
4th transfer	0xA000_1003	0xB000_0030
	…	…

Example 2: When address bit 8 is selected in the SACM field and address bit 0 is selected in the DACM field

SAC:	Programmed to decrement the address
DAC:	Programmed to decrement the address
TrSiz:	Programmed to a transfer size of 16 bits
Source address:	0xA000_1000
Destination address:	0xB000_0000
SACM:	010 → Bit 8 is the source address bit at which address decrement occurs.
DACM:	000 → Bit 0 is the destination address bit at which address decrement occurs.

	Source	Destination
1st transfer	0xA000_1000	0xB000_0000
2nd transfer	0x9FFF_FF00	0xAFFF_FFFE
3rd transfer	0x9FFF_FE00	0xAFFF_FFFC
4th transfer	0x9FFF_FD00	0xAFFF_FFFA
	…	…

10.4.2 Transfer Request Generation

A DMA request must be issued for the DMAC to initiate a data transfer. Each DMA channel in the DMAC supports two types of request generation method: internal and external. In either request generation mode, once a DMA channel is started, a DMA request causes the DMAC to arbitrate for the bus and begin transferring data.

- Internal request generation

  A channel is programmed for internal request by clearing the ExR bit in the CCRn. In internal request generation mode, a transfer request is generated as soon as the Str bit in the CCRn is set.

  An internally generated request keeps a transfer request pending until the transfer is complete. If no transition to a higher-priority DMA channel or a bus master occurs, the channel will use 100% of the available bus bandwidth to transfer all data continuously.

  Internally generated requests support only memory-to-memory transfer.

- External request generation

  A channel is programmed for external request by setting the ExR bit in the CCRn. In external request generation mode, setting the Str bit in the CCRn puts the channel in Ready state. While in Ready state, assertion of the $\overline{\text{INTDREQn}}$ signal (where n is the channel number) coming from the Interrupt Controller (INTC), or the $\overline{\text{DREQn}}$ signal coming from an external device, causes a transfer request to be generated. Externally generated requests support data transfers from memory to memory and between memory and an I/O peripheral.

  The TMP1962 can recognize a transfer request with the low level of $\overline{\text{INTDREQn}}$ or the falling edge or low level of $\overline{\text{DREQn}}$.

  The transfer size, i.e., the amount of data to be transferred in response to a transfer request, is programmed in the TrSiz field in the CCRn. The transfer size can be 32 bits, 16 bits or 8 bits.

  Details of transfer request generation by $\overline{\text{INTDREQn}}$ and $\overline{\text{DREQn}}$ are described below.

(1) Transfer request coming from the INTC

A transfer request is removed by assertion of the $\overline{\text{DACKn}}$ signal (where n is the channel number). $\overline{\text{DACKn}}$ is asserted: 1) when an I/O peripheral bus cycle has completed, and 2) when the Byte Count Register (BCRn) has reached zero in memory-to-memory transfer. Consequently, a memory-to-I/O or I/O-to-memory transfer request terminates after one DMA bus cycle completes, whereas memory-to-memory transfer can continuously move large blocks of data in response to a single DMA request.

The INTC might clear $\overline{\text{INTDREQn}}$ before the DMAC accepts it and begins a data transfer. It must be noted that, even if that happens, a DMA bus cycle might be executed after the interrupt request has been cleared.

(2) Transfer request coming from an external device

In Edge mode, each transfer request requires the deassertion and assertion of the $\overline{\text{DREQn}}$ signal to produce an effective edge. In Level mode, a continuous transfer request can be made by holding an effective level. Memory-to-memory transfer supports Low Level-Sensitive mode only. I/O-to-memory transfer supports Falling Edge-Sensitive mode only.

- Level mode

  In Level mode, the DMAC samples the $\overline{\text{DREQn}}$ signal on the rising edge of the internal system clock. If $\overline{\text{DREQn}}$ is sampled low when the corresponding channel is in Ready state, the DMAC starts transferring data. To detect the low level of $\overline{\text{DREQn}}$, clear the PosE bit (bit 13) of the CCRn register to 0. The $\overline{\text{DACKn}}$ signal is also low active.

  Once the external device has asserted $\overline{\text{DREQn}}$, it must be held low until $\overline{\text{DACKn}}$ is asserted. If $\overline{\text{DREQn}}$ is deasserted before $\overline{\text{DACKn}}$ is asserted, the DMAC may not recognize the transfer request.

  If $\overline{\text{DREQn}}$ is not sampled low, the DMAC assumes that there is no transfer request for the channel and starts transferring data for another channel or relinquishes the bus, entering Ready state.

  The quantity of data transferred with a single transfer request is specified in the TrSiz field (bits 3 and 2) of the CCRn register.

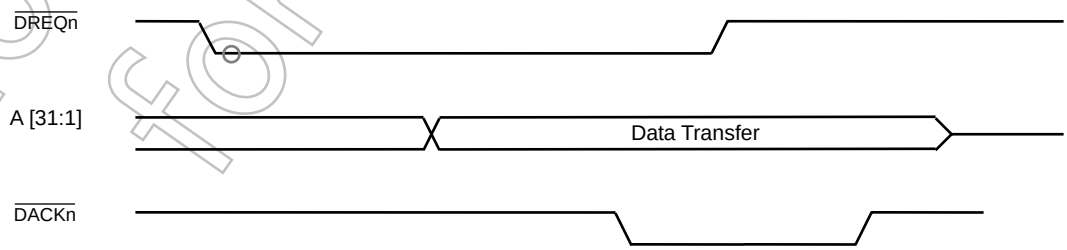


Figure 10.13 Transfer Request Timing (Level Mode)

- Edge mode

  In Edge mode, the DMAC is driven by the falling edge of the DREQn signal. If the DMAC detects the falling edge of DREQn on the rising edge of the internal system clock (samples DREQn high on the previous system clock edge and low on the current edge) when the corresponding channel is in Ready state, the DMAC assumes that there is a transfer request on the channel and starts transferring data. To detect the falling edge of DREQn, clear the PosE bit (bit 13) and Lev bit (bit 12) of the CCRn register to 0. The DACKn signal is low active.

  After asserting the DACKn signal, the DMAC transfers next data if it detects another falling edge of DREQn. If the DMAC does not detect a falling edge of DREQn after asserting DACKn, it assumes that there is no transfer request for the channel and starts transferring data for another channel or relinquishes the bus, entering Ready state.

  The quantity of data transferred with a single transfer request is specified in the TrSiz field (bits 3 and 2) of the CCRn register.

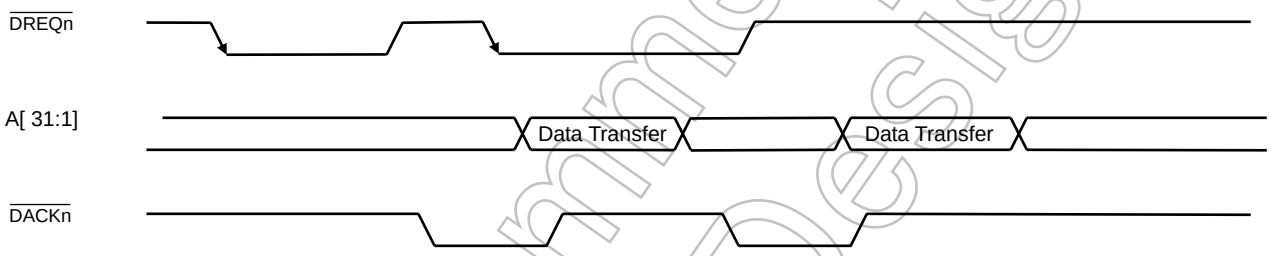


Figure 10.14 Transfer Request Timing (Edge Mode)

10.4.3 DMA Address Modes

DMA transfer is generally performed in either of two address modes: dual-address mode and single-address mode. In dual-address mode, both the source and destination devices are explicitly addressed. In single-address mode, only either the source device or the destination device is explicitly addressed. The TMP1962, however, supports dual-address mode only.

In dual-address mode, two bus transfers occur: a read from the source device, and a write to the destination device. In the source read cycle, data is read from the source address and placed in the DMAC internal Data Holding Register (DHR). Then, in the destination write cycle, the data held in the DHR is written to the destination address.

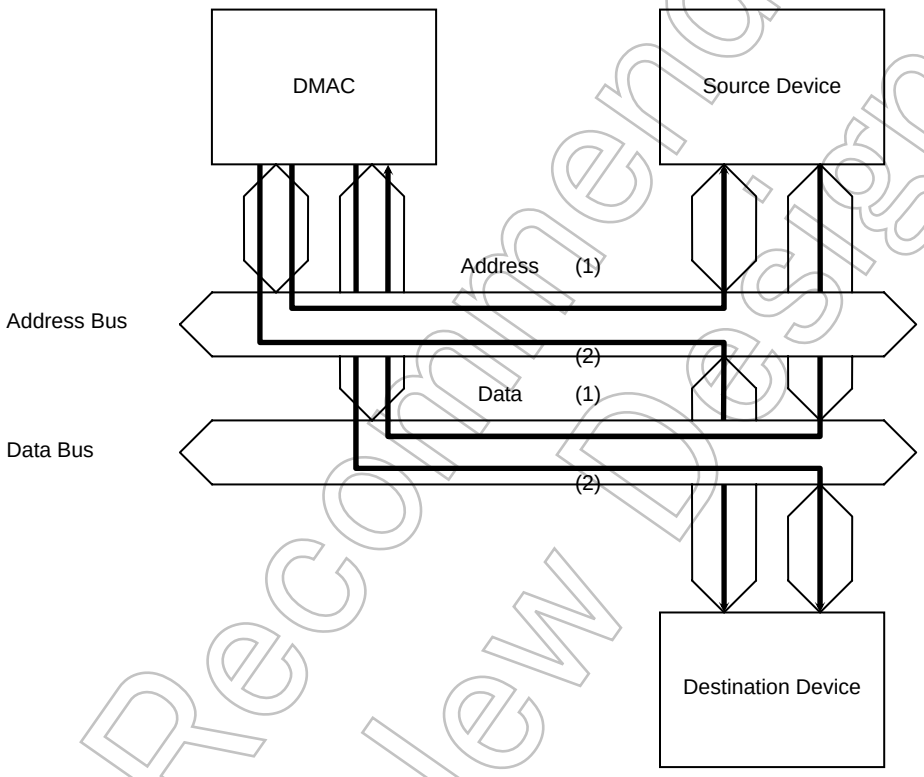


Figure 10.15 Dual-Address Transfer Mode

The transfer size programmed into the CCRn.TrSiz field determines the amount of data that is transferred from a source device to a destination device in response to a DMA request. The transfer size can be 32 bits, 16 bits or 8 bits.

The internal DHR is a 32-bit register that serves as a buffer for the data being transferred from a source device to a destination device during dual-address mode.

Memory accesses occur in a manner to fulfill the CCRn.TrSiz setting. Remember that the CS/Wait Controller supports either 16-bit or 8-bit bus accesses for external memory. If the DMA transfer size is programmed to 32 bits in CCRn.TrSiz, DMA read and write cycles each take up to four bus cycles to complete. A 16-bit data bus, as programmed in the CS/Wait Controller, requires two independent bus cycles to complete a 32-bit transfer. Likewise, an 8-bit data bus requires four independent bus cycles to complete a 32-bit transfer.

Memory-to-I/O and I/O-to-memory DMA transfers are governed by the setting of the CCRn.DPS field in addition to the setting of CCRn.TrSiz. The DPS field defines the port size of a source or destination I/O peripheral. The I/O port size can be 32 bits, 16 bits or 8 bits.

If the transfer size is equal to the I/O port size, an I/O access takes a single read or single write cycle. If the I/O port size is less than the programmed transfer size, the internal 32-bit DHR serves as a buffer for the data being transferred. For example, assume that the transfer size is programmed to 32 bits. If the source I/O port size is 8 bits and the destination memory width is 32 bits, then four 8-bit read cycles occur, followed by a 32-bit write cycle. (If the destination is an external memory with a 16-bit data bus, the write cycle takes two bus cycles.) The 32 bits of data are buffered in the DHR until the destination write cycle occurs.

Source and destination addresses can be programmed to increment or decrement after each transfer. The SARn and DARn change, if so programmed, after each data transfer, depending on the transfer size, i.e., the programmed TrSiz value. The BRCn is decremented by TrSiz for each data transfer. It is forbidden to program the device port size (DPS) to a value greater than the DMA transfer size (TrSiz). The relationships between TrSiz and DPS are summarized below.

Table 10.2 DMA Transfer Sizes and Device Port Sizes (in Dual-Address Mode)

TrSiz	DPS	Number of I/O Bus Cycles
0x (32 bits)	0x (32 bits)	1
0x (32 bits)	10 (16 bits)	2
0x (32 bits)	11 (8 bits)	4
10 (16 bits)	0x (32 bits)	Setting prohibited
10 (16 bits)	10 (16 bits)	1
10 (16 bits)	11 (8 bits)	2
11 (8 bits)	0x (32 bits)	Setting prohibited
11 (8 bits)	10 (16 bits)	Setting prohibited
11 (8 bits)	11 (8 bits)	1

10.4.4 DMA Channel Operation

Each DMA channel is started by setting the Str bit in the CCRn to 1. Once started, the DMAC checks the channel setups for configuration errors. If no configuration error is present, the channel enters Ready state.

When a DMA request is detected while in Ready state, the DMAC arbitrates for the bus and begins transferring data.

The channel can terminate by normal completion or from an error. The state of termination is indicated in the CSRn.

Channel startup

A DMA channel is started by setting the Str bit in the CCRn.

Once started, the DMAC checks the channel setups for configuration errors. If a configuration error is detected, the channel terminates abnormally. If no configuration error is present, the channel enters Ready state. Once a channel enters Ready state, the Act bit in the CSRn is set to 1.

If the channel is programmed for internal requests, the channel requests the bus and starts

transferring data immediately. If the channel is programmed for external requests, $\overline{\text{INTDREQn}}$ or $\overline{\text{DREQn}}$ must be asserted before the channel requests the bus.

Channel termination

A DMA channel can terminate by normal completion or from an error. The status of a DMA operation can be determined by reading the CSRn.

A channel terminates abnormally if an attempt is made to set the Str bit in the CCRn when the NC or AbC bit in the CSRn is set.

Normal termination

A DMA channel terminates by normal completion in the following case. Normal completion always occurs at the boundary of transfers programmed into the CCRn.TrSize field.

- Data transfers have terminated, with the BCRn decremented to 0.

Abnormal termination

The paragraphs that follow summarize the cases in which a DMA channel terminates from an error.

- Configuration errors

  A configuration error results when the channel initialization contains inconsistencies or errors. A configuration error is reported before any data transfer takes place; therefore, in case of a configuration error, the SARn, DARn and BCRn remain unaltered. When a DMA channel has terminated from a configuration error, the AbC and Conf bits in the CSRn are set. A configuration error occurs for the following cases:

- Both the CCRn.SIO and CCRn.DIO bits are set.
- The CCRn.Str bit is set when the NC or AbC bit in the CSRn is set.

- The BCRn contains a value that is not an integer multiple of the transfer size programmed into the CCRn.TrSiz field.
- The SARn or DARn contains a value that is not an integer multiple of the transfer size programmed into the CCRn.TrSiz field.
- The CCRn.TrSiz and CCRn.DPS fields contain illegal combinations.
- The CCRn.Str bit is set when the BCRn contains a value of zero.
- Bus errors

  When a DMA channel has terminated from a bus error, the AbC bit and the BES or BED bit in the CSRn are set.
- A bus error has been reported during a source read or destination write cycle.

Note: The contents of the BCRn, SARn and DARn are not guaranteed when a channel has terminated due to a bus error. Chapter 20 lists the reserved addresses that, if accessed, cause a bus error.

10.4.5 DMA Channel Priority

The DMAC provides a fixed priority for the eight channels, with channel 0 always having the highest priority and channel 7 the lowest. For example, when transfer requests occur on channels 0 and 1 simultaneously, the channel 0 request is serviced first. The channel 1 request is left pending. So that the channel 1 request is serviced, it must be maintained until data transfer completes on channel 0. Remember that the internally generated request is kept until the servicing of the request is finished. External transfer requests come from the Interrupt Controller (INTC). The INTC can program any interrupts to be used as a DMA trigger instead of as an interrupt request. If such an interrupt is programmed for edge sensitivity, the INTC internally maintains a transfer request. However, a level-sensitive interrupt is not held in the INTC; thus the interrupt request signal must remain asserted until the servicing of the DMA request begins.

A higher-priority channel always gets the attention of the DMAC. If a transfer request occurs on channel 0 while a request on channel 1 is being serviced, the servicing of the channel 1 request is suspended temporarily in order to service the channel 0 request first. After the channel 0 request has been serviced, channel 1 resumes the remaining data transfer.

Channel transitions take place at the boundary of a transfer size programmed for the current channel being serviced; that is, after all data in the DHR are written to a destination.

Interrupts

The DMAC can generate an interrupt request (INTDMAn) to the TX19 core processor upon the completion of a channel operation: either by normal channel termination or by abnormal termination of a bus cycle.

- Normal completion interrupt

  When a channel operation terminates by normal completion, the NC bit in the CSRn is set to 1. At this time, if the NIEn bit in the CCRn is set, an interrupt request is generated to the TX19 core processor.
- Abnormal completion interrupt

  When a channel operation terminates abnormally, the AbC bit in the CSRn register is set to 1. At this time, if the AbIEn bit in the CCRn register is set, an interrupt request is generated to the TX19 core processor.

10.5 DMA Transfer Timing

All DMAC operations are synchronous to the rising edges of the internal system clock.

10.5.1 Dual-Address Mode

- Memory-to-memory transfer

  Figure 10.16 shows a DMA cycle from one external 16-bit memory to another, with the transfer size programmed to 16 bits. A block of data is transferred until the BCRn register reaches 0.

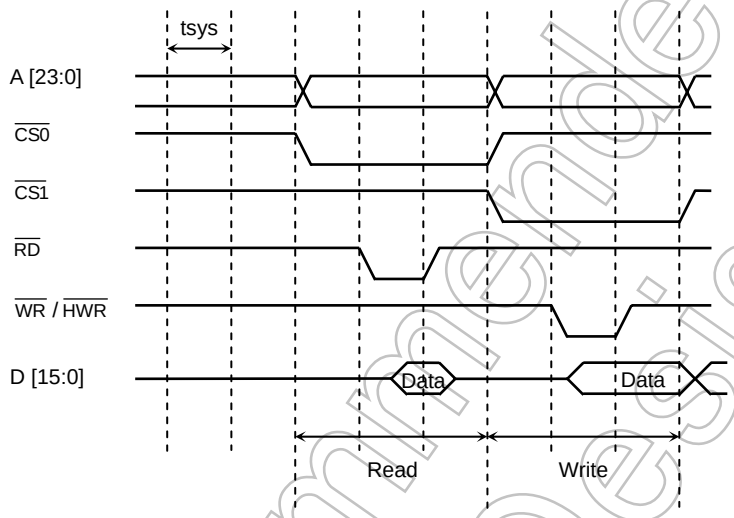


Figure 10.16 Memory-to-Memory Transfer (Dual-Address Mode)

- Memory-to-I/O transfer

  Figure 10.17 shows a DMA cycle from a 16-bit memory to an 8-bit I/O peripheral, with the transfer size programmed to 16 bits.

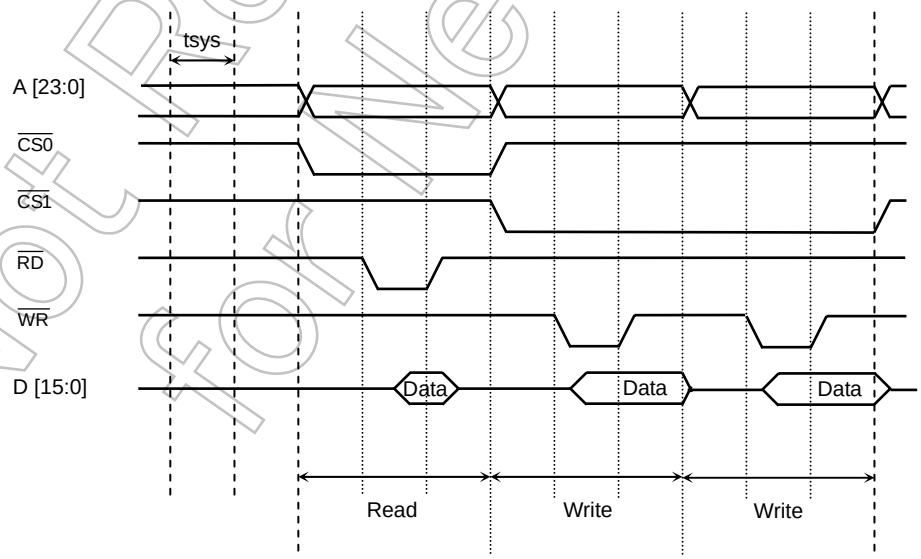


Figure 10.17 Memory-to-I/O Transfer (Dual-Address Mode)

- I/O-to-memory transfer

  Figure 10.18 shows a DMA cycle from an 8-bit I/O peripheral to a 16-bit memory, with the transfer size programmed to 16 bits.

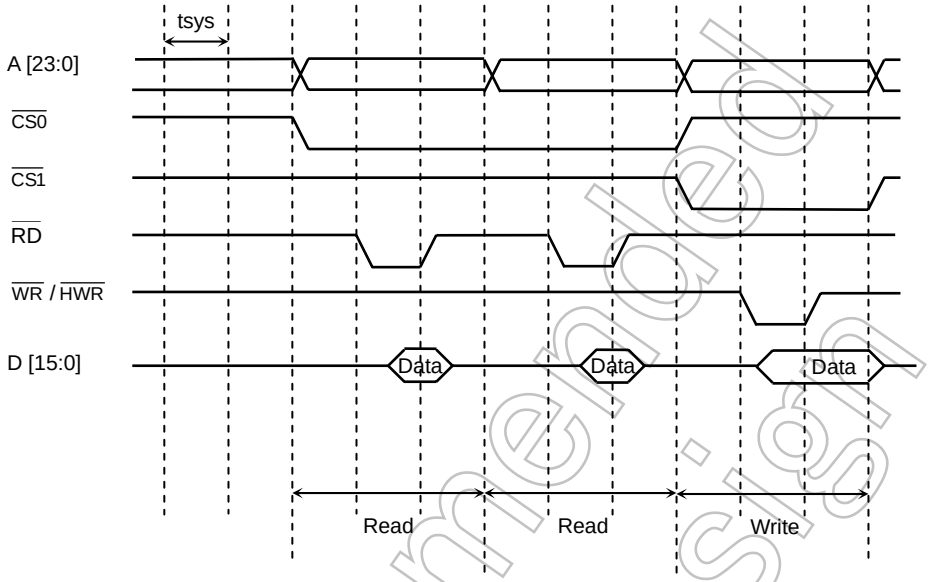


Figure 10.18 I/O-to-Memory Transfer (Dual-Address Mode)

10.5.2 Transfer Mode Responded to DREQn

- Transfer from the on-chip RAM to an external memory (multiplex bus, 5 waits insertion, level mode)

  Figure 10.19 shows two DMA cycles from the on-chip RAM to a 16-bit external memory, with the transfer size programmed to 16 bits.

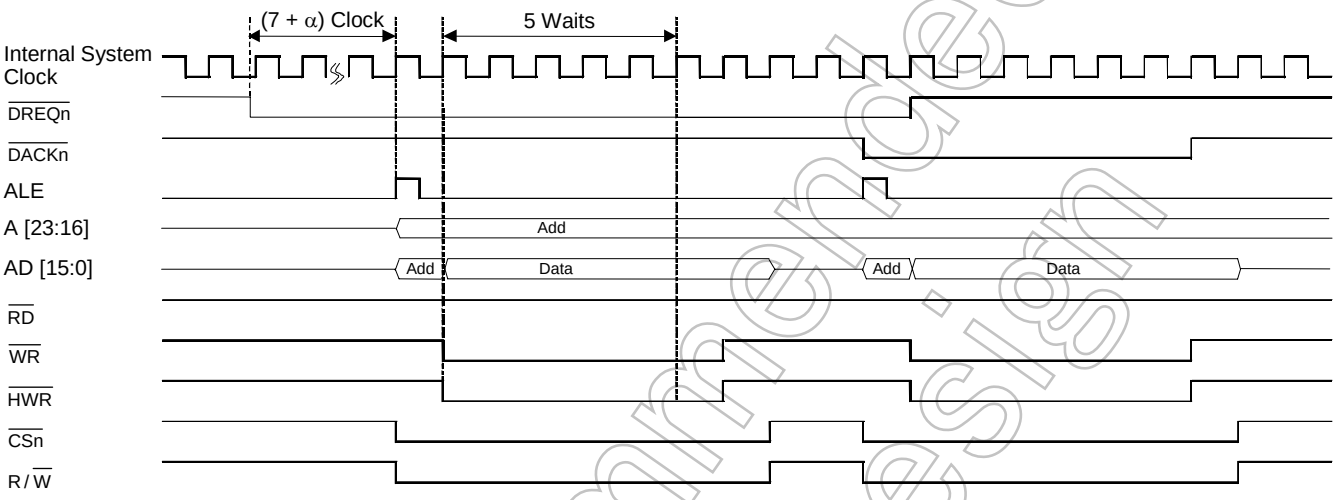


Figure 10.19 Level Mode (Transfer from the On-chip RAM to an External Memory)

- Transfer from an external memory to the on-chip RAM (multiplex bus, 5 waits insertion, level mode)

  Figure 10.20 shows two DMA cycles from a 16-bit external memory to the on-chip RAM, with the transfer size programmed to 16 bits.

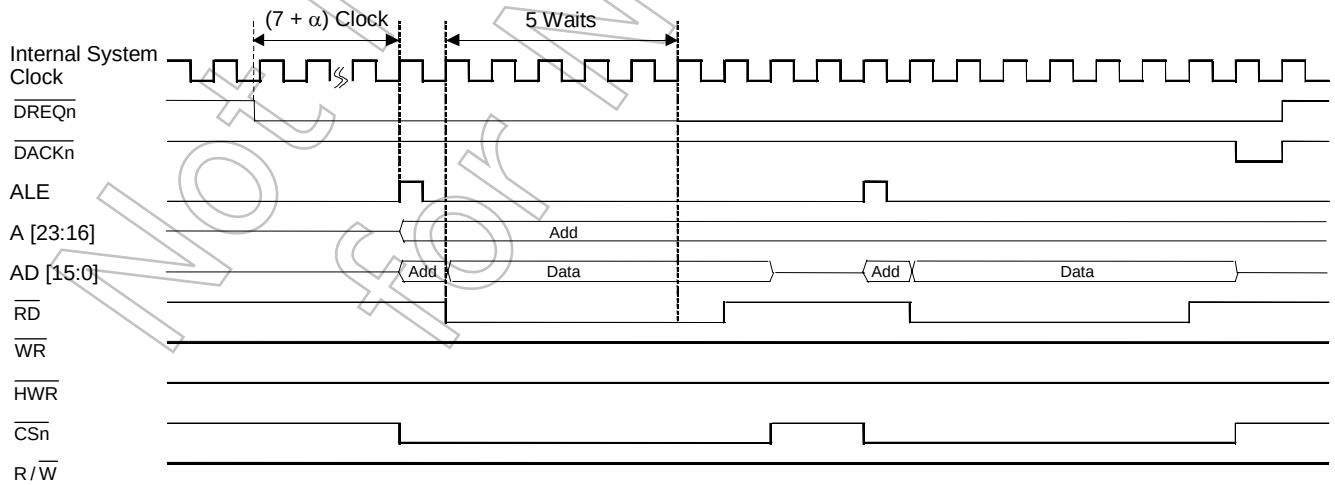


Figure 10.20 Level Mode (Transfer from an External Memory to the On-chip RAM)

- Transfer from the on-chip RAM to an external memory (separate bus, 5 waits insertion, level mode)

  Figure 10.21 shows two DMA cycles from the on-chip RAM to a 16-bit external memory, with the transfer size programmed to 16 bits.

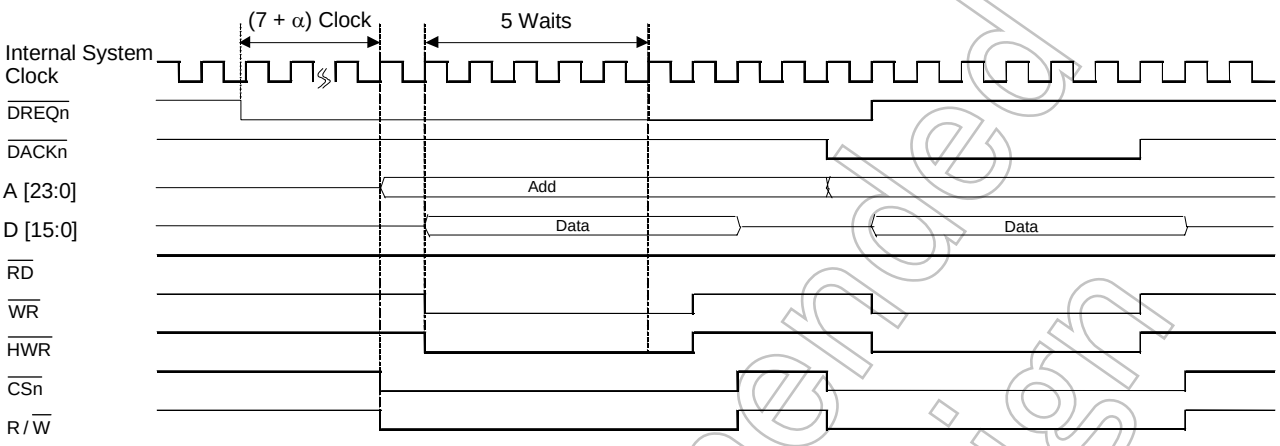


Figure 10.21 Level Mode (Transfer from the On-chip RAM to an External Memory)

- Transfer from an external memory to the on-chip RAM (separate bus, 5 waits insertion, level mode)

  Figure 10.22 shows two DMA cycles from a 16-bit external memory to the on-chip RAM, with the transfer size programmed to 16 bits.

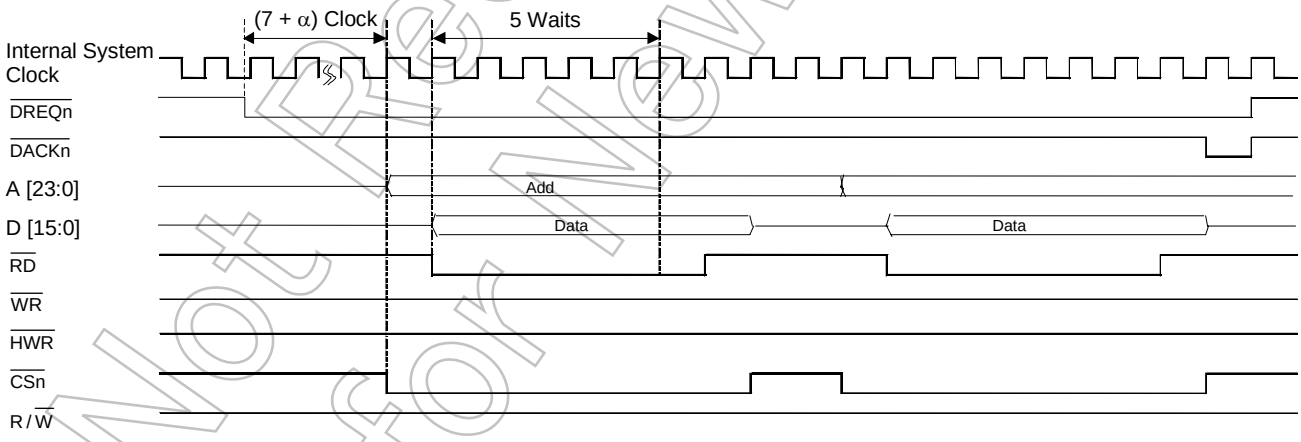


Figure 10.22 Level Mode (Transfer from an External Memory to the On-chip RAM)

- Transfer from the on-chip RAM to an external memory (multiplex bus, 5 waits insertion, edge mode)

  Figure 10.23 shows a DMA cycle from the on-chip RAM to a 16-bit external memory, with the transfer size programmed to 16 bits.

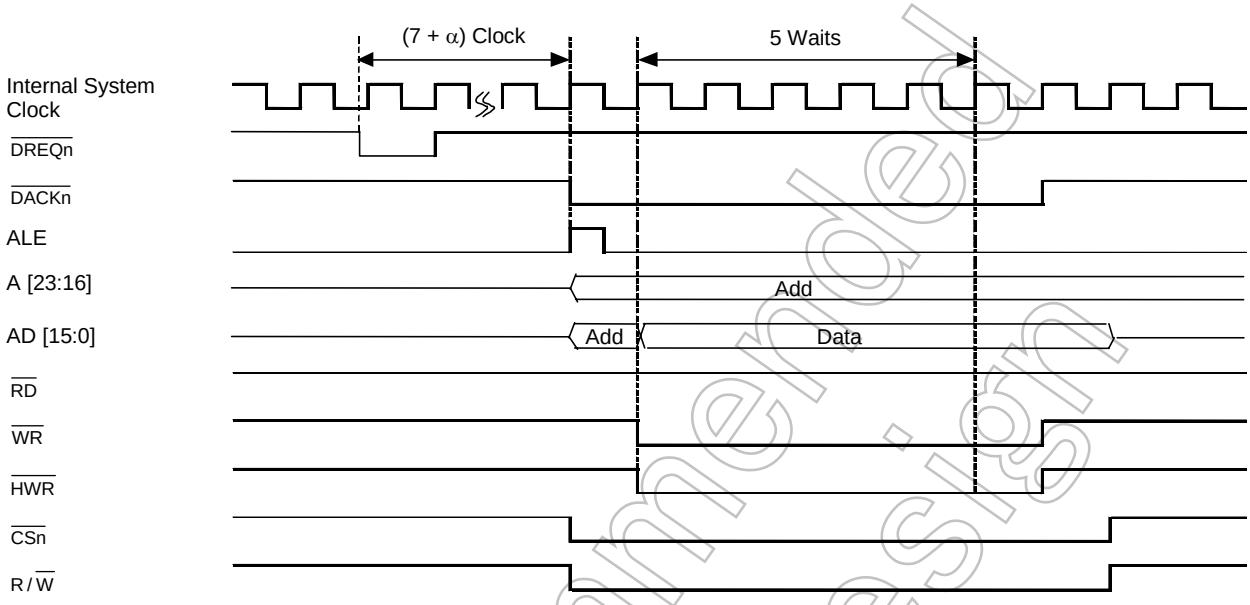


Figure 10.23 Edge Mode (Transfer from the On-chip RAM to an External Memory)

- Transfer from an external memory to the on-chip RAM (multiplex bus, 5 waits insertion, edge mode)

  Figure 10.24 shows a DMA cycle from a 16-bit external memory to the on-chip RAM, with the transfer size programmed to 16 bits.

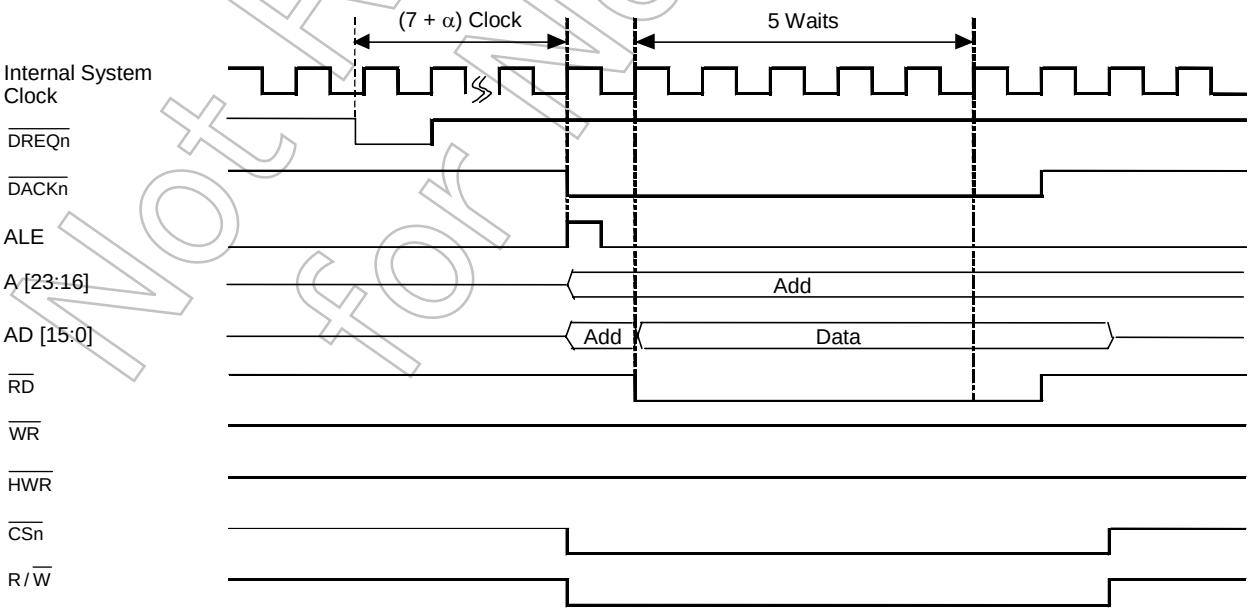


Figure 10.24 Edge Mode (Transfer from an External Memory to the On-chip RAM)

- Transfer from the on-chip RAM to an external memory (separate bus, 5 waits insertion, edge mode)

  Figure 10.25 shows a DMA cycle from the on-chip RAM to a 16-bit external memory, with the transfer size programmed to 16 bits.

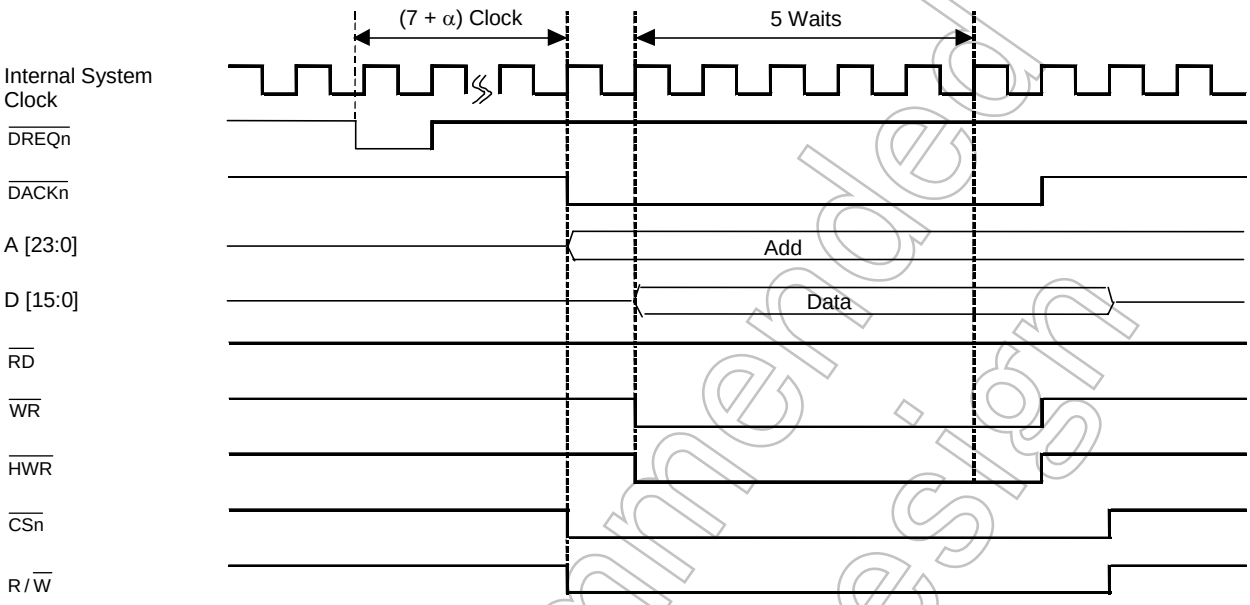


Figure 10.25 Edge Mode (Transfer from the On-chip RAM to an External Memory)

- Transfer from an external memory to the on-chip RAM (separate bus, 5 waits insertion, edge mode)

  Figure 10.26 shows a DMA cycle from a 16-bit external memory to the on-chip RAM, with the transfer size programmed to 16 bits.

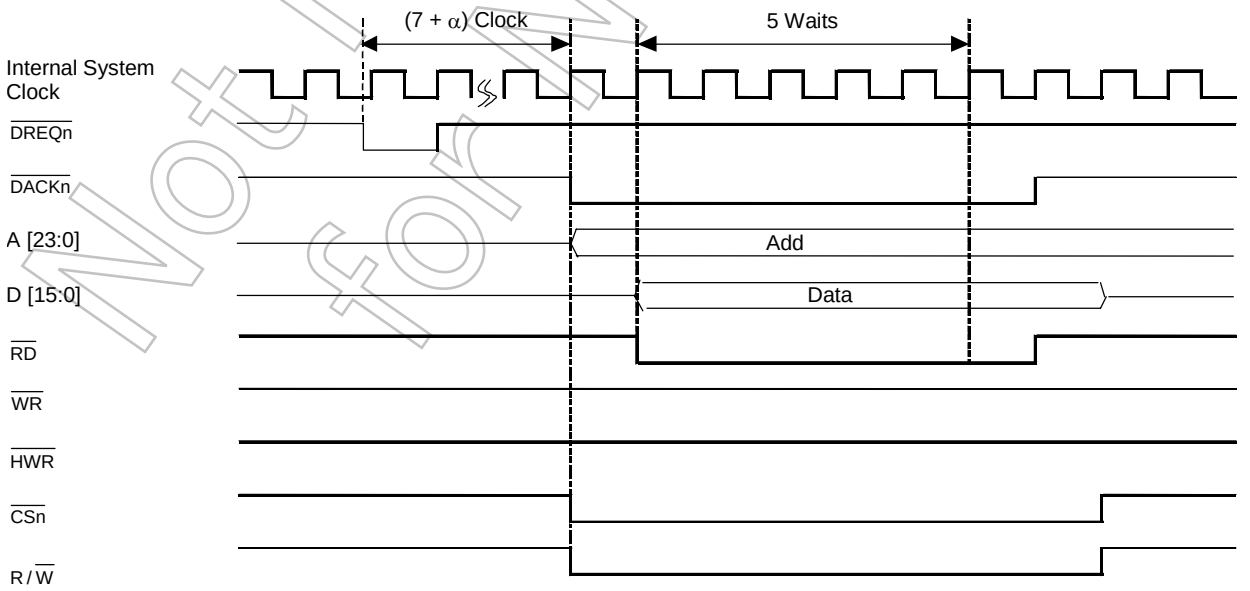


Figure 10.26 Edge Mode (Transfer from an External Memory to the On-chip RAM)

10.6 Programming Example

The following illustrates the programming required to transfer data from an SIO receive buffer (SCnBUF) to the on-chip RAM.

(1) DMAC settings:

- DMA channel used: Channel 0
- Source address: SC1BUF
- Destination address: 0xFFFF_9800 (physical address)
- Number of bytes transferred: 256

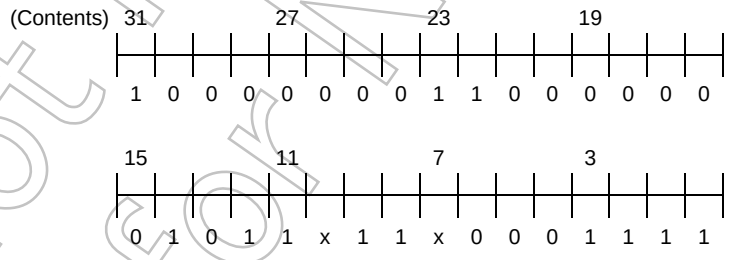
(2) SIO settings:

- Data format: 8 bits, UART
- SIO channel used: Channel 1
- Transfer rate: 9600 bps

DMA channel 0 is used for the transfer. The SIO1 receive interrupt is used as a trigger to start the DMA channel.

(3) DMA channel 0 settings:

```
DCR     ←  0x8000_0000          /* Reset DMAC * /
IMCD    ←  31          23
           xxxx, xxxx, xx10, x100   /* Interrupt level = 4 (arbitrary) * /
INTCLR  ←  0x36                 /* IVR[9:4] * /
DTCR0   ←  0x0000_0000          /* DACM = 000 * /
                                /* SACM = 000 * /
SAR0    ←  0xFFFF_F208          /* Physical address of SC1BUF */
DAR0    ←  0xFFFF_9800          /* Physical address of destination */
BCR0    ←  0x0000_00FF          /* 256 (Number of bytes to be transferred) /
CCR0    ←  0x80C0_5B0F
```



(4) SIO channel 1 settings:

```
IMC4    ←  31          23
           xxxx, xxxx, xx11, 1000   /* Use INTRX1 as a DMA trigger and select DMA ch. 0 * /
INTCLR  ←  0x12                 /* IVR[9:4]; clear INTRX1 * /
SC1MOD0 ←  0x29                 /* UART mode, 8-bit data format, baud rate generator * /
SC1CR   ←  0x00
BR1CR   ←  0x1F                 /* @fc = 40.5 MHz (approx. 1.05 Mbps) */
```

11. 8-Bit Timers (TMRAs)

The TMP1962 has a twelve-channel 8-bit timer (TMRA0-TMRAB), which is comprised of six modules named TMRA01, TMRA23, TMRA45, TMRA67, TMRA89 and TMRAAB. The TMRA01 contains the TMRA0 and the TMRA1, the TMRA23 contains the TMRA2 and TMRA3, and so on. Each timer module has the following operating modes:

- 8/16/24/32-Bit Interval Timer mode
- 8-Bit Programmable Pulse Generation (PPG) mode (variable frequency, variable duty cycle)
- 8-Bit Pulse Width Modulated (PWM) Signal Generation mode (fixed frequency, variable duty cycle)

Figure 11.1 is a block diagram of the TMRA01. The main components of a timer channel are an 8-bit up-counter, an 8-bit comparator and an 8-bit timer register. Two timer channels share a prescaler and a timer flip-flop.

A total of eight registers provide control over the operating modes and timer flip-flops for each timer module. The six modules are functionally equivalent and can be independently programmed. In the following sections, any references to the TMRA01 also apply to the other modules. Table 11.1 gives the pins and registers for the six timer modules.

Table 11.1 Pins and Registers for TMRAs (1/3)

Module / Specifications		TMRA01	TMRA23
External Pins	External clock input	TA0IN (Shared with PA0)	TA2IN (Shared with PA2)
	Timer flip-flop output	TA1OUT (Shared with PA1)	TA3OUT (Shared with PA3)
Registers (Addresses)	Timer Run register	TA01RUN (0xFFFF_F103)	TA23RUN (0xFFFF_F10B)
	Timer Control register	TA01CR (0xFFFF_F102)	TA23CR (0xFFFF_F10A)
	Timer registers	TA0REG (0xFFFF_F101) TA1REG (0xFFFF_F100)	TA2REG (0xFFFF_F109) TA3REG (0xFFFF_F108)
	Timer Mode register	TA01MOD (0xFFFF_F107)	TA23MOD (0xFFFF_F10F)
	Timer Flip-Flop Control register	TA1FFCR (0xFFFF_F106)	TA3FFCR (0xFFFF_F10E)
	Timer Interrupt Mask register	TAG0IM (0xFFFF_F105)	
	Timer Interrupt Status register	TAG0ST (0xFFFF_F104)	

Table 11.1 Pins and Registers for TMRAs (2/3)

Module / Specifications		TMRA45	TMRA67
External Pins	External clock input	TA4IN (Shared with PL0)	TA6IN (Shared with PL1)
	Timer flip-flop output	TA5OUT (Shared with PA4)	TA7OUT (Shared with PA5)
Registers (Addresses)	Timer Run register	TA45RUN (0xFFFF_F113)	TA67RUN (0xFFFF_F11B)
	Timer Control register	TA45CR (0xFFFF_F112)	TA67CR (0xFFFF_F11A)
	Timer registers	TA4REG (0xFFFF_F111) TA5REG (0xFFFF_F110)	TA6REG (0xFFFF_F119) TA7REG (0xFFFF_F118)
	Timer Mode register	TA45MOD (0xFFFF_F117)	TA67MOD (0xFFFF_F11F)
	Timer Flip-Flop Control register	TA5FFCR (0xFFFF_F116)	TA7FFCR (0xFFFF_F11E)
	Timer Interrupt Mask register	TAG1IM (0xFFFF_F115)	
	Timer Interrupt Status register	TAG1ST (0xFFFF_F114)	

Table 11.1 Pins and Registers for TMRAs (3/3)

Module / Specifications		TMRA89	TMRAAB
External Pins	External clock input	TA8IN (Shared with PL2)	TAAIN (Shared with PL3)
	Timer flip-flop output	TA9OUT (Shared with PA6)	TABOUT (Shared with PA7)
Registers (Addresses)	Timer Run register	TA89RUN (0xFFFF_F123)	TAABRUN (0xFFFF_F12B)
	Timer Control register	TA89CR (0xFFFF_F122)	TAABCR (0xFFFF_F12A)
	Timer registers	TA8REG (0xFFFF_F121) TA9REG (0xFFFF_F120)	TAAREG (0xFFFF_F129) TABREG (0xFFFF_F128)
	Timer Mode register	TA89MOD (0xFFFF_F127)	TAABMOD (0xFFFF_F12F)
	Timer Flip-Flop Control register	TA9FFCR (0xFFFF_F126)	TABFFCR (0xFFFF_F12E)
	Timer Interrupt Mask register	TAG2IM (0xFFFF_F125)	
	Timer Interrupt Status register	TAG2ST (0xFFFF_F124)	

11.1 TMRA Block Diagram

Only the TMRA01 block diagram is shown. The other timer modules are the same as the TMRA01 except the pin and register names.

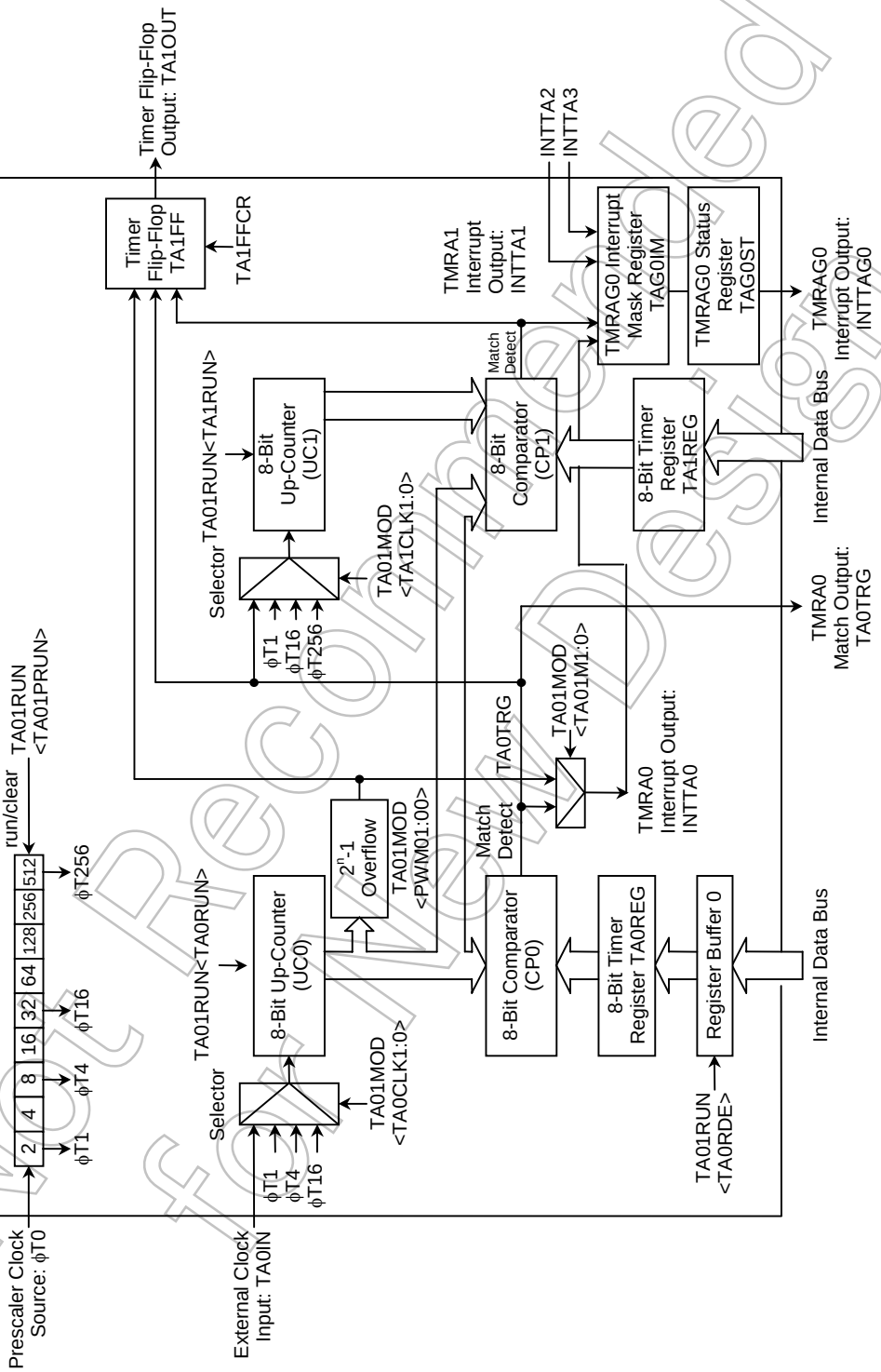


Figure 11.1 TMRA01 Block Diagram

11.2 Timer Components

11.2.1 Prescaler

The TMRA01 has a 9-bit prescaler that slows the rate of a clocking source to the counters. The prescaler clock source (ϕT0) can be selected from fperiph/4, fperiph/8 and fperiph/16 by programming the PRCK[1:0] field of the SYSCR0 located within the CG.

fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The TA01PRUN bit in the TA01RUN register allows the enabling and disabling of the prescaler for the TMRA01. A write of 1 to this bit starts the prescaler. A write of 0 to this bit clears and halts the prescaler. Table 11.2 shows prescaler output clock resolutions.

Table 11.2 Prescaler Output Clock Resolutions

@fc = 40.5 MHz

Peripheral Clock Source FPSEL	Clock Gear Value GEAR[1:0]	Prescaler Clock Source PRCK[1:0]	Prescaler Output Clock Resolution			
			ϕT1	ϕT4	ϕT16	ϕT256
0(fgear)	00(fc)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
		01(fperiph/8)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{12}(101 μs)
		10(fperiph/4)	fc/2^3(0.2 μs)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^{11}(50.6 μs)
	01(fc/2)	00(fperiph/16)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{10}(25.3 μs)	fc/2^{14}(405 μs)
		01(fperiph/8)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
		10(fperiph/4)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{12}(101 μs)
	10(fc/4)	00(fperiph/16)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{11}(50.6 μs)	fc/2^{15}(809 μs)
		01(fperiph/8)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{10}(25.3 μs)	fc/2^{14}(405 μs)
		10(fperiph/4)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
	11(fc/8)	00(fperiph/16)	fc/2^8(6.32 μs)	fc/2^{10}(25.3 μs)	fc/2^{12}(101 μs)	fc/2^{16}(1618 μs)
		01(fperiph/8)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{11}(50.6 μs)	fc/2^{15}(809 μs)
		10(fperiph/4)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{10}(25.3 μs)	fc/2^{14}(405 μs)
1(fc)	00(fc)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
		01(fperiph/8)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{12}(101 μs)
		10(fperiph/4)	fc/2^3(0.2 μs)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^{11}(50.6 μs)
	01(fc/2)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
		01(fperiph/8)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{12}(101 μs)
		10(fperiph/4)	fc/2^3(0.2 μs)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^{11}(50.6 μs)
	10(fc/4)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
		01(fperiph/8)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{12}(101 μs)
		10(fperiph/4)	—	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^{11}(50.6 μs)
	11(fc/8)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{13}(202 μs)
		01(fperiph/8)	—	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{12}(101 μs)
		10(fperiph/4)	—	fc/2^5(0.79 μs)	fc/2^7(3.16 μs)	fc/2^{11}(50.6 μs)

Note 1: The prescaler's output clock ϕTn must be selected so that ϕTn < fsys/2 is satisfied.
Note 2: Do not change the clock gear value while the timer is running.
Note 3: The - character means "Setting prohibited."

11.2.2 Up-Counters (UC0 and UC1)

The timer module contains two 8-bit binary up-counters, each of which is driven by a clock independently selected by the TA01MOD register.

The clock input to the UC0 is either one of three prescaler outputs (ϕT1, ϕT4, ϕT16) or the external clock applied to the TA0IN pin. Which clock is to use is programmed into the TA0CLK[1:0] field of the TA01MOD register.

Possible clock sources for the UC1 depend on the selected operating mode. If cascade connection is not used, the clock input to the UC1 is either one of three prescaler outputs (ϕT1, ϕT16, ϕT256) or the TMRA0 comparator match-detect output.

If cascade connection is used to select 16-Bit Timer mode, the clock input to the UC1 is the UC0 overflow output. If cascade connection is used for 24-Bit Timer mode, the UC1 overflow output is used as the clock input to the UC2 of the TMRA23. If cascade connection is used for 32-Bit Timer mode, the UC2 overflow output is used as the clock input to the UC3.

The TA0RUN and TA1RUN bits in the TA01RUN register are used to start counting and to stop and clear the counter. Upon reset, the up-counter is set to 00H and the whole timer module is disabled.

11.2.3 Timer Registers (TA0REG and TA1REG)

Each timer register is an 8-bit register containing a time constant. When the up-counter reaches the time constant value in the timer register, the comparator block generates a match-detect signal. When the time constant is set to 00H, a match occurs upon a counter overflow.

One of the two timer registers, TA0REG, is double-buffered. The double-buffering function can be enabled and disabled through the programming of the TA0RDE bit in the TA01RUN: 0 = disable, 1 = enable.

If double-buffering is enabled, the TA0REG latches a new time constant value from the register buffer. This takes place upon detection of a 2^n-1 overflow in PWM mode and upon a match between the UC0 and the TA1REG in PPG mode. Double-buffering must be disabled in interval timer modes.

A reset clears the TA01RUN.TA0RDE bit to 0, disabling the double-buffering function. To use this function, the TA01RUN.TA0RDE bit must be set to 1 after loading the TA0REG with a time constant. When TA01RUN.TA0RDE = 1, the next time constant can be written to the register buffer.

Figure 11.2 illustrates the double-buffer structure for the TA0REG.

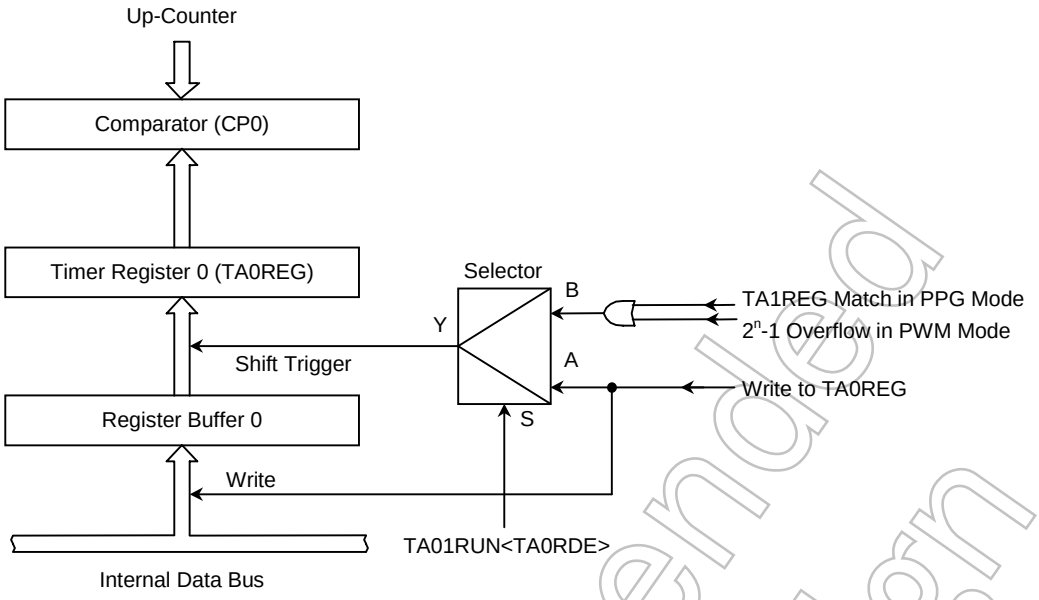


Figure 11.2 Timer Register 0 (TA0REG) Structure

Note 1: The timer register and the corresponding register buffer are mapped to the same address. When TA01RUN.TA0RDE = 0, a time constant value is written to both the timer register and the register buffer; when TA01RUN.TA0RDE = 1, a time constant value is written only to the register buffer.

Note 2: The timer registers are write-only registers.

11.2.4 Comparators (CP0 and CP1)

The comparator compares the output of the 8-bit up-counter with a time constant value in the 8-bit timer register. When a match is detected, an interrupt (INTTA0/INTTA1) is generated and the timer flip-flop is toggled, if so enabled.

11.2.5 Timer Flip-Flop (TA1FF)

The timer flip-flop (TA1FF) is toggled, if so enabled, each time the comparator match-detect output is asserted. The toggling of the timer flip-flop can be enabled and disabled through the programming of the TAFF1IE bit in the TA1FFCR.

A reset clears the TAFF1IE bit, disabling the toggling of the TA1FF. The TA1FF can be initialized to 1 or 0 by writing 01 or 10 to the TAFF1C[1:0] field in the TA1FFCR. Additionally, a write of 00 by software causes the TA1FF to be toggled to the opposite value.

The value of the TA1FF can be driven onto the TA1OUT pin, which is multiplexed with PA1. The Port A registers (PACR and PAFC) must be programmed to configure the PA1/TA1OUT pin as TA1OUT.

11.2.6 Interrupt Mask Register (TAG0IM)

TMRA interrupts are classified into the following three groups:

Interrupt Group 0 (INTTAG0): INTTA0, INTTA1, INTTA2, INTTA3

Interrupt Group 1 (INTTAG1): INTTA4, INTTA5, INTTA6, INTTA7

Interrupt Group 2 (INTTAG2): INTTA8, INTTA9, INTTAA, INTTAB

Interrupts that belong to the same group are assumed as the same interrupt source when sent to the Interrupt Controller (INTC). An interrupt mask register (TAGnIM) is provided for each interrupt group. Setting a bit in the TAGnIM masks the corresponding interrupt source so that the INTC will not generate the interrupt. Upon reset, all interrupt sources are enabled (not masked).

11.2.7 Interrupt Status Register (TAG0ST)

An interrupt status register (TAGnST) is provided for each interrupt group. When an interrupt occurs, the flag bit corresponding to the interrupt source is set to 1. Reading the TAGnST register clears all bits that have been set. Any interrupt sources masked in the TAGnIM register are disabled although flags are set when corresponding interrupts occur.

Note: If any of INTTA0 to INTTA3 occurs while the TAG0ST is being read, the corresponding flags are handled as follows: (If the flag is set and read simultaneously.)

- **If 1 is read, the flag is cleared.**
- **If 0 is read, the flag is set after the read.**

11.3 Register Description

TMRA01 Run Register

TA01RUN (0xFFFF_F103)

	7	6	5	4	3	2	1	0
Bit Symbol	TA0RDE	TA01C1	TA01C0		I2TA01	TA01PRUN	TA1RUN	TA0RUN
Read/Write	R/W				R/W			
Reset Value	0	0	0		0	0	0	0
Function	Double-buffering 0: Disable 1: Enable	Cascade connection 00: 8- or 16-bit mode 01: Setting prohibited 10: Setting prohibited 11: First stage of cascade			IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TA0RUN: Runs or stops the TMRA0.

TA1RUN: Runs or stops the TMRA1.

TA01PRUN: Runs or stops the TMRA01 prescaler.

I2TA01: Enables or disables the operation of the TMRA0-TMRA3 in IDLE mode.

TA01C[1:0]: Specifies how the TMRA01 is used in cascade connection. When this field is set to 00, either 8- or 16-bit mode is selected according to the settings of the TA01M[1:0] bits in the TMRA01 Mode register. When this field is set to 11 (first stage of cascade), the TMRA01 is combined with the TMRA23 to form a 24- or 32-bit timer.

TA0RDE: Enables or disables double-buffering.

Note: Bit 4 of the TA01RUN is read as undefined.

TMRA01 Control Register

TA01CR (0xFFFF_F102)

	7	6	5	4	3	2	1	0
Bit Symbol	TA01EN							
Read/Write	R/W							
Reset Value	0	0	0					
Function	TMRA01 operation 0: Disable 1: Enable	Must be written as 00.						

TA01EN: Enables or disables the operation of the TMRA01. If the TMRA01 is disabled, no clock pulses are supplied to the TMRA01 registers other than the TA01CR, so that power consumption in the system can be reduced (only the TA01CR can be read or written). To use the TMRA01, set the TA01EN bit to 1 before configuring other registers of the TMRA01. Once the TMRA01 operates, all settings in its registers are held if it is disabled. The TA01EN bit enables or disables the operation of the TMRA0-TMRA3. (Enable or disable all channels of the TMRA0-TMRA3.)

Note: Bits 5 and 6 of the TA01CR are read as 0.

TMRA0 Register

TA0REG (0xFFFF_F101)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA0REG are read as undefined.

TMRA1 Register

TA1REG (0xFFFF_F100)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA1REG are read as undefined.

TMRA01 Mode Register

TA01MOD (0xFFFF_F107)

	7	6	5	4	3	2	1	0
Bit Symbol	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6 - 1 10: 2^7 - 1 11: 2^8 - 1		TMRA1 clock source 00: TA0TRG 01: ϕT1 (prescaler) 10: ϕT16 (prescaler) 11: ϕT256 (prescaler)		TMRA0 clock source 00: TA0IN input 01: ϕT1 (prescaler) 10: ϕT4 (prescaler) 11: ϕT16 (prescaler)	

TA0CLK[1:0]: Selects the TMRA0 clock source.

TA1CLK[1:0]: Selects the TMRA1 clock source (when the TA01M[1:0] field is set to other than 01). When TA01M[1:0] = 01, the TMRA0 overflow output is always the TMRA1 clock source regardless of the settings in TA1CLK[1:0].

PWM0[1:0]: Selects the period for 8-bit PWM mode. The PWM period will be $(2^n - 1)$ x clock source period.

TA01M[1:0]: Selects the TMRA01 operating mode. When this field is set to 00, the TMRA01 is used as two independent 8-bit timers, TMRA0 and TMRA1.

TMRA1 Flip-Flop Control Register

TA1FFCR (0xFFFF_F106)

	7	6	5	4	3	2	1	0
Bit Symbol					TAFF1C1	TAFF1C0	TAFF1IE	TAFF1IS
Read/Write					R/W			
Reset Value					1	1	0	0
Function					00: Toggles TA1FF. (software toggle) 01: Sets TA1FF to 1. 10: Clears TA1FF to 0. 11: Don't care. This field is always read as 11.		TA1FF toggle enable 0: Disable 1: Enable	TA1FF toggle trigger 0: TMRA0 1: TMRA1

TAFF1IS: Specifies whether Timer Flip-Flop 1 (TA1FF) is toggled by a TMRA0 match detection signal or a TMRA1 match detection signal. This bit is valid only in 8-bit timer mode; it is a don't-care bit in other modes.

Note: Bits 4, 5, 6 and 7 of the TA1FFCR are read as undefined.

TMRAG0 Interrupt Mask Register

TAG0IM (0xFFFF_F105)

	7	6	5	4	3	2	1	0
Bit Symbol					TAIM3	TAIM2	TAIM1	TAIM0
Read/Write					R/W			
Reset Value					0	0	0	0
Function					1: Masks INTTA3.	1: Masks INTTA2.	1: Masks INTTA1.	1: Masks INTTA0.

Note: Bits 4, 5, 6 and 7 of the TAG0IM are read as undefined.

TMRAG0 Status Register

TAG0ST (0xFFFF_F104)

	7	6	5	4	3	2	1	0
Bit Symbol					INTTA3	INTTA2	INTTA1	INTTA0
Read/Write					R			
Reset Value					0	0	0	0
Function					0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

Note 1: Reading the TAG0ST register results in bits 0, 1, 2 and 3 being cleared.
Note 2: Bits 4, 5, 6 and 7 of the TAG0ST are read as undefined.
Note 3: The flag bit corresponding to the interrupt source being masked is set, but the interrupt is not signaled.

TMRA23 Run Register

TA23RUN (0xFFFF_F10B)

	7	6	5	4	3	2	1	0
Bit Symbol	TA2RDE	TA23C1	TA23C0		I2TA23	TA23PRUN	TA3RUN	TA2RUN
Read/Write	R/W				R/W			
Reset Value	0	0	0		0	0	0	0
Function	Double-buffering 0: Disable 1: Enable	Cascade connection 00: 8- or 16-bit mode 01: 24-bit cascade 10: 32-bit cascade 11: Setting prohibited			IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TA2RUN: Runs or stops the TMRA2.

TA3RUN: Runs or stops the TMRA3.

TA23PRUN: Runs or stops the TMRA23 prescaler.

I2TA23: Enables or disables the operation of the TMRA0-TMRA3 in IDLE mode.

TA23C[1:0]: Specifies how the TMRA23 is used in cascade connection. When this field is set to 00, either 8- or 16-bit mode is selected according to the settings of the TA23M[1:0] bits in the TMRA23 Mode register. When this field is set to 01 (24-bit cascade), the TMRA2 is cascaded with the TMRA01. When this field is set to 10 (32-bit cascade), the TMRA0 to TMRA3 are cascaded.

TA2RDE: Enables or disables double-buffering.

Note: Bit 4 of the TA23RUN is read as undefined.

TMRA23 Control Register

TA23CR (0xFFFF_F10A)

	7	6	5	4	3	2	1	0
Bit Symbol	TA23EN							
Read/Write	R/W							
Reset Value	0	0	0					
Function	TMRA23 operation 0: Disable 1: Enable	Must be written as 00.						

TA23EN: Enables or disables the operation of the TMRA23. If the TMRA23 is disabled, no clock pulses are supplied to the TMRA23 registers other than the TA23CR, so that power consumption in the system can be reduced (only the TA23CR can be read or written). To use the TMRA23, set the TA23EN bit to 1 before configuring other registers of the TMRA23. Once the TMRA23 operates, all settings in its registers are held if it is disabled.

Note: Bits 5 and 6 of the TA23CR are read as 0.

TMRA2 Register

TA2REG (0xFFFF_F109)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA2REG are read as undefined.

TMRA3 Register

TA3REG (0xFFFF_F108)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA3REG are read as undefined.

TMRA23 Mode Register

TA23MOD (0xFFFF_F10F)

	7	6	5	4	3	2	1	0
Bit Symbol	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6 - 1 10: 2^7 - 1 11: 2^8 - 1		TMRA3 clock source 00: TA2TRG 01: ϕT1 (prescaler) 10: ϕT16 (prescaler) 11: ϕT256 (prescaler)		TMRA2 clock source 00: TA2IN input 01: ϕT1 (prescaler) 10: ϕT4 (prescaler) 11: ϕT16 (prescaler)	

TA2CLK[1:0]: Selects the TMRA2 clock source.

TA3CLK[1:0]: Selects the TMRA3 clock source (when the TA23M[1:0] field is set to other than 01). When TA23M[1:0] = 01, the TMRA2 overflow output is always the TMRA3 clock source regardless of the settings in TA3CLK[1:0].

PWM2[1:0]: Selects the period for 8-bit PWM mode. The PWM period will be $(2^n - 1)$ x clock source period.

TA23M[1:0]: Selects the TMRA23 operating mode. When this field is set to 00, the TMRA23 is used as two independent 8-bit timers, TMRA2 and TMRA3.

TMRA3 Flip-Flop Control Register

TA3FFCR (0xFFFF_F10E)

	7	6	5	4	3	2	1	0
Bit Symbol					TAFF3C1	TAFF3C0	TAFF3IE	TAFF3IS
Read/Write					R/W			
Reset Value					1	1	0	0
Function					00: Toggles TA3FF. (software toggle) 01: Sets TA3FF to 1. 10: Clears TA3FF to 0. 11: Don't care. This field is always read as 11.		TA3FF toggle enable 0: Disable 1: Enable	TA3FF toggle trigger 0: TMRA2 1: TMRA3

TAFF3IS: Specifies whether Timer Flip-Flop 3 (TA3FF) is toggled by a TMRA2 match detection signal or a TMRA3 match detection signal. This bit is valid only in 8-bit timer mode; it is a don't-care bit in other modes.

Note: Bits 4, 5, 6 and 7 of the TA3FFCR are read as undefined.

TMRA45 Run Register

TA45RUN (0xFFFF_F113)

	7	6	5	4	3	2	1	0
Bit Symbol	TA4RDE	TA45C1	TA45C0		I2TA45	TA45PRUN	TA5RUN	TA4RUN
Read/Write	R/W				R/W			
Reset Value	0	0	0		0	0	0	0
Function	Double-buffering 0: Disable 1: Enable	Cascade connection 00: 8- or 16-bit mode 01: Setting prohibited 10: Setting prohibited 11: First stage of cascade			IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TA4RUN: Runs or stops the TMRA4.

TA5RUN: Runs or stops the TMRA5.

TA45PRUN: Runs or stops the TMRA45 prescaler.

I2TA45: Enables or disables the operation of the TMRA4-TMRA7 in IDLE mode.

TA45C[1:0]: Specifies how the TMRA45 is used in cascade connection. When this field is set to 00, either 8- or 16-bit mode is selected according to the settings of the TA45M[1:0] bits in the TMRA45 Mode register. When this field is set to 11 (first stage of cascade), the TMRA45 is combined with the TMRA67 to form a 24- or 32-bit timer.

TA4RDE: Enables or disables double-buffering.

Note: Bit 4 of the TA45RUN is read as undefined.

TMRA45 Control Register

TA45CR (0xFFFF_F112)

	7	6	5	4	3	2	1	0
Bit Symbol	TA45EN							
Read/Write	R/W							
Reset Value	0	0	0					
Function	TMRA45 operation 0: Disable 1: Enable	Must be written as 00.						

TA45EN: Enables or disables the operation of the TMRA45. If the TMRA45 is disabled, no clock pulses are supplied to the TMRA45 registers other than the TA45CR, so that power consumption in the system can be reduced (only the TA45CR can be read or written). To use the TMRA45, set the TA45EN bit to 1 before configuring other registers of the TMRA45. Once the TMRA45 operates, all settings in its registers are held if it is disabled. The TA45EN bit enables or disables the operation of the TMRA4-TMRA7. (Enable or disable all channels of the TMRA4-TMRA7.)

Note: Bits 5 and 6 of the TA45CR are read as 0.

TMRA4 Register

TA4REG (0xFFFF_F111)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA4REG are read as undefined.

TMRA5 Register

TA5REG (0xFFFF_F110)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA5REG are read as undefined.

TMRA45 Mode Register

TA45MOD (0xFFFF_F117)

	7	6	5	4	3	2	1	0
Bit Symbol	TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLK1	TA4CLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6 - 1 10: 2^7 - 1 11: 2^8 - 1		TMRA5 clock source 00: TA4TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA4 clock source 00: TA4IN input 01: ϕT1 10: ϕT4 11: ϕT16	

TA4CLK[1:0]: Selects the TMRA4 clock source.

TA5CLK[1:0]: Selects the TMRA5 clock source (when the TA45M[1:0] field is set to other than 01). When TA45M[1:0] = 01, the TMRA4 overflow output is always the TMRA5 clock source regardless of the settings in TA5CLK[1:0].

PWM4[1:0]: Selects the period for 8-bit PWM mode. The PWM period will be $(2^n - 1)$ x clock source period.

TA45M[1:0]: Selects the TMRA45 operating mode. When this field is set to 00, the TMRA45 is used as two independent 8-bit timers, TMRA4 and TMRA5.

TMRA5 Flip-Flop Control Register

TA5FFCR (0xFFFF_F116)

	7	6	5	4	3	2	1	0
Bit Symbol					TAFF5C1	TAFF5C0	TAFF5IE	TAFF5IS
Read/Write					R/W			
Reset Value					1	1	0	0
Function					00: Toggles TA5FF. (software toggle) 01: Sets TA5FF to 1. 10: Clears TA5FF to 0. 11: Don't care. This field is always read as 11.		TA5FF toggle enable 0: Disable 1: Enable	TA5FF toggle trigger 0: TMRA4 1: TMRA5

TAFF5IS: Specifies whether Timer Flip-Flop 5 (TA5FF) is toggled by a TMRA4 match detection signal or a TMRA5 match detection signal. This bit is valid only in 8-bit timer mode; it is a don't-care bit in other modes.

Note: Bits 4, 5, 6 and 7 of the TA5FFCR are read as undefined.

TMRAG1 Interrupt Mask Register

TAG1IM (0xFFFF_F115)

	7	6	5	4	3	2	1	0
Bit Symbol					TAIM7	TAIM6	TAIM5	TAIM4
Read/Write					R/W			
Reset Value					0	0	0	0
Function					1: Masks INTTA7.	1: Masks INTTA6.	1: Masks INTTA5.	1: Masks INTTA4.

Note: Bits 4, 5, 6 and 7 of the TAG1IM are read as undefined.

TMRAG1 Status Register

TAG1ST (0xFFFF_F114)

	7	6	5	4	3	2	1	0
Bit Symbol					INTTA7	INTTA6	INTTA5	INTTA4
Read/Write					R			
Reset Value					0	0	0	0
Function					0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

Note 1: Reading the TAG1ST register results in bits 0, 1, 2 and 3 being cleared.
Note 2: Bits 4, 5, 6 and 7 of the TAG1ST are read as undefined.
Note 3: The flag bit corresponding to the interrupt source being masked is set, but the interrupt is not signaled.

TMRA67 Run Register

TA67RUN (0xFFFF_F11B)

	7	6	5	4	3	2	1	0
Bit Symbol	TA6RDE	TA67C1	TA67C0		I2TA67	TA67PRUN	TA7RUN	TA6RUN
Read/Write	R/W				R/W			
Reset Value	0	0	0		0	0	0	0
Function	Double-buffering 0: Disable 1: Enable	Cascade connection 00: 8- or 16-bit mode 01: 24-bit cascade 10: 32-bit cascade 11: Setting prohibited			IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TA6RUN: Runs or stops the TMRA6.

TA7RUN: Runs or stops the TMRA7.

TA67PRUN: Runs or stops the TMRA67 prescaler.

I2TA67: Enables or disables the operation of the TMRA4-TMRA7 in IDLE mode.

TA67C[1:0]: Specifies how the TMRA67 is used in cascade connection. When this field is set to 00, either 8- or 16-bit mode is selected according to the settings of the TA67M[1:0] bits in the TMRA67 Mode register. When this field is set to 01 (24-bit cascade), the TMRA6 is cascaded with the TMRA45. When this field is set to 10 (32-bit cascade), the TMRA4 to TMRA7 are cascaded.

TA6RDE: Enables or disables double-buffering.

Note: Bit 4 of the TA67RUN is read as undefined.

TMRA67 Control Register

TA67CR (0xFFFF_F11A)

	7	6	5	4	3	2	1	0
Bit Symbol	TA67EN							
Read/Write	R/W							
Reset Value	0	0	0					
Function	TMRA67 operation 0: Disable 1: Enable	Must be written as 00.						

TA67EN: Enables or disables the operation of the TMRA67. If the TMRA67 is disabled, no clock pulses are supplied to the TMRA67 registers other than the TA67CR, so that power consumption in the system can be reduced (only the TA67CR can be read or written). To use the TMRA67, set the TA67EN bit to 1 before configuring other registers of the TMRA67. Once the TMRA67 operates, all settings in its registers are held if it is disabled.

Note: Bits 5 and 6 of the TA67CR are read as 0.

TMRA6 Register

TA6REG (0xFFFF_F119)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA6REG are read as undefined.

TMRA7 Register

TA7REG (0xFFFF_F118)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA7REG are read as undefined.

TMRA67 Mode Register

TA67MOD (0xFFFF_F11F)

	7	6	5	4	3	2	1	0
Bit Symbol	TA67M1	TA67M0	PWM61	PWM60	TA7CLK1	TA7CLK0	TA6CLK1	TA6CLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6 - 1 10: 2^7 - 1 11: 2^8 - 1		TMRA7 clock source 00: TA6TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA6 clock source 00: TA6IN input 01: ϕT1 10: ϕT4 11: ϕT16	

TA6CLK[1:0]: Selects the TMRA6 clock source.

TA7CLK[1:0]: Selects the TMRA7 clock source (when the TA67M[1:0] field is set to other than 01). When TA67M[1:0] = 01, the TMRA6 overflow output is always the TMRA7 clock source regardless of the settings in TA7CLK[1:0].

PWM6[1:0]: Selects the period for 8-bit PWM mode. The PWM period will be $(2^n - 1)$ x clock source period.

TA67M[1:0]: Selects the TMRA67 operating mode. When this field is set to 00, the TMRA67 is used as two independent 8-bit timers, TMRA6 and TMRA7.

TMRA7 Flip-Flop Control Register

TA7FFCR (0xFFFF_F11E)

	7	6	5	4	3	2	1	0
Bit Symbol					TAFF7C1	TAFF7C0	TAFF7IE	TAFF7IS
Read/Write					R/W			
Reset Value					1	1	0	0
Function					00: Toggles TA7FF. (software toggle) 01: Sets TA7FF to 1. 10: Clears TA7FF to 0. 11: Don't care. This field is always read as 11.		TA7FF toggle enable 0: Disable 1: Enable	TA7FF toggle trigger 0: TMRA6 1: TMRA7

TAFF7IS: Specifies whether Timer Flip-Flop 7 (TA7FF) is toggled by a TMRA6 match detection signal or a TMRA7 match detection signal. This bit is valid only in 8-bit timer mode; it is a don't-care bit in other modes.

Note: Bits 4, 5, 6 and 7 of the TA7FFCR are read as undefined.

TMRA89 Run Register

TA89RUN (0xFFFF_F123)

	7	6	5	4	3	2	1	0
Bit Symbol	TA8RDE	TA89C1	TA89C0		I2TA89	TA89PRUN	TA9RUN	TA8RUN
Read/Write	R/W				R/W			
Reset Value	0	0	0		0	0	0	0
Function	Double-buffering 0: Disable 1: Enable	Cascade connection 00: 8- or 16-bit mode 01: Setting prohibited 10: Setting prohibited 11: First stage of cascade			IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TA8RUN: Runs or stops the TMRA8.

TA9RUN: Runs or stops the TMRA9.

TA89PRUN: Runs or stops the TMRA89 prescaler.

I2TA89: Enables or disables the operation of the TMRA8-TMRAB in IDLE mode.

TA89C[1:0]: Specifies how the TMRA89 is used in cascade connection. When this field is set to 00, either 8- or 16-bit mode is selected according to the settings of the TA89M[1:0] bits in the TMRA89 Mode register. When this field is set to 11 (first stage of cascade), the TMRA89 is combined with the TMRAAB to form a 24- or 32-bit timer.

TA8RDE: Enables or disables double-buffering.

Note: Bit 4 of the TA89RUN is read as undefined.

TMRA89 Control Register

TA89CR (0xFFFF_F122)

	7	6	5	4	3	2	1	0
Bit Symbol	TA89EN							
Read/Write	R/W							
Reset Value	0	0	0					
Function	TMRA89 operation 0: Disable 1: Enable	Must be written as 00.						

TA89EN: Enables or disables the operation of the TMRA89. If the TMRA89 is disabled, no clock pulses are supplied to the TMRA89 registers other than the TA89CR, so that power consumption in the system can be reduced (only the TA89CR can be read or written). To use the TMRA89, set the TA89EN bit to 1 before configuring other registers of the TMRA89. Once the TMRA89 operates, all settings in its registers are held if it is disabled.

The TA89EN bit enables or disables the operation of the TMRA8-TMRAB. (Enable or disable all channels of the TMRA8-TMRAB.)

Note: Bits 5 and 6 of the TA89CR are read as 0.

TMRA8 Register

TA8REG (0xFFFF_F121)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TA8REG are read as undefined.

TMRA9 Register

TA9REG (0xFFFF_F120)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Note: Bits 7-0 of the TA9REG are read as undefined.

TMRA89 Mode Register

TA89MOD (0xFFFF_F127)

	7	6	5	4	3	2	1	0
Bit Symbol	TA89M1	TA89M0	PWM81	PWM80	TA9CLK1	TA9CLK0	TA8CLK1	TA8CLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6 - 1 10: 2^7 - 1 11: 2^8 - 1		TMRA9 clock source 00: TA8TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA8 clock source 00: TA8IN input 01: ϕT1 10: ϕT4 11: ϕT16	

TA8CLK[1:0]: Selects the TMRA8 clock source.

TA9CLK[1:0]: Selects the TMRA9 clock source (when the TA89M[1:0] field is set to other than 01). When TA89M[1:0] = 01, the TMRA8 overflow output is always the TMRA9 clock source regardless of the settings in TA9CLK[1:0].

PWM8[1:0]: Selects the period for 8-bit PWM mode. The PWM period will be $(2^n - 1)$ x clock source period.

TA89M[1:0]: Selects the TMRA89 operating mode. When this field is set to 00, the TMRA89 is used as two independent 8-bit timers, TMRA8 and TMRA9.

TMRA9 Flip-Flop Control Register

TA9FFCR (0xFFFF_F126)

	7	6	5	4	3	2	1	0
Bit Symbol					TAFF9C1	TAFF9C0	TAFF9IE	TAFF9IS
Read/Write					R/W			
Reset Value					1	1	0	0
Function					00: Toggles TA9FF. (software toggle) 01: Sets TA9FF to 1. 10: Clears TA9FF to 0. 11: Don't care. This field is always read as 11.		TA9FF toggle enable 0: Disable 1: Enable	TA9FF toggle trigger 0: TMRA8 1: TMRA9

TAFF9IS: Specifies whether Timer Flip-Flop 9 (TA9FF) is toggled by a TMRA8 match detection signal or a TMRA9 match detection signal. This bit is valid only in 8-bit timer mode; it is a don't-care bit in other modes.

Note: Bits 4, 5, 6 and 7 of the TA9FFCR are read as undefined.

TMRAG2 Interrupt Mask Register

TAG2IM (0xFFFF_F125)

	7	6	5	4	3	2	1	0
Bit Symbol					TAIMB	TAIMA	TAIM9	TAIM8
Read/Write					R/W			
Reset Value					0	0	0	0
Function					1: Masks INTTAB.	1: Masks INTTAA.	1: Masks INTTA9.	1: Masks INTTA8.

Note: Bits 4, 5, 6 and 7 of the TAG2IM are read as undefined.

TMRAG2 Status Register

TAG2ST (0xFFFF_F124)

	7	6	5	4	3	2	1	0
Bit Symbol					INTTAB	INTTAA	INTTA9	INTTA8
Read/Write					R			
Reset Value					0	0	0	0
Function					0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

Note 1: Reading the TAG2ST register results in bits 0, 1, 2 and 3 being cleared.
Note 2: Bits 4, 5, 6 and 7 of the TAG2ST are read as undefined.
Note 3: The flag bit corresponding to the interrupt source being masked is set, but the interrupt is not signaled.

TMRAAB Run Register

TAABRUN (0xFFFF_F12B)

	7	6	5	4	3	2	1	0
Bit Symbol	TAARDE	TAABC1	TAABC0		I2TAAB	TAABPRUN	TABRUN	TAARUN
Read/Write	R/W				R/W			
Reset Value	0	0	0		0	0	0	0
Function	Double-buffering 0: Disable 1: Enable	Cascade connection 00: 8- or 16-bit mode 01: 24-bit cascade 10: 32-bit cascade 11: Setting prohibited			IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TAARUN: Runs or stops the TMRAA.

TABRUN: Runs or stops the TMRAB.

TAABPRUN: Runs or stops the TMRAAB prescaler.

I2TAAB: Enables or disables the operation of the TMRA8-TMRAB in IDLE mode.

TAABC[1:0]: Specifies how the TMRAAB is used in cascade connection. When this field is set to 00, either 8- or 16-bit mode is selected according to the settings of the TAABM[1:0] bits in the TMRAAB Mode register. When this field is set to 01 (24-bit cascade), the TMRAA is cascaded with the TMRA89. When this field is set to 10 (32-bit cascade), the TMRA8 to TMRAB are cascaded.

TAARDE: Enables or disables double-buffering.

Note: Bit 4 of the TAABRUN is read as undefined.

TMRAAB Control Register

TAABCR (0xFFFF_F12A)

	7	6	5	4	3	2	1	0
Bit Symbol	TAABEN							
Read/Write	R/W							
Reset Value	0	0	0					
Function	TMRAAB operation 0: Disable 1: Enable	Must be written as 00.						

TAABEN: Enables or disables the operation of the TMRAAB. If the TMRAAB is disabled, no clock pulses are supplied to the TMRAAB registers other than the TAABCR, so that power consumption in the system can be reduced (only the TAABCR can be read or written). To use the TMRAAB, set the TAABEN bit to 1 before configuring other registers of the TMRAAB. Once the TMRAAB operates, all settings in its registers are held if it is disabled.

Note: Bits 5 and 6 of the TAABCR are read as 0.

TMRAA Register

TAAREG (0xFFFF_F129)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TAAREG are read as undefined.

TMRAB Register

TABREG (0xFFFF_F128)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Function	Timer register							

Note: Bits 7-0 of the TABREG are read as undefined.

TMRAAB Mode Register

TAABMOD (0xFFFF_F12F)

	7	6	5	4	3	2	1	0
Bit Symbol	TAABM1	TAABM0	PWMA1	PWMA0	TABCLK1	TABCLK0	TAACLK1	TAACLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6 - 1 10: 2^7 - 1 11: 2^8 - 1		TMRAB clock source 00: TAATRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRAA clock source 00: TAAIN input 01: ϕT1 10: ϕT4 11: ϕT16	

TAACLK[1:0]: Selects the TMRAA clock source.

TABCLK[1:0]: Selects the TMRAB clock source (when the TAABM[1:0] field is set to other than 01). When TAABM[1:0] = 01, the TMRAA overflow output is always the TMRAB clock source regardless of the settings in TABCLK[1:0].

PWMA[1:0]: Selects the period for 8-bit PWM mode. The PWM period will be $(2^n - 1)$ x clock source period.

TAABM[1:0]: Selects the TMRAAB operating mode. When this field is set to 00, the TMRAAB is used as two independent 8-bit timers, TMRAA and TMRAB.

TMRAB Flip-Flop Control Register

TABFFCR (0xFFFF_F12E)

	7	6	5	4	3	2	1	0
Bit Symbol					TAFFBC1	TAFFBC0	TAFFBIE	TAFFBIS
Read/Write					R/W			
Reset Value					1	1	0	0
Function					00: Toggles TABFF. (software toggle) 01: Sets TABFF to 1. 10: Clears TABFF to 0. 11: Don't care. This field is always read as 11.		TABFF toggle enable 0: Disable 1: Enable	TA3FF toggle trigger 0: TMRAA 1: TMRAB

TAFFBIS: Specifies whether Timer Flip-Flop B (TABFF) is toggled by a TMRAA match detection signal or a TMRAB match detection signal. This bit is valid only in 8-bit timer mode; it is a don't-care bit in other modes.

Note: Bits 4, 5, 6 and 7 of the TABFFCR are read as undefined.

11.4 Operating Modes

11.4.1 8-Bit Interval Timer Mode

The TMRA0 and the TMRA1 can be independently programmed as 8-bit interval timers. Programming these timers should only be attempted when the timers are not running.

(1) Generating periodic interrupts

In the following example, the TMRA1 is used to accomplish periodic interrupt generation. First, stop the TMRA1 (if it is running). Then, set the operating mode, clock source and interrupt interval in the TA01MOD and TA1REG registers. Then, enable the INTTA1 interrupt and start the TMRA1.

Example: Generating the INTTA1 interrupt at a 20-µs interval (fc = 40.5 MHz)

Clocking conditions:
- System clock: High-speed (fc)
- Prescaler clock: fperiph/4 (fperiph = fsys)

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	0	0	X	–	–	0	–	Stops and clears the TMRA1.
TA01MOD	← 0	0	X	X	1	0	X	X	Selects 8-Bit Interval Timer mode and ϕT1 as the clock source (which provides a 0.2-µs resolution @fc = 40.5 MHz).
TA1REG	← 0	1	1	0	0	1	0	0	Sets the time constant value in the TA1REG (20 µs ÷ ϕT1 = 100 (64H)).
IMC6LH	← X	X	1	1	0	1	0	1	Enables INTTAGA0 and sets the interrupt level to 5. INTTA1 must always be programmed to be rising-edge triggered.
TA01RUN	← –	–	–	X	–	1	1	–	Starts the TMRA1.

X = Don't care, - = No change

Refer to Table 11.2 when selecting a timer clock source.

Note: The clock inputs to the TMRA0 and the TMRA1 can be one of the following:
TMRA0: TA0IN input, ϕT1, ϕT4 or ϕT16
TMRA1: Match-detect signal from the TMRA0, ϕT1, ϕT16 or ϕT256

(2) Generating a square wave with a 50% duty cycle

The 8-Bit Interval Timer mode can be used to generate square-wave output. This is accomplished by toggling the timer flip-flop (TA1FF) periodically. The TA1FF state can be driven out to the TA1OUT pin. Both the TMRA0 and the TMRA1 can be used as square-wave generators. The following shows an example using the TMRA1.

Example: Generating square-wave output with a 1.2-µs period on the TA1OUT pin (fc = 40.5 MHz)

Clocking conditions:
- System clock: High-speed (fc)
- High-speed clock gear: x1 (fc)
- Prescaler clock: fperiph/4 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
TA01RUN	←	–	0	0	X	–	–	0	–	Stops and clears the TMRA1.
TA01MOD	←	0	0	X	X	0	1	–	–	Selects 8-Bit Interval Timer mode and φT1 as the clock source (which provides a 0.2-µs resolution @fc = 40.5 MHz).
TA1REG	←	0	0	0	0	0	0	1	1	Sets the time constant value in the TA1REG (1.2 µs ÷ φT1 ÷ 2 = 3).
TA1FFCR	←	X	X	X	X	1	0	1	1	Clears the TA1FF to 0 and selects the TMRA1 match-detect output as a toggle-trigger signal.
PACR	←	–	–	–	–	–	–	1	–	Configures PA1 as the TA1OUT output pin.
PAFC	←	–	–	–	–	–	–	1	–	
TA01RUN	←	–	–	–	X	–	1	1	–	Starts the TMRA1.

X = Don't care, – = No change

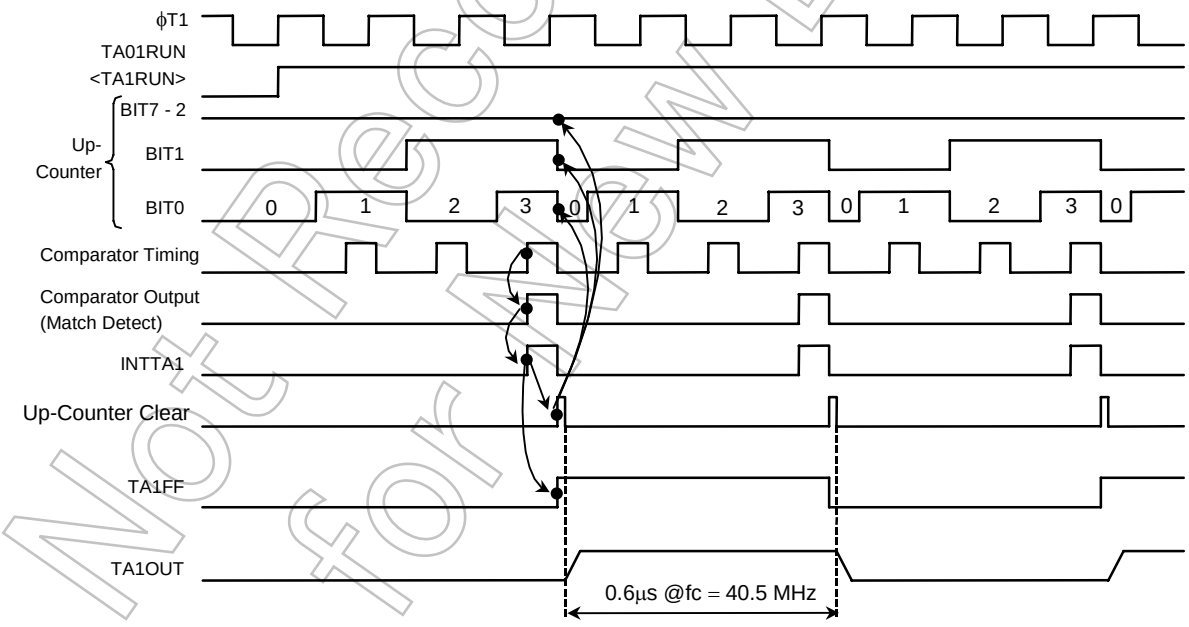


Figure 11.3 Square-Wave Generation (50% Duty Cycle)

(3) Using the TMRA0 match-detect output as a trigger for the TMRA1

Set the TMRA01 in 8-Bit Interval Timer mode. Select the TMRA0 comparator match-detect output as the clock source for the TMRA1.

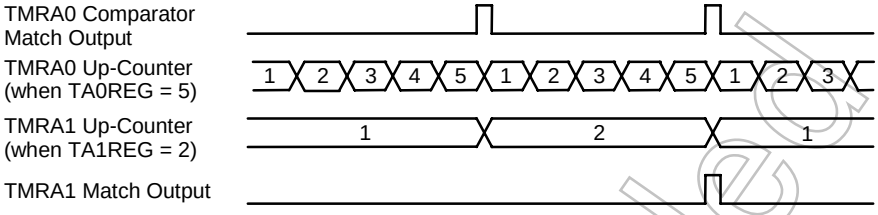


Figure 11.4 Using the TMRA0 Match-Detect Output as a Trigger for the TMRA1

11.4.2 16/24/32-Bit Interval Timer Mode

(1) 16-Bit Interval Timer mode

The TMRA0 and the TMRA1 are cascadable to form a 16-bit interval timer. The TMRA01 is put in 16-Bit Interval Timer mode by programming the TA01M[1:0] field in the TA01MOD register to 01.

In 16-Bit Interval Timer mode, the TMRA1 is clocked by the counter overflow output from the TMRA0. In this mode, the TA1CLK[1:0] bits in the TA01MOD register are Don't-cares. The clock input to the TMRA0 can be selected as shown in Table 11.4.

Write the lower eight bits of a time constant value to the TA0REG and the upper eight bits to the TA1REG. Programming these registers should only be attempted when the timers are not running.

Example: Generating the INTTA1 interrupt at a 0.1-second interval (fc = 40.5 MHz)

Clocking conditions:	System clock:	High-speed (fc)
	High-speed clock gear:	x1 (fc)
	Prescaler clock:	fperiph/4 (fperiph = fsys)

Under the above conditions, ϕT16 has a period of 3.16 μs @ 40.5 MHz. When ϕT16 is used as the TMRA0 clock source, the required time constant value is calculated as follows:

0.1 s ÷ 3.16 μs = 31646 = 7B9EH

Thus, the TA1REG is to be set to 7BH and the TA0REG to 9EH.

Every time the up-counter UC0 reaches the value in the TA0REG, the TMRA0 comparator generates a match-detect output, but the UC0 continues counting up. A match between the UC0 and the TA0REG does not cause an INTTA0 interrupt.

Every time the up-counter UC1 reaches the value in the TA1REG, the TMRA1 comparator generates a match-detect output. When the TMRA0 and TMRA1 match-detect outputs are asserted simultaneously, both the up-counters (UC0 and UC1) are reset to 00H and an interrupt is generated on INTTA1. Also, if so enabled, the timer flip-flop (TA1FF) is toggled.

Example: TA1REG = 04H and TA0REG = 80H

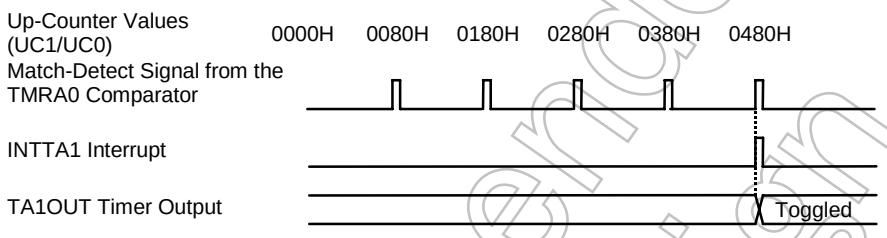


Figure 11.5 Timer Output in 16-Bit Interval Timer Mode

(2) 24-Bit Interval Timer mode

The pair of the TMRA0 and TMRA1 can further be cascaded with the TMRA2 to form a 24-bit interval timer.

In 24-Bit Interval Timer mode, the TMRA1 is clocked by the counter overflow output from the TMRA0. In this mode, the TA1CLK[1:0] bits in the TA01MOD register are Don't-cares. The TMRA2 is clocked by the counter overflow output from the TMRA1. The clock input to the TMRA0 can be selected as shown in Table 11.4.

Write the lowest eight bits of a time constant value to the TA0REG, the middle eight bits to the TA1REG and the highest eight bits to the TA2REG. Programming these registers should only be attempted when the timers are not running.

(3) 32-Bit Interval Timer mode

The TMRA0, TMRA1, TMRA2 and TMRA3 are put in 32-Bit Interval Timer mode by programming the TA01M[1:0] field in the TA01MOD register to 01 and the TA32M0 bit in the TA32MOD register to 1.

In 32-Bit Interval Timer mode, the TMRA1 is clocked by the counter overflow output from the TMRA0. In this mode, the TA1CLK[1:0] bits in the TA01MOD register are Don't-cares. Likewise, the TMRA3 is clocked by the counter overflow output from the TMRA2 and the TA3CLK[1:0] bits in the TA23MOD register are Don't-cares. The TMRA2 is clocked by the counter overflow output from the TMRA1. The clock input to the TMRA0 can be selected as shown in Table 11.4.

Write the lowest eight bits of a time constant value to the TA0REG, the next eight bits to the TA1REG, the next eight bits to the TA2REG and the highest eight bits to the TA3REG. Programming these registers should only be attempted when the timers are not running.

(4) Cascade combinations

In 16-Bit Interval Timer mode:

Upper	Lower
TMRA3	TMRA2
TMRA7	TMRA6
TMRAB	TMRAA

Upper	Lower
TMRA1	TMRA0
TMRA5	TMRA4
TMRA9	TMRA8

In 24-Bit Interval Timer mode:

Highest		Lowest
TMRA2	TMRA1	TMRA0
TMRA6	TMRA5	TMRA4
TMRAA	TMRA9	TMRA8

Note: In 24-Bit Interval Timer mode, the TMRA3, TMRA7 and TMRAB can operate as an independent 8-bit timer.

In 32-Bit Interval Timer mode:

Highest			Lowest
TMRA3	TMRA2	TMRA1	TMRA0
TMRA7	TMRA6	TMRA5	TMRA4
TMRAB	TMRAA	TMRA9	TMRA8

11.4.3 8-Bit Programmable Pulse Generation (PPG) Mode

The 8-Bit PPG mode can be used to generate a square wave with any frequency and duty cycle, as shown below. The pulse can be high-going and low-going, as determined by the initial setting of the timer flip-flop (TA1FF). This mode is supported by the TMRA0, but not by the TMRA1. The square wave output is driven to the TA1OUT pin (which is multiplexed with PA1).

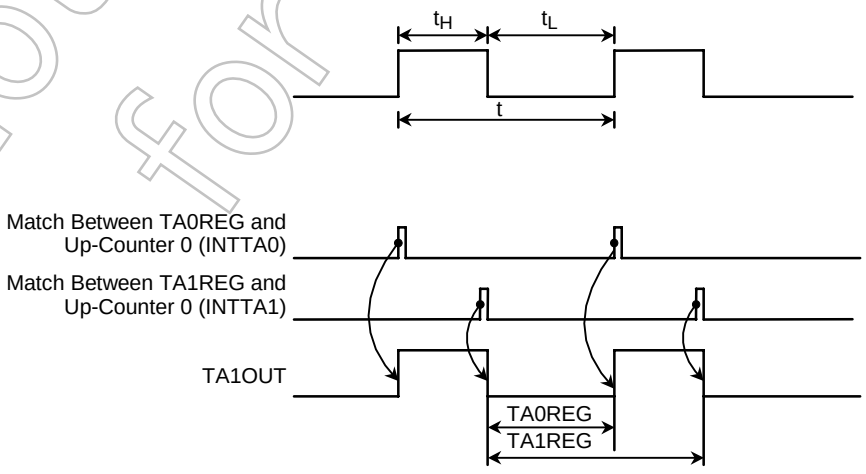


Figure 11.6 8-Bit PPG Output Waveform

In this mode, a square wave is generated by toggling the timer flip-flop (TA1FF). The TA1FF changes state every time a match is detected between the UC0 and the TA0REG and between the UC0 and the TA1REG.

The TA0REG must be set to a value less than the TA1REG value.

In this mode, the TMRA1 up-counter (UC1) cannot be independently used; however, the TMRA1 must be put in a running state by setting the TA1RUN bit in the TA01RUN register to 1.

Figure 11.7 shows a functional diagram of 8-Bit PPG mode.

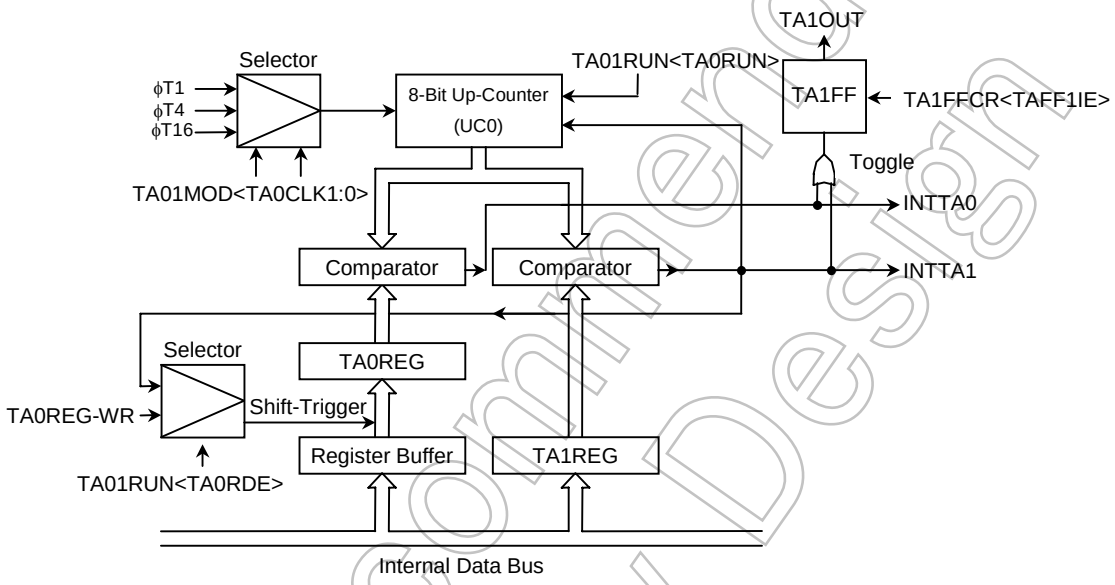


Figure 11.7 Functional Diagram of 8-Bit PPG Mode

In 8-Bit PPG mode, if the double-buffering function is enabled, the TA0REG value can be changed dynamically by writing a new value into the register buffer. Upon a match between the TA1REG and the UC0, the TA0REG latches a new value from the register buffer.

The TA0REG can be loaded with a new value upon every match, thus making it easy to generate a square wave with a variable duty cycle.

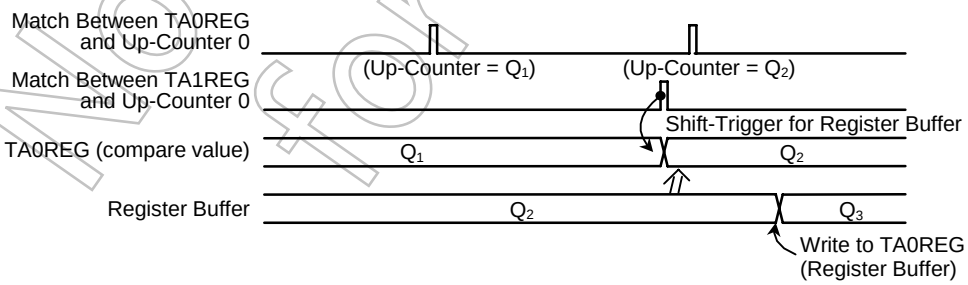
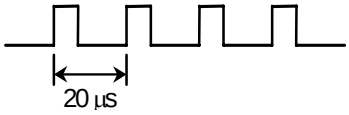


Figure 11.8 Register Buffer Operation

Example: Generating a 50-kHz square wave with a 25% duty cycle (fc = 40.5 MHz)



Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	x1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

The time constant values to be loaded into the TA0REG and TA1REG are determined as follows:

A 50-kHz waveform has a period of 20 μs. Under the above clocking conditions, ϕT1 has a 0.2-μs resolution (@fc = 40.5 MHz). When ϕT1 is used as the timer clock source, the TA1REG should be loaded with:

20 μs ÷ 0.2 μs = 100 (64H)

With a 25% duty cycle, the high pulse width is calculated as 20 μs × 1/4 = 5 μs. Thus, the TA0REG should be loaded with:

5 μs ÷ 0.2 μs = 25 (19H)

		7	6	5	4	3	2	1	0	
TA01RUN	←	0	0	0	X	–	0	0	0	Stops and clears the TMRA0 and the TMRA1.
TA01MOD	←	1	0	X	X	X	X	0	1	Selects 8-Bit PPG mode and ϕT1 as the clock source.
TA0REG	←	0	0	0	1	1	0	0	1	Writes 19H.
TA1REG	←	0	1	1	0	0	1	0	0	Writes 64H.
TA1FFCR	←	X	X	X	X	0	1	1	X	Sets the TA1FF to 1 and enables toggling.
										If these bits are set to 10, a low-going pulse is generated.
PACR	←	–	–	–	–	–	–	1	–	Configures PA1 as the TA1OUT output pin.
PAFC	←	–	–	–	–	–	–	1	–	
TA01RUN	←	1	–	–	X	–	1	1	1	Starts the TMRA0 and the TMRA1.

X = Don't care, – = No change

11.4.4 8-Bit PWM Generation Mode

The TMRA0 can be used as a pulse-width modulated (PWM) signal generator with up to 8 bits of resolution. This mode is supported by the TMRA0, but not by the TMRA1. The PWM signal is driven out on the TA1OUT pin (which is multiplexed with P71).

While the TMRA01 is in this mode, the TMRA1 is usable as an 8-bit interval timer.

The timer flip-flop toggles when the up-counter (UC0) reaches the TA0REG value and when a 2^n-1 counter overflow occurs, where n is programmable to 6, 7 or 8 through the PWM[01:00] field in the TA01MOD register. The UC0 is reset to 00H upon a 2^n-1 overflow.

In 8-Bit PWM Generation mode, the following must be satisfied:

(TA0REG value) < (2^n-1 counter overflow value)

(TA0REG value) ≠ 0

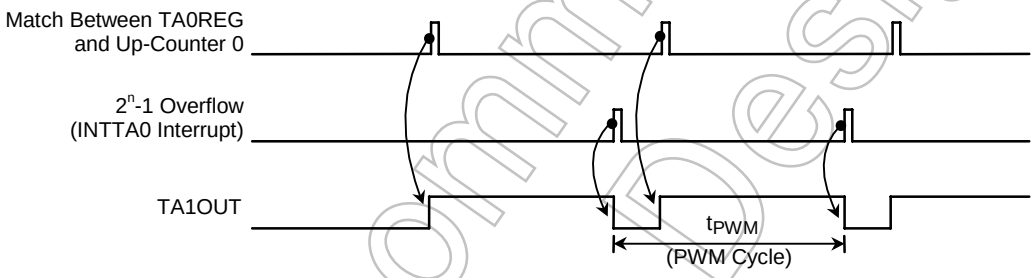


Figure 11.9 8-Bit PWM Signal Generation

Figure 11.10 shows a functional diagram of 8-Bit PWM Generation mode.

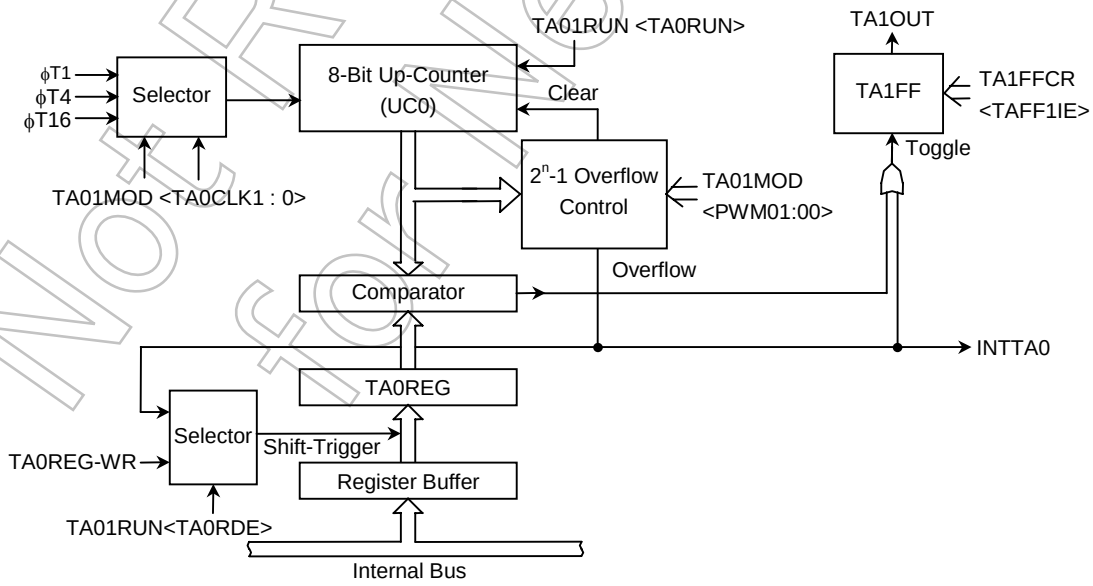


Figure 11.10 Functional Diagram of 8-Bit PWM Generation Mode

In 8-Bit PWM Generation mode, if the double-buffering function is enabled, the TA0REG value (i.e., the duty cycle) can be changed dynamically by writing a new value into the register buffer. Upon a 2^n-1 counter overflow, the TA0REG latches a new value from the register buffer.

The TA0REG can be loaded with a new value upon every counter overflow, thus making it easy to generate a PWM signal with a variable duty cycle.

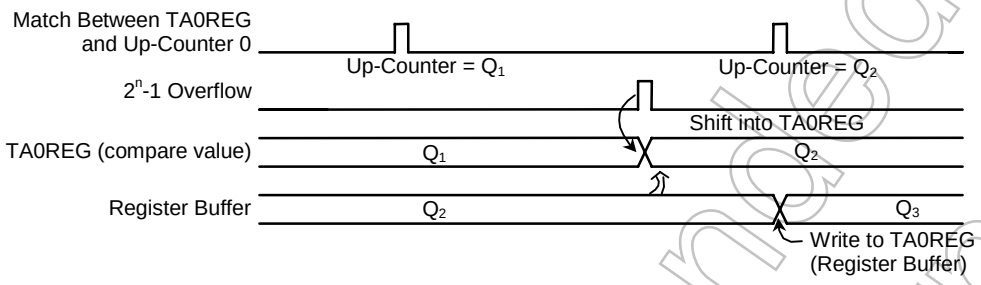
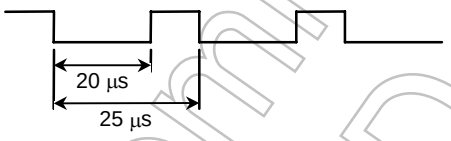


Figure 11.11 Register Buffer Operation

Example: Generating a PWM signal as shown below on the TA1OUT pin (fc = 40.5 MHz)



Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	x1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

Under the above conditions, ϕT1 has a 0.2-μs (0.197-μs) period (@fc = 40.5 MHz).

25 μs ÷ 0.197 μs = 127

which is equal to 2^7 - 1.

20 μs ÷ 0.2 μs = 100 = 64H

Hence, the time constant value to be programmed into the TA0REG is 64H.

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	0	0	X	–	–	–	0	Stops and clears the TMRA0.
TA01MOD	← 1	1	1	0	–	–	0	1	Selects 8-Bit PWM mode (period = 2^7-1) and ϕT1 as the clock source.
TA0REG	← 0	1	1	0	0	1	0	0	Writes 64H.
TA1FFCR	← X	X	X	X	1	0	1	X	Clears the TA1FF to 0 and enables toggling.
PACR	← –	–	–	–	–	–	1	–	Configures PA1 as the TA1OUT output pin.
PAFC	← –	–	–	–	–	–	1	–	
TA01RUN	← 1	–	–	X	–	1	–	1	Starts the TMRA0.

X = Don't care, – = No change

Table 11.3 PWM Period

@fc = 40.5 MHz

Peripheral Clock Select FPSEL	Clock Gear Value GEAR[1:0]	Prescaler Clock Source PRCK[1:0]	PWM Period								
			2^6-1			2^7-1			2^8-1		
			ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16
0(fgear)	00(fc)	00(fperiph/16)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
		01(fperiph/8)	24.9 μs	99.6 μs	398 μs	50.2 μs	201 μs	803 μs	101 μs	403 μs	1.61 ms
		10(fperiph/4)	12.4 μs	49.8 μs	199 μs	25.1 μs	100 μs	401 μs	50.3 μs	201 μs	806 μs
	01(fc/2)	00(fperiph/16)	99.6 μs	398 μs	1.59 ms	201 μs	803 μs	3.21 ms	403 μs	1.61 ms	6.45 ms
		01(fperiph/8)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
		10(fperiph/4)	24.9 μs	99.6 μs	398 μs	50.2 μs	201 μs	803 μs	101 μs	403 μs	1.61 ms
	10(fc/4)	00(fperiph/16)	199 μs	796 μs	3.19 ms	401 μs	1.61 ms	6.42 ms	806 μs	3.22 ms	12.9 ms
		01(fperiph/8)	99.6 μs	398 μs	1.59 ms	201 μs	803 μs	3.21 ms	403 μs	1.61 ms	6.45 ms
		10(fperiph/4)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
	11(fc/8)	00(fperiph/16)	398 μs	1.59 ms	6.37 ms	803 μs	3.21 ms	12.8 ms	1.61 ms	6.45 ms	25.8 ms
		01(fperiph/8)	199 μs	796 μs	3.19 ms	401 μs	1.61 ms	6.42 ms	806 μs	3.22 ms	12.9 ms
		10(fperiph/4)	99.6 μs	398 μs	1.59 ms	201 μs	803 μs	3.21 ms	403 μs	1.61 ms	6.45 ms
1(fc)	00(fc)	00(fperiph/16)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
		01(fperiph/8)	24.9 μs	99.6 μs	398 μs	50.2 μs	201 μs	803 μs	101 μs	403 μs	1.61 ms
		10(fperiph/4)	12.4 μs	49.8 μs	199 μs	25.1 μs	100 μs	401 μs	50.3 μs	201 μs	806 μs
	01(fc/2)	00(fperiph/16)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
		01(fperiph/8)	24.9 μs	99.6 μs	398 μs	50.2 μs	201 μs	803 μs	101 μs	403 μs	1.61 ms
		10(fperiph/4)	12.4 μs	49.8 μs	199 μs	25.1 μs	100 μs	401 μs	50.3 μs	201 μs	806 μs
	10(fc/4)	00(fperiph/16)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
		01(fperiph/8)	24.9 μs	99.6 μs	398 μs	50.2 μs	201 μs	803 μs	101 μs	403 μs	1.61 ms
		10(fperiph/4)	—	49.8 μs	199 μs	—	100 μs	401 μs	—	201 μs	806 μs
	11(fc/8)	00(fperiph/16)	49.8 μs	199 μs	796 μs	100 μs	401 μs	1.61 ms	201 μs	806 μs	3.22 ms
		01(fperiph/8)	—	99.6 μs	398 μs	—	201 μs	803 μs	—	403 μs	1.61 ms
		10(fperiph/4)	—	49.8 μs	199 μs	—	100 μs	401 μs	—	201 μs	806 μs

Note 1: The prescaler's output clock ϕTn must be selected so that ϕTn < fsys/2 is satisfied.
Note 2: The - character means "Setting prohibited."

11.4.5 Operating Mode Summary

Table 11.4 shows the settings for the TMRA01 for each of the operating modes.

Table 11.4 Register Settings for Each Operating Mode

Register	TA01RUN	TA01MOD				TA1FFCR
Field	<TA01C1:0>	<TA01M1:0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	TAFF1IS
Function	Cascade Connection	Interval Timer Mode	PWM Period	UC1 Clock Source	UC0 Clock Source	Timer Flip-Flop Toggle Trigger
8-Bit Timer x 2ch	00	00	—	Match output from UC0 φT1, φT16, φT256 (00, 01, 10, 11)	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	0: UC0 output 1: UC1 output
16-Bit Timer Mode	00	01	—	—	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	—
24-Bit Timer Mode	11	—	—	—	External clock, φT1, φT4, φT16 (00, 01, 10, 11	—
32-Bit Timer Mode	11	—	—	—	External clock, φT1, φT4, φT16 (00, 01, 10, 11	—
8-Bit PPG x 1ch	00	10	—	—	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	—
8-Bit PWM x 1ch 8-Bit Timer (Note 2)	00	11	2^6-1, 2^7-1, 2^8-1 (01, 10, 11)	φT1, φT16 , φT256 (01, 10, 11)	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	PWM output

Note 1: - = Don't care

Note 2: In 8-Bit PWM mode, the UC1 can be used as an 8-bit timer. However, the match-detect output from the UC0 cannot be used as a clock source for the UC1, and the timer output is not available for the UC1.

12. 16-Bit Timer/Event Counters (TMRBs)

The TMP1962 has a 16-bit timer/event counter consisting of four identical channels (TMRB0-TMRB3). Each channel has the following four basic operating modes:

- 16-Bit Interval Timer mode
- 16-Bit Event Counter mode
- 16-Bit Programmable Pulse Generation (PPG) mode
- 2-Phase Pulse Input Counter mode (TMRB2 and TMRB3 only)

Each channel has the capture capability used to latch the value of the counter. The capture capability allows:

- Frequency measurement
- Pulse width measurement
- Time difference measurement

The main components of a TMRBn block are a 16-bit up-counter, two 16-bit timer registers (one of which is double-buffered), two 16-bit capture registers, two comparators, capture control logic, a timer flip-flop and its associated control logic.

A total of thirteen registers provide control over the operating modes and timer flip-flops for each of the TMRB0 to TMRB3. Each channel is independently programmable and functionally equivalent except that the TMRB2 and TMRB3 support the 2-phase pulse count function. In the following sections, any references to the TMRB0 also apply to all the other channels. Table 12.1 gives the pins and registers for the four channels.

Table 12.1 Pins and Registers for the TMRB0-TMRB3

Specifications \ Channel		TMRB0	TMRB1	TMRB2	TMRB3
External Pins	External clock/capture trigger inputs	TB0IN0 (shared with PL4) TB0IN1 (shared with PL5)	TB1IN0 (shared with PL6) TB1IN1 (shared with PL7)	TB2IN0 (shared with PB2) TB2IN1 (shared with PB3)	TB3IN0 (shared with PB5) TB3IN1 (shared with PB6)
	Capture trigger timer	TA3OUT	TA3OUT	TA3OUT	TA3OUT
	Timer flip-flop output	TB0OUT (shared with PB0)	TB1OUT (shared with PB1)	TB2OUT (shared with PB4)	TB3OUT (shared with PB7)
Registers (Addresses)	Timer Run register	TB0RUN (0xFFFF_F143)	TB1RUN (0xFFFF_F153)	TB2RUN (0xFFFF_F163)	TB3RUN (0xFFFF_F173)
	Timer Control register	TB0CR (0xFFFF_F142)	TB1CR (0xFFFF_F152)	TB2CR (0xFFFF_F162)	TB3CR (0xFFFF_F172)
	Timer Mode register	TB0MOD (0xFFFF_F141)	TB1MOD (0xFFFF_F151)	TB2MOD (0xFFFF_F161)	TB3MOD (0xFFFF_F171)
	Timer Flip-Flop Control register	TB0FFCR (0xFFFF_F140)	TB1FFCR (0xFFFF_F150)	TB2FFCR(0xFFFF_F160)	TB3FFCR (0xFFFF_F170)
	Timer Status register	TB0ST (0xFFFF_F147)	TB1ST (0xFFFF_F157)	TB2ST (0xFFFF_F167)	TB3ST (0xFFFF_F177)
	Timer registers	TB0RG0L (0xFFFF_F14B) TB0RG0H (0xFFFF_F14A) TB0RG1L (0xFFFF_F149) TB0RG1H (0xFFFF_F148)	TB1RG0L (0xFFFF_F15B) TB1RG0H (0xFFFF_F15A) TB1RG1L (0xFFFF_F159) TB1RG1H (0xFFFF_F158)	TB2RG0L (0xFFFF_F16B) TB2RG0H (0xFFFF_F16A) TB2RG1L (0xFFFF_F169) TB2RG1H (0xFFFF_F168)	TB3RG0L (0xFFFF_F17B) TB3RG0H (0xFFFF_F17A) TB3RG1L (0xFFFF_F179) TB3RG1H (0xFFFF_F178)
	Capture registers	TB0CP0L (0xFFFF_F14F) TB0CP0H (0xFFFF_F14E) TB0CP1L (0xFFFF_F14D) TB0CP1H (0xFFFF_F14C)	TB1CP0L (0xFFFF_F15F) TB1CP0H (0xFFFF_F15E) TB1CP1L (0xFFFF_F15D) TB1CP1H (0xFFFF_F15C)	TB2CP0L (0xFFFF_F16F) TB2CP0H (0xFFFF_F16E) TB2CP1L (0xFFFF_F16D) TB2CP1H (0xFFFF_F16C)	TB3CP0L (0xFFFF_F17F) TB3CP0H (0xFFFF_F17E) TB3CP1L (0xFFFF_F17D) TB3CP1H (0xFFFF_F17C)

12.1 Block Diagrams

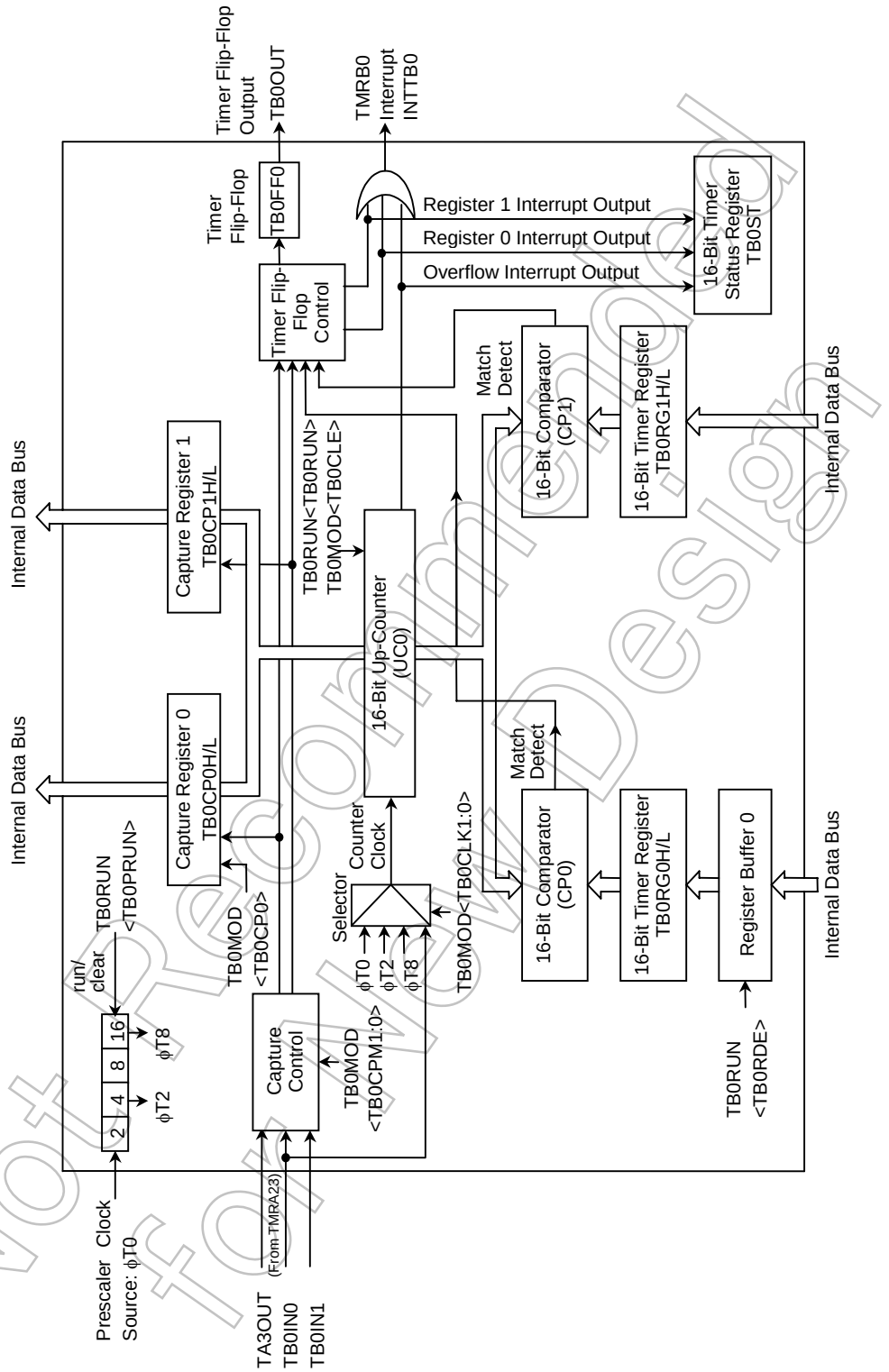


Figure 12.1 TMRB0 Block Diagram (TMRB1 is similar to the above.)

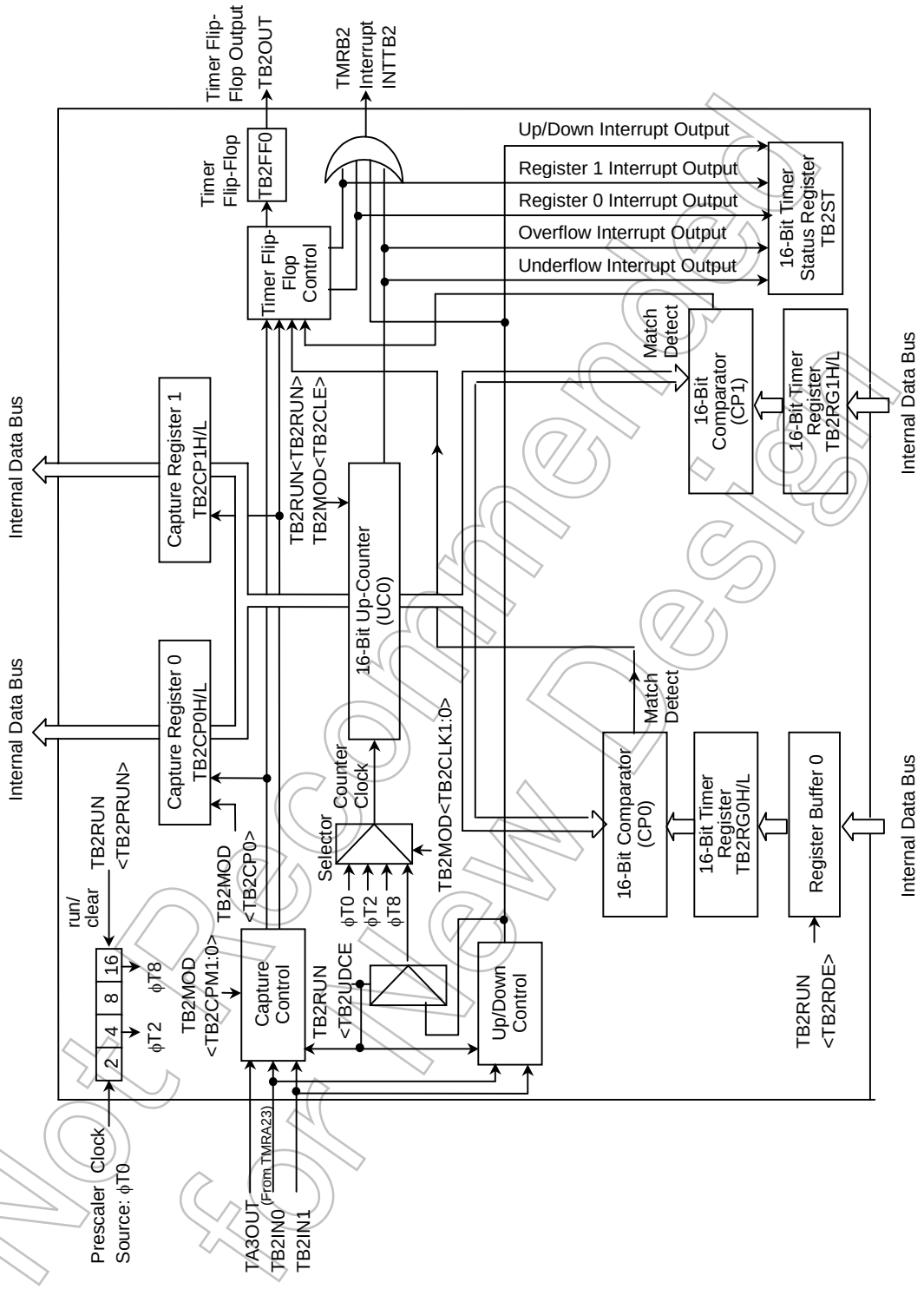


Figure 12.2 TMRB2 Block Diagram (TMRB3 is similar to the above.)

12.2 Timer Components

12.2.1 Prescaler

The TMRB0 has a 5-bit prescaler that slows the rate of a clocking source to the counter. The prescaler clock source (ϕT0) can be selected from fperiph/4, fperiph/8 and fperiph/16 by programming the PRCK[1:0] field of the SYSCR0 located within the CG. fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The TB0RUN bit in the TB0RUN register allows the enabling and disabling of the TMRB0 prescaler. A write of 1 to this bit starts the prescaler. A write of 0 to this bit clears and halts the prescaler. Table 12.2 shows prescaler output clock resolutions.

Table 12.2 Prescaler Output Clock Resolutions

@fc = 40.5 MHz

Peripheral Clock Select FPSEL	Clock Gear Value GEAR[1:0]	Prescaler Clock Source PRCK[1:0]	Prescaler Output Clock Resolution		
			ϕT0	ϕT2	ϕT8
0 (fgear)	00 (fc)	00 (fperiph/16)	fc/2^4(0.4 μμs)	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
		01 (fperiph/8)	fc/2^3(0.2 μs)	fc/2^5(0.8 μs)	fc/2^7 (3.16 μs)
		10 (fperip/4)	fc/2^2(0.1 μs)	fc/2^4(0.4 μs)	fc/2^6 (1.58 μs)
	01 (fc/2)	00 (fperiph/16)	fc/2^5(0.8 μs)	fc/2^7(3.16 μs)	fc/2^9 (12.6 μs)
		01 (fperiph/8)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
		10 (fperip/4)	fc/2^3(0.2 μs)	fc/2^5(0.8 μs)	fc/2^7 (3.16 μs)
	10 (fc/4)	00 (fperiph/16)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{10} (25.3 μs)
		01 (fperiph/8)	fc/2^5(0.8 μs)	fc/2^7(3.16 μs)	fc/2^9 (12.6 μs)
		10 (fperip/4)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
	11 (fc/8)	00 (fperiph/16)	fc/2^7(3.16 μs)	fc/2^9(12.6 μs)	fc/2^{11} (50.6 μs)
		01 (fperiph/8)	fc/2^6(1.58 μs)	fc/2^8(6.32 μs)	fc/2^{10} (25.3 μs)
		10 (fperip/4)	fc/2^5(0.8 μs)	fc/2^7(3.16 μs)	fc/2^9 (12.6 μs)
1 (fc)	00 (fc)	00 (fperiph/16)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
		01 (fperiph/8)	fc/2^3(0.2 μs)	fc/2^5(0.8 μs)	fc/2^7 (3.16 μs)
		10 (fperip/4)	fc/2^2(0.1 μs)	fc/2^4(0.4 μs)	fc/2^6 (1.58 μs)
	01 (fc/2)	00 (fperiph/16)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
		01 (fperiph/8)	fc/2^3(0.2 μs)	fc/2^5(0.8 μs)	fc/2^7 (3.16 μs)
		10 (fperip/4)	—	fc/2^4(0.4 μs)	fc/2^6 (1.58 μs)
	10 (fc/4)	00 (fperiph/16)	fc/2^4(0.4 μs)	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
		01 (fperiph/8)	—	fc/2^5(0.8 μs)	fc/2^7 (3.16 μs)
		10 (fperip/4)	—	fc/2^4(0.4 μs)	fc/2^6 (1.58 μs)
	11 (fc/8)	00 (fperiph/16)	—	fc/2^6(1.58 μs)	fc/2^8 (6.32 μs)
		01 (fperiph/8)	—	fc/2^5(0.8 μs)	fc/2^7 (3.16 μs)
		10 (fperip/4)	—	—	fc/2^6 (1.58 μs)

Note 1: The prescaler's output clock ϕTn must be selected so that the relationship ϕTn < fsys/2 is satisfied.
Note 2: Do not change the clock gear value while the timer is running.
Note 3: The - character means "Setting prohibited."

12.2.2 Up-Counter (UC0)

The TMRB0 contains a 16-bit binary up-counter, which is driven by a clock selected by the TB0CLK[1:0] field in the TB0MOD register.

The clock input to the UC0 is either one of three prescaler outputs (ϕT1, ϕT4, ϕT16) or the external clock applied to the TB0IN0 pin. The TB0RUN bit in the TB0RUN register is used to start the UC0 and to stop and clear the UC0. The UC0 is cleared to 0000H, if so enabled, when it reaches the value in the TB0RG1H/L register. The TB0CLE bit in the TB0MOD register allows the user to enable and disable this clearing.

If it is disabled, the UC0 acts as a free-running counter. An overflow interrupt (INTTB01) is generated upon a counter overflow.

The TMRB2 and TMRB3 support the 2-phase pulse input count function. Setting the TB2UDCE bit in the TR2RUN register to 1 selects 2-Phase Pulse Count mode, causing the TMRB2 to operate as an up/down counter, which is initialized to 0x7FFF. When the counter overflows, it is reloaded with 0x0000. When the counter underflows, it is reloaded with 0xFFFF. In other modes, the TMRB2 and TMRB3 only operate as up-counters.

12.2.3 Timer Registers (TB0RG0H/L and TB0RG1H/L)

Each timer channel has two 16-bit timer registers containing a time constant. When the up-counter reaches the time constant value in each timer register, the associated comparator block generates a match-detect signal.

Each of the timer registers (TB0RG0H/L, TB0RG1H/L) can be written with either a halfword-store instruction or a series of two byte-store instructions. When byte-store instructions are used, the low-order byte must be stored first, followed by the high-order byte.

One of the two timer registers, TB0RG0, is double-buffered. The double-buffering function can be enabled and disabled through the programming of the TB0RDE bit in the TB0RUN: 0=disable, 1=enable. If double-buffering is enabled, the TB0RG0 latches a new time constant value from the register buffer. This takes place when a match is detected between the UC0 and the TB0RG1.

Upon reset, the contents of the TB0RG0 and TB0RG1 are undefined; thus, they must be loaded with valid values before the timer can be used. A reset clears the TB0RUN.TB0RDE bit to 0, disabling the double-buffering function. To use this function, the TB0RUN.TB0RDE bit must be set to 1 after loading the TB0RG0 and TB0RG1 with time constants. When TB0RUN.TB0RDE=1, the next time constant can be written to the register buffer.

The TB0RG0 and the corresponding register buffer are mapped to the same address (0xFFFF_F18A and 0xFFFF_F18B). When TB0RUN.TB0RDE = 0, a time constant value is written to both the TB0RG0 and the register buffer; when TB0RUN.TB0RDE = 1, a time constant value is written only to the register buffer. Therefore, the double-buffering function should be disabled when writing an initial time constant to the timer register.

12.2.4 Capture Registers (TB0CP0H/L and TB0CP1H/L)

The capture registers are 16-bit registers used to latch the value of the up-counter (UC0).

Each of the capture registers can be read with either a halfword-load instruction or a series of two byte-load instructions. When byte-load instructions are used, the low-order byte must be read first, followed by the high-order byte.

12.2.5 Capture Control Logic

The capture control logic controls the capture of an up-counter (UC0) value into the capture registers (TB0CP0 and TB0CP1). The TB0CPM[1:0] field in the TB0MOD register selects a capture trigger input to be sensed by the capture control logic.

Furthermore, a counter value can be captured under software control; a write of 0 to the TB0MOD.TB0CP0 bit causes the current UC0 value to be latched into the TB0CP0. To use the capture capability, the prescaler must be running (i.e., TB0RUN.TB0PRUN=1).

In 2-Phase Pulse Count mode (for the TMRB2 and TMRB3 only), the counter value is captured under software control.

Note 1: Reading the eight low-order bits of a capture register disables the capture capability. Reading the eight high-order bits thereafter re-enables the capture capability.

Note 2: Do not stop the timer after only reading the eight low-order bits of a capture register. If this is done, the capture capability continues to remain in the disabled state even after the timer is restarted.

12.2.6 Comparators (CP0 and CP1)

The TMRB0 contains two 16-bit comparators. The CP0 block compares the output of the up-counter (UC0) with a time constant value in the TB0RG0. The CP1 block compares the output of the UC0 with a time constant value in the TB0RG1. When a match is detected, an interrupt (INTTB0) is generated.

12.2.7 Timer Flip-Flop (TB0FF0)

The timer flip-flop (TB0FF0) is toggled, if so enabled, upon assertion of match-detect signals from the comparators and latch signals from the capture control logic. The toggling of the TB0FF0 can be enabled and disabled through the programming of the TB0C1T1, TB0C0T1, TB0E1T1 and TB0E0T1 bits in the TB0FFCR register.

Upon reset, the TB0FF0 assumes an undefined state. The TB0FF0 can be initialized to 1 or 0 by writing 01 or 10 to the TB0FF0C[1:0] field in the TB0FFCR. A write of 01 to this field sets the TB0FF0; a write of 10 to this field clears the TB0FF0. Additionally, a write of 00 causes the TB0FF0 to be toggled to the opposite value.

The value of the TB0FF0 can be driven onto the TB0OUT pin, which is multiplexed with PB0. The Port B registers (PBCR and PBFC) must be programmed to configure the PB0/TB0OUT pin as TB0OUT.

12.3 Register Description

TMRBn Run Register (n = 0 or 1)

TBnRUN (0xFFFF_F1x3)

	7	6	5	4	3	2	1	0
Bit Symbol	TBnRDE	—			I2TBn	TBnPRUN		TBnRUN
Read/Write	R/W	R/W		R/W	R/W	R/W		R/W
Reset Value	0	0		0	0	0		0
Function	Double-buffering 0: Disable 1: Enable	Must be written as 0.		Must be written as 0.	IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TBnRUN: Runs or stops the TMRBn.

TBnPRUN: Runs or stops the TMRBn prescaler.

I2TBn: Enables or disables the operation of the TMRBn in IDLE mode.

TBnRDE: Enables or disables double-buffering.

Note: Bits 1 and 5 of the TBnRUN are read as 0.

TMRBn Run Register (m = 2 or 3)

TBmRUN (0xFFFF_F1x3)

	7	6	5	4	3	2	1	0
Bit Symbol	TBmRDE	—	UDmCK	TBmUDCE	I2TBm	TBmPRUN		TBmRUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset Value	0	0	0	0	0	0		0
Function	Double-buffering 0: Disable 1: Enable	Must be written as 0.	Must be written as 1.	2-phase counter enable 0: Disable 1: Enable	IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run		

TBmRUN: Runs or stops the TMRBm.

TBmPRUN: Runs or stops the TMRBm prescaler.

I2TBm: Enables or disables the operation of the TMRBm in IDLE mode.

TBmUDCE: Enables or disables the 2-phase pulse input count function.

UDmCK: Selects the sampling clock for the 2-phase pulse input count function.

TBmRDE: Enables or disables double-buffering.

Note 1: Bit 1 of the TBmRUN is read as 0.

Note 2: When bit 4 of the TBmRUN is set to 1, the TMRBm enters 2-Phase Pulse Input Count mode and the counter operates as an up/down counter. When the bit is cleared to 0, the TMRBm enters normal timer mode and the counter operates as an up-counter only.

Figure 12.3 TMRB Registers

TMRBn Control Register (n = 0 to 3)

TBnCR (0xFFFF_F1x2)

	7	6	5	4	3	2	1	0
Bit Symbol	TBnEN	—						
Read/Write	R/W	R/W						
Reset Value	0	0						
Function	TMRBn operation 0: Disable 1: Enable	Must be written as 0.						

TBnEN: Enables or disables the operation of the TMRBn. If the TMRBn is disabled, no clock pulses are supplied to the TMRBn registers other than the TBnCR, so that power consumption in the system can be reduced (only the TBnCR can be read or written). To use the TMRBn, set the TBnEN bit to 1 before configuring other registers of the TMRBn. Once the TMRBn operates, all settings in its registers are held if it is disabled.

Note: Bits 0 to 5 of the TBnCR are read as 0.

TMRBn Mode Register (n = 0 to 3)

TBnMOD (0xFFFF_F1x1)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	TBnCP0	TBnCPM1	TBnCPM0	TBnCLE	TBnCLK1	TBnCLK0
Read/Write			W	R/W				
Reset Value	0	0	1	0	0	0	0	0
Function	Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TBnIN0 ↑ TBnIN1 ↑ 10: TBnIN0 ↑ TBnIN0 ↓ 11: TA3OUT ↑ TA3OUT ↓		Up-counter clear control 0: Disable 1: Enable	Clock source 00: TBnIN0 input 01: φT0 10: φT2 11: φT8	

TBnCLK[1:0]: Selects the clock source for the TMRBn.

TBnCLE: Enables or disables the clearing of the TMRBn up-counter.

0: Disables clearing.

1: Enables the up-counter to be cleared upon a match with TBnRG1.

TBnCPM[1:0]: Specifies the TMRBn capture timing.

00: Disables the capture function.

01: Latches the counter value into Capture Register 0 (TBnCP0) at rising edges of TBnIN0.
Latches the counter value into Capture Register 1 (TBnCP1) at rising edges of TBnIN1.

10: Latches the counter value into Capture Register 0 (TBnCP0) at rising edges of TBnIN0.
Latches the counter value into Capture Register 1 (TBnCP1) at falling edges of TBnIN0.

11: Latches the counter value into Capture Register 0 (TBnCP0) at rising edges of TA3OUT (8-bit timer match output).
Latches the counter value into Capture Register 1 (TBnCP1) at falling edges of TA3OUT.

TMRB0 to 3: TA3OUT

TBnCP0: Writing 0 to this bit latches the counter value into Capture Register 0 (TBnCP0).

Note: Bit 5 of the TBnMOD is read as 1.

Figure 12.4 TMRB Registers

TMRBn Flip-Flop Control Register (n = 0 to 3)

TBnFFCR (0xFFFF_F1x0)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	TBnC1T1	TBnC0T1	TBnE1T1	TBnE0T1	TBnFF0C1	TBnFF0C0
Read/Write	W*		R/W				W*	
Reset Value	1	1	0	0	0	0	1	1
Function	Must be written as 11. * This field is always read as 11.		TBnFF0 toggle trigger 0: Trigger disabled 1: Trigger enabled When the up-counter value is latched into TBnCP1	When the up-counter value is latched into TBnCP0	When the up-counter value reaches TBnRG1	When the up-counter value reaches TBnRG0	TBnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * This field is always read as 11.	

TBnFF0C[1:0]: Controls the timer flip-flop.

00: Toggles TBnFF0. (software toggle)

01: Sets TBnFF0 to 1.

10: Clears TBnFF0 to 0.

11: Don't care.

TBnE[1:0]: Enables or disables the toggling of the timer-flip flop when the up-counter value reaches the value stored in Timer Register 0 or 1 (TBnRG0/1).

TBnC[1:0]: Enables or disables the toggling of the timer-flip flop when the up-counter value is latched into Capture Register 0 or 1 (TBnCP0/1).

Figure 12.5 TMRB Registers

TMRBn Status Register (n = 0 or 1)

TBnST (0xFFFF_F1x7)

	7	6	5	4	3	2	1	0
Bit Symbol						INTTBOFn	INTTBn1	INTTBn0
Read/Write						R		
Reset Value						0	0	0
Function						0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

INTTBn0: Timer Register 0 (TBnRG0) match-detected interrupt

INTTBn1: Timer Register 1 (TBnRG1) match-detected interrupt

INTTBOFn: Up-counter overflow interrupt

Note: When an interrupt occurs, the corresponding flag in the TBnST is set and the INTC is notified of the interrupt. Reading the TBnST register results in all its flags being cleared.

TMRBm Status Register (m = 2 or 3)

(1) When TBmRUN.TBmUDCE = 0: Normal timer mode

TBmST (0xFFFF_F1x7)

	7	6	5	4	3	2	1	0
Bit Symbol						INTTBOFm	INTTBm1	INTTBm0
Read/Write						R		
Reset Value						0	0	0
Function						0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

INTTBm0: Timer Register 0 (TBmRG0) match-detected interrupt

INTTBm1: Timer Register 1 (TBmRG1) match-detected interrupt

INTTBOFm: Up-counter overflow interrupt

Note: When an interrupt occurs, the corresponding flag in the TBmST is set and the INTC is notified of the interrupt. Reading the TBmST register results in all its flags being cleared.

(2) When TBmRUN.TBmUDCE = 1: 2-Phase Pulse Input Count mode

TBmST (0xFFFF_F1x7)

	7	6	5	4	3	2	1	0
Bit Symbol				INTTBUDm	INTTBUDFm	INTTBOUFm		
Read/Write				R				
Reset Value				0	0	0		
Function				Up/down count 0: No count 1: Count (in-cremented or de-cremented)	Underflow 0: No underflow 1: Underflow	Overflow 0: No overflow 1: Overflow		

INTTBOVFm: Up/down counter overflow interrupt

INTTBUDFm: Up/down counter underflow interrupt

INTTBUDm: Up- or down-count interrupt

Note: When an interrupt occurs, the corresponding flag in the TBmST is set and the INTC is notified of the interrupt. Reading the TBmST register results in all its flags being cleared.

Figure 12.6 TMRB Registers

12.4 Operating Modes

12.4.1 16-Bit Interval Timer Mode

In the following example, the TMRB0 is used to accomplish periodic interrupt generation. The interval time is set in Timer Register 1 (TB0RG1), and the INTTB01 interrupt is enabled.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	0	–	0	X	0	Stops the TMRB0.
IMC7LH	←	X	X	1	1	0	1	0	0	Enables INTTB0 and sets its priority level to 4.
TB0FFCR	←	1	1	0	0	0	0	1	1	Disables the timer flip-flop toggle trigger.
TB0MOD	←	0	0	1	0	0	1	*	*	Selects a prescaler output clock as the timer clock source and disables the capture function.
				(** = 01, 10, 11)						
TB0RG1	←	*	*	*	*	*	*	*	*	Sets the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TB0RUN	←	0	0	X	0	–	1	X	1	Starts the TMRB0.

X = Don't care, - = No change

12.4.2 16-Bit Event Counter Mode

This mode is used to count events by interpreting the rising edges of the external counter clock (TB0IN0) as events.

The up-counter counts up on each rising clock edge. The counter value can be latched into a capture register under software control. To determine the number of events (i.e., cycles) counted, the value in the capture register must be read.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	0	–	0	X	0	Stops the TMRB0.
PLCR	←	–	–	–	0	–	–	–	–	Configures the PL4 pin for Input mode.
PLFC	←	–	–	–	1	–	–	–	–	
IMC7LH	←	X	X	1	1	0	1	0	0	Enables INTTB0 (interrupt level = 4).
TB0FFCR	←	1	1	0	0	0	0	1	1	Disables the timer flip-flop toggle trigger.
TB0MOD	←	0	0	1	0	0	1	0	0	Selects the TB0IN0 input as the timer clock source.
TB0RG1	←	*	*	*	*	*	*	*	*	Sets a count value (16 bits).
TB0RUN	←	0	0	X	0	–	1	X	1	Starts the TMRB0.

X = Don't care, - = No change

Even when the timer is used for event counting, the prescaler must be programmed to run (i.e., the TB0RUN.TB0PRUN bit must be set to 1).

12.4.3 16-Bit Programmable Pulse Generation (PPG) Mode

The 16-Bit PPG mode can be used to generate a square wave with any frequency and duty cycle. The pulse can be high-going and low-going, as determined by the initial setting of the timer flip-flop (TB0FF).

A square wave is generated by toggling the timer flip-flop every time the up-counter UC0 reaches the values in each timer register (TB0RG0 and TB0RG1). The square-wave output is driven to the TB0OUT pin. In this mode, the following relationship must be satisfied:

(TB0RG0 value) < (TB0RG1 value)

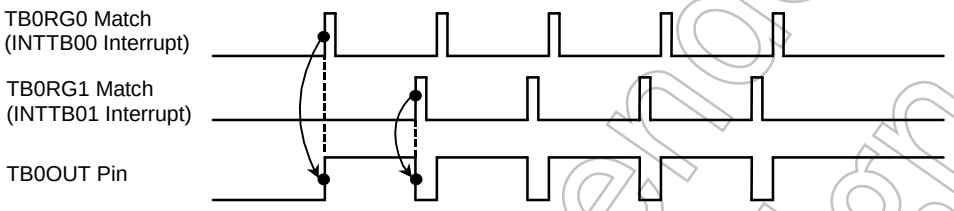


Figure 12.7 PPG Output Waveform

If the double-buffering function is enabled, the TB0RG0 value can be changed dynamically by writing a new value into the register buffer. Upon a match between the TB0RG1 and the UC0, the TB0RG0 latches a new value from the register buffer. The TB0RG0 can be loaded with a new value upon every match, thus making it easy to generate a square wave with virtually any duty cycle.

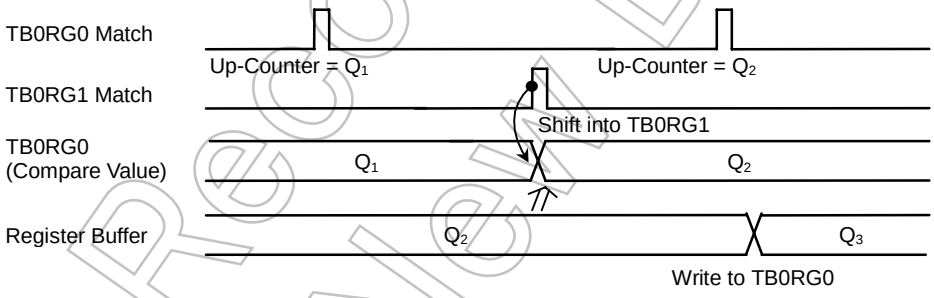


Figure 12.8 Register Buffer Operation

Figure 12.9 shows a functional diagram of 16-Bit PPG mode.

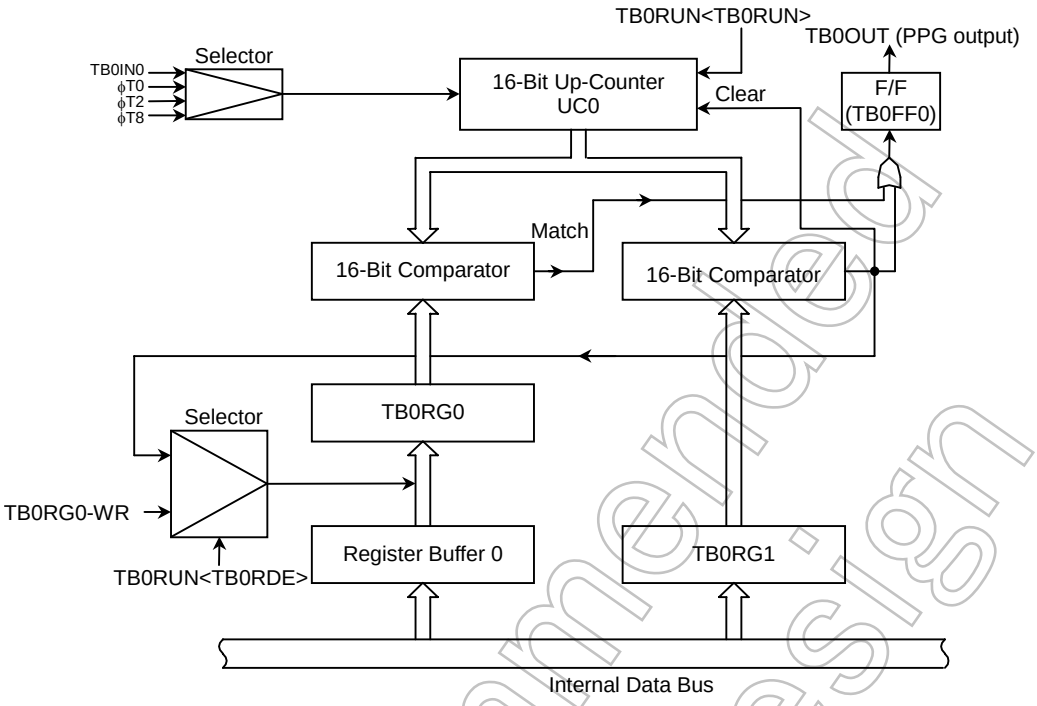


Figure 12.9 Functional Diagram of 16-Bit PPG Mode

The following is an example of running the timer in 16-Bit PPG mode.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	0	–	0	X	0	Disables the TB0RG0 double-buffering and stops the TMRB0.
TB0RG0	←	*	*	*	*	*	*	*	*	Defines the duty cycle (16 bits).
TB0RG1	←	*	*	*	*	*	*	*	*	Defines the cycle period (16 bits).
TB0RUN	←	1	0	X	0	–	0	X	0	Enables the TB0RG0 double-buffering. (The duty cycle and cycle period are changed by the INTTB01 interrupt.)
TB0FFCR	←	X	X	0	0	1	1	1	0	Toggles the TB0FF0 when a match is detected between UC0 and TB0RG0 and between UC0 and TB0RG1. Initially clears the TB0FF0 to 0.
TB0MOD	←	0	0	1	0	0	1	*	*	Selects a prescaler output clock as the timer clock source and disables the capture function. (** = 01, 10, 11)
PBCR	←	–	–	–	–	–	–	–	1	Configures the PB0 pin as TB0OUT.
PBFC	←	–	–	–	–	–	–	–	1	
TB0RUN	←	1	0	X	0	–	1	X	1	Starts the TMRB0.

X = Don't care, - = No change

12.4.4 Timing and Measurement Functions Using the Capture Capability

The capture capability of the TMRBn provides versatile timing and measurement functions, including the following:

(1) One-shot pulse generation using an external trigger pulse

(2) Frequency measurement

(3) Pulse width measurement

(4) Time difference measurement

- One-shot pulse generation using an external trigger pulse

  The TMRBn can be used to produce a one-time pulse as follows.

  The 16-bit up-counter (UC2) is programmed to function as a free-running counter, clocked by one of the prescaler outputs. The TB2IN0 pin is used as an active-high external trigger pulse input for latching the counter value into Capture Register 0 (TB2CP0).

  The Interrupt Controller (INTC) must be programmed to generate an INT5 interrupt upon detection of a rising edge on the TB2IN0 pin. A one-shot pulse has a delay and width controlled by the values stored in the timer registers (TB2RG0 and TB2RG1). Programming the TB2RG0 and TB2RG1 is the responsibility of the INT5 interrupt handler. The TB2RG0 is loaded with the sum of the TB2CP0 value (c) plus the pulse delay (d) − i.e., (c) + (d). The TB2RG1 is loaded with the sum of the TB2RG0 value plus the pulse width (p) − i.e., (c) + (d) + (p).

  Next, the TB2E1T1 and TB2E0T1 bits in the Timer Flip-Flop Control register (TB2FFCR) are set to 11, so that the timer flip-flop (TB2FF0) will toggle when a match is detected between the UC2 and the TB2RG0 and between the UC2 and the TB2RG1. With the TB2FF0 toggled twice, a one-shot pulse is produced. Upon a match between the UC2 and the TB2RG1, the TMRB2 generates the INTTB2 interrupt, which must disable the toggle trigger for the TB2FF0.

  Figure 12.10 depicts one-shot pulse generation, with annotations showing (c), (d) and (p).

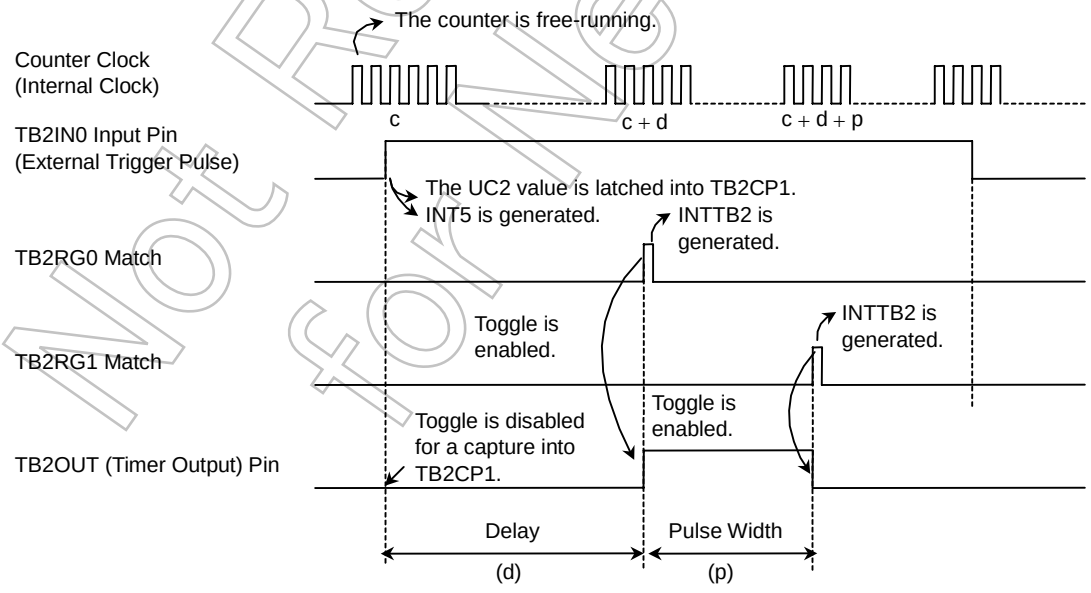


Figure 12.10 One-Shot Pulse Generation (with a Delay)

Example: Generating a one-shot pulse with a width of 2 ms and a delay of 3 ms on assertion of an external trigger pulse on the TB2IN0 pin

Clocking conditions:

System clock: High-speed (fc)
High-speed clock gear: x1 (fc)
Prescaler clock: fperiph/4 (fperiph = fsys)

Settings in the main routine

		7	6	5	4	3	2	1	0	
										Places the counter in Free-Running mode.
TB2MOD	←	X	X	1	0	1	0	0	1	Selects φT0 as the counter clock source.
										Latches UC2 value into TB2CP0 at rising edges of the TB2IN0 input.
TB2FFCR	←	X	X	0	0	0	0	1	0	Clears TB2FF0 to 0.
										Disables the toggle trigger for TB2FF0.
PBCR	←	–	-	–	1	–	–	–	–	Configures the PB4 pin as TB2OUT.
PBFC	←	–	-	–	1	–	–	–	–	
IMC2HL	←	X	X	1	1	0	1	0	0	Enables INT5 and disables INTTB2.
IMCCLH	←	X	X	1	1	0	0	0	0	
TB2RUN	←	–	0	X	0	–	1	X	1	Starts the TMRB2.

Settings in INT5

TB2RG0	←	TB0CP0 + 3ms/φT1								
TB2RG1	←	TB0RG0 + 2ms/φT1								
TB2FFCR	←	X	X	–	–	1	1	–	–	Enables the TB2FF0 toggle trigger for TB2RG0 and TB2RG1 matches.
IMCCLH	←	X	X	1	1	0	1	0	0	Enables INTTB2.

Settings in INTTB2

TB2FFCR	←	X	X	–	–	0	0	–	–	Disables the TB2FF0 toggle trigger for TB2RG0 and TB2RG1 matches.
IMCCLH	←	X	X	1	1	0	0	0	0	Disables INTTB2.

X = Don't care, - = No change

If no delay is necessary, enable the TB2FF0 toggle trigger for a capture of the UC2 value into the TB2CP0. Use the INT5 interrupt to load the TB2RG1 with a sum of the TB2CP0 value (c) plus the pulse width (p) and to enable the TB2FF0 toggle trigger for a match between the UC2 and TB2RG1 values. A match generates the INTTB2 interrupt, which then is to disable the TB2FF0 toggle trigger.

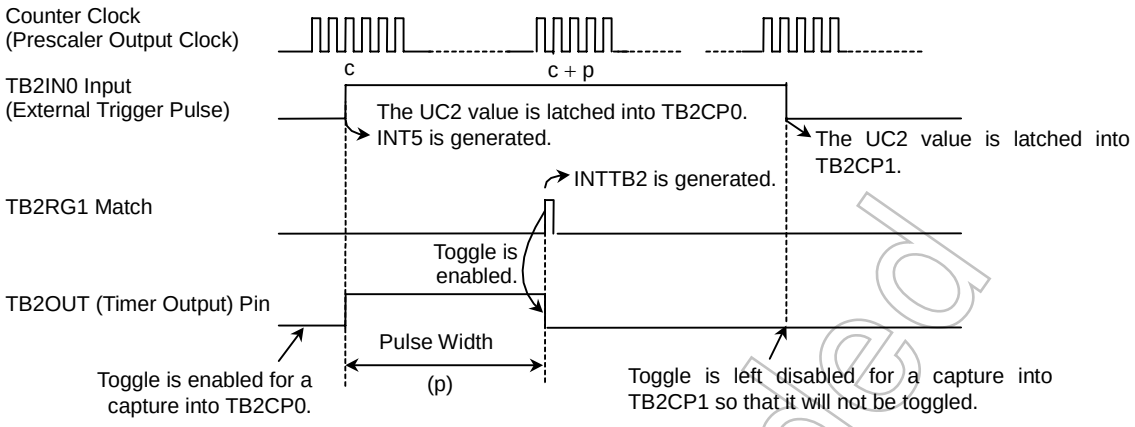


Figure 12.11 One-Shot Pulse Generation (without a Delay)

- Frequency measurement

  The capture function can be used to measure the frequency of an external clock. Frequency measurement requires a 16-bit TMRBn channel running in Event Counter mode and the 8-bit TMRA01. The timer flip-flop (TA1FF) in the TMRA01 is used to define the duration during which a measurement is taken.

  Select the TB0IN0 pin as the clock source for the TMRB0. Set the TB0CPM[1:0] field in the TB0MOD to 11 to select the TA1FF output signal from the TMRA01 as a capture trigger input. This causes the TMRB0 to latch the 16-bit up-counter (UC0) value into Capture Register 0 (TB0CP0) on the low-to-high transition of the TA1FF and into Capture Register 1 (TB0CP1) on the next high-to-low transition of the TA1FF.

  Either the INTTA0 or INTTA1 interrupt generated by the 8-bit timer can be used to make a frequency calculation.

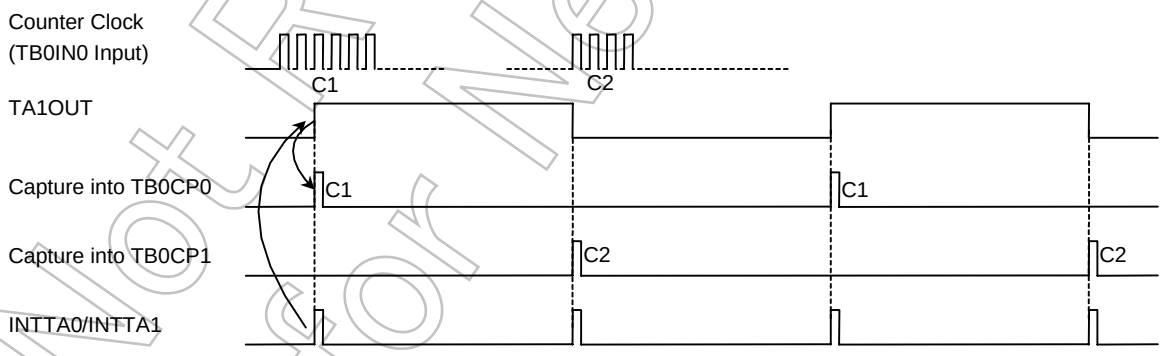


Figure 12.12 Frequency Measurement

For example, if the TA1FF of the 8-bit timer is programmed to be at logic 1 for a period of 0.5 seconds and the difference between the values captured into the TB0CP0 and TB0CP1 is 100, then the TB0IN0 frequency is calculated as 100 ÷ 0.5 s = 200 Hz.

- Pulse width measurement

  The capture function can be used to measure the pulse width of an external clock. The external clock is applied to the TB2IN0 pin. The up-counter (UC2) is programmed to operate as a free-running counter, clocked by one of the prescaler outputs. The capture function is used to latch the UC2 value into Capture Register 0 (TB2CP0) at the clock rising edge and into Capture Register 1 (TB2CP1) at the next clock falling edge. An INT5 interrupt is generated at the falling edge of the TB2IN0 input.

  Multiplying the counter clock period by the difference between the values captured into the TB2CP0 and TB2CP1 gives the high pulse width of the TB2IN0 clock.

  For example, if the prescaler output clock has a period of 0.5 µs and the difference between the TB2CP0 and TB2CP1 is 100, the high pulse width is calculated as 0.5 µs x 100 = 50 µs.

  Measuring a pulse width exceeding the maximum counting time for the UC2, which depends on the clock source, requires software programming.

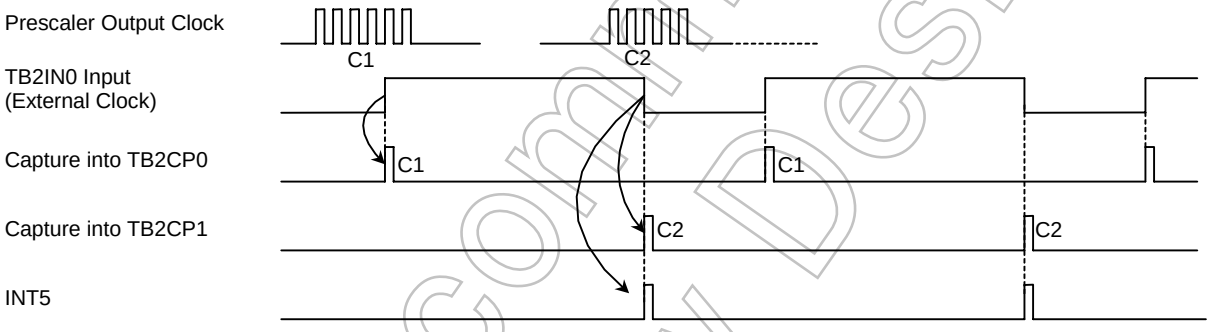


Figure 12.13 Pulse Width Measurement

The low pulse width can be measured by the second INT5 interrupt. This is accomplished by multiplying the counter clock period by the difference between the TB2CP0 value at the first C2 and the TB2CP1 value at the second C1.

- Time difference measurement

  The capture function can be used to measure the time difference between two event occurrences. The 16-bit up-counter (UC2) is programmed to operate as a free-running counter. The UC2 value is latched into Capture Register 0 (TB2CP0) on the rising edge of TB2IN0. An INT5 interrupt is generated at this time.

  Then, the UC2 value is latched into Capture Register 1 (TB2CP1) on the rising edge of TB2IN1. An INT6 interrupt is generated at this time.

  The time difference between the two events that occurred on the TB2IN0 and TB2IN1 pins is calculated by multiplying the counter clock period by the difference between the TB2CP1 and TB2CP0 values.

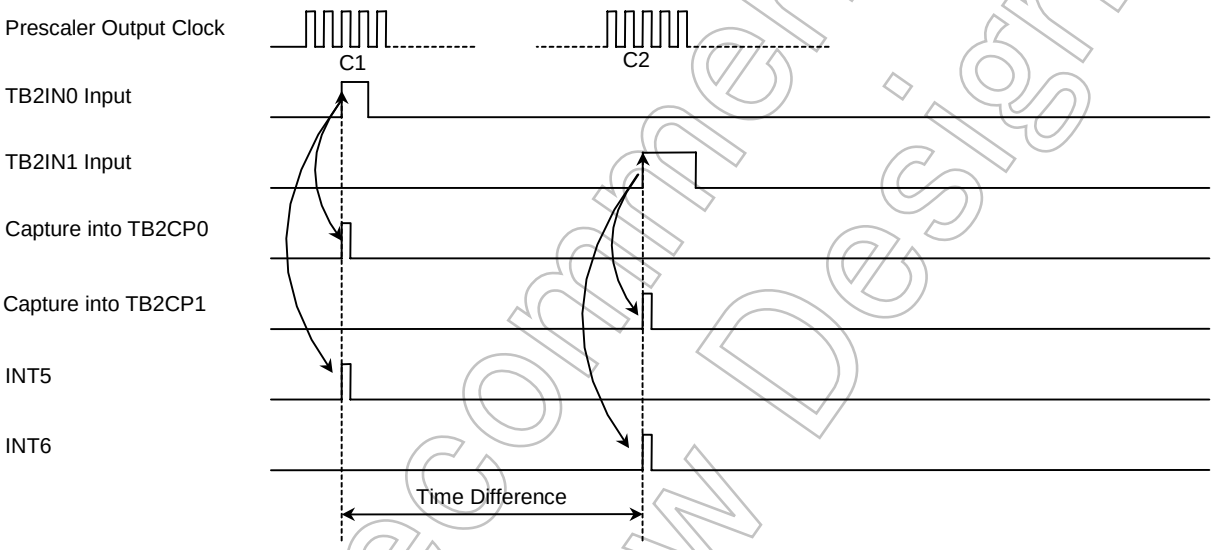


Figure 12.14 Time Difference Measurement

12.4.5 2-Phase Pulse Input Count Mode (TMRB2 and TMRB3)

The TMRB2 and TMRB3 are functionally equivalent. This section only describes the TMRB2.

In 2-Phase Pulse Input Count mode, the counter operates as an up/down counter that increments and decrements according to transitions in the states of 2-phase clocks, input through the TB2IN0 and TB2IN1 pins. An interrupt occurs when the counter increments or decrements, or when it overflows or underflows.

(1) Count operation

- Increment

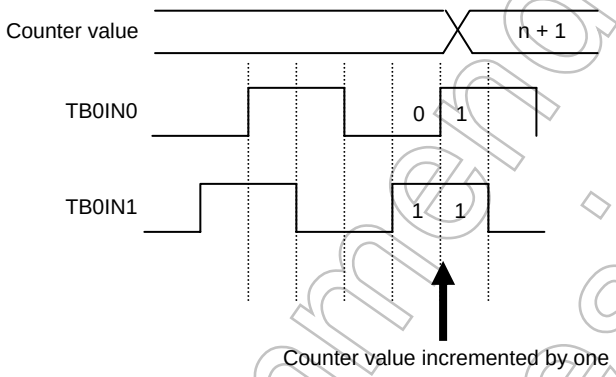


Figure 12.15 When the Counter Increments

- Decrement

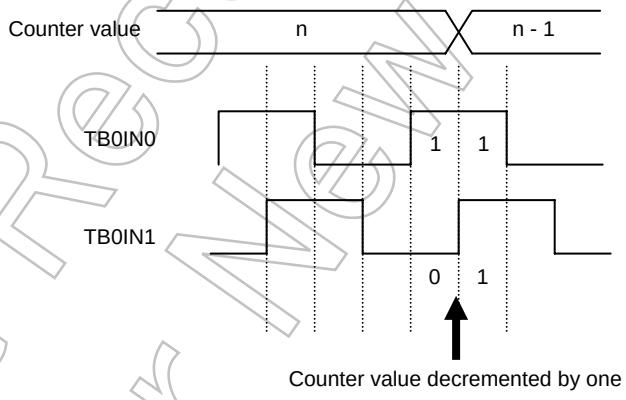


Figure 12.16 When the Counter Decrements

TMRB2 Run Register (TB2RUN)

	7	6	5	4	3	2	1	0
Bit Symbol	TB2RDE		UD2CK	TB2UDCE	I2TB2	TB2PRUN		TB2RUN
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
Reset Value	0		0	0	0	0		0
Function	Double-buffering 0: Disable 1: Enable		Sampling clock 0: Setting prohibited 1: fsys/2	2-phase counter enable 0: Disable 1: Enable	IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run (count up)		

Figure 12.17 2-Phase Pulse Input Count Mode Register

Setting the bit 5 (UD2CK) of the TB2RUN register to 1 specifies the sampling clock frequency.

UD2CK (sampling clock) = 1 : fsys/2 (fsys/8-Hz sampling)

Exiting STOP mode

When the TMP1962 enters STOP mode, the 2-phase counter holds the current status. If the combination of the held status and the status of inputs used for terminating STOP mode satisfies the increment or decrement conditions, the counter increments or decrements when the TMP1962 exits STOP mode. If a fixed status is required after recovery from STOP mode, initialize the 2-phase counter to 0x7FFF once the TMP1962 exits STOP mode, by clearing the TB2RUN.TB2UCDE bit to 0 and then re-setting it to 1.

(2) Operating mode

Whether external signals input through the TB2IN0 and TB2IN1 pins are directed to the regular 16-bit timer (capture input) or the up/down counter depends on register settings.

- In up/down counter mode, only software capture is available. The timer cannot capture data based on external clock timing.
- In up/down counter mode, the comparator is disabled; comparison with the timer register is not performed.
- Input clock sampling is based clock (system clock).

  The maximum input frequency is fsys/16 Hz .

Enabling the up/down counter

Clear the TB2CLK[0:1] bits of the TB2MOD register to 00, thus turning the prescaler off. Then, use bit 4 (TB2UDCE) of the TB2RUN register to specify whether the counter will operate as an up/down counter or a normal up-counter.

TB2UDCE (up/down counter enable) = 0: Normal 16-bit timer operation

= 1: Up/down counter operation

TMRB2 Run Register (TB2RUN)

	7	6	5	4	3	2	1	0
Bit Symbol	TB2RDE		UD2CK	TB2UDCE	I2TB2	TB2PRUN		TB2RUN
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
Reset Value	0		0	0	0	0		0
Function	Double-buffering 0: Disable 1: Enable		Sampling clock 0: Setting prohibited 1: fsys/2	2-phase counter enable 0: Disable 1: Enable	IDLE 0: Off 1: On	Timer run/stop control 0: Stop & clear 1: Run (count up)		

Figure 12.18 Up/Down Counter Enable Bit

(3) Interrupts

- NORMAL mode

  Enable the INTTB2 interrupt in the Interrupt Controller (INTC). An INTTB2 interrupt occurs when the counter increments or decrements. The interrupt service routine can read the TMRB2 status register (TB2ST) to determine whether an overflow or underflow has occurred. If an overflow occurs, the INTTBOUF2 bit of the TB2ST register is set to 1. If an underflow occurs, the INTTBUDF2 bit of the TB2ST register is set to 1. Reading the TB2ST register clears all of its bits. An overflow causes the counter value to be 0x0000 and an underflow causes the counter value to be 0xFFFF. In either case, counting continues.

TB2ST (0xFFFF_F167)

	7	6	5	4	3	2	1	0
Bit Symbol				INTTBUD2	INTTBUDF2	INTTBOUF2		
Read/Write				R				
Reset Value				0	0	0		
Function				Up/down count 0: No count 1: Count (in-cremented or de-cremented)	Underflow 0: No underflow 1: Underflow	Overflow 0: No overflow 1: Overflow		

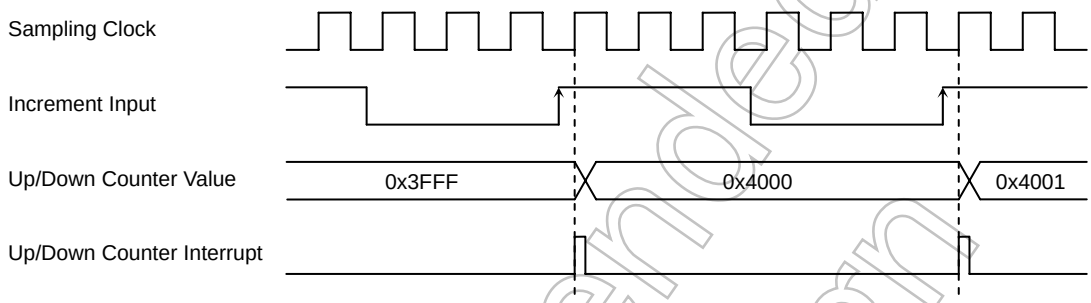
Figure 12.19 TMRB2 Status Register

- STOP mode

  In STOP mode, the 2-phase pulse input counter is stopped. Enable the INT5 or INT6 for STOP wake-up signaling in the Clock Generator (CG). When the counter increments or decrements, an INT5 or INT6 interrupt occurs, causing the TMP1962 to exit STOP mode. Upon recovery from STOP mode, the TMP1962 enters NORMAL or SLOW mode after a specified warm-up time, restarting the counter. If the combination of the held status and the status of inputs used for terminating STOP mode satisfies the increment or decrement conditions, the counter increments or decrements.

(4) Up/down counter

When 2-Phase Pulse Input Count mode is selected (TB2RUN.TB2UCDE = 1), the up-counter operates as an up/down counter and is initialized to 0x7FFF. When the counter overflows, it is reset to 0x0000 and continues counting. When the counter underflows, it is reset to 0xFFFF and continues counting. Once an interrupt occurs, the interrupt service routine can determine the overflow or underflow status by reading the counter value and the status flags in the TB2ST.



Note 1: The increment or decrement input must be high before and after effective input.

Note 2: The counter value must be read in the INTTB2 interrupt service routine.

If it is read in the INT5 or INT6 interrupt routine used for STOP wake-up signaling, the value may vary depending on whether the increment/decrement conditions are satisfied or on the delay between recovery from STOP mode and the start of counting.

13. 32-Bit Input Capture (TMRC)

The TMP1962 contains a 32-bit input capture circuit block (TMRC), which consists of a 1-channel 32-bit time base timer (TBT), eight 32-bit input capture registers (TCCAP0-TCCAP7) and eight 32-bit compare registers (TCCMP0-TCCMP7).

Figure 13.1 shows a block diagram of the TMRC.

13.1 TMRC Block Diagram

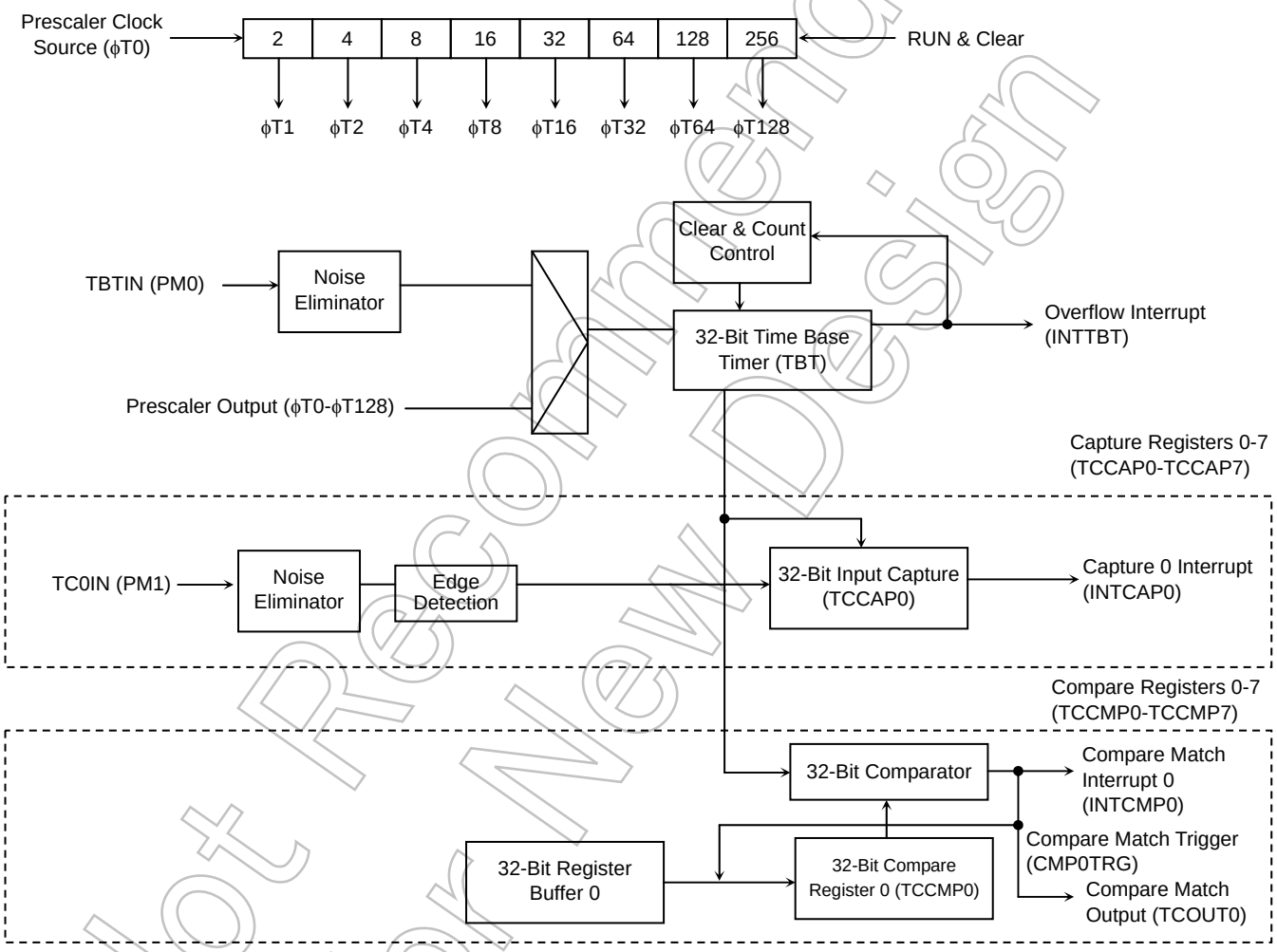


Figure 13.1 TMRC Block Diagram

13.2 Timer Components

13.2.1 Prescaler

The TMRC has an 8-bit prescaler that slows the rate of a clocking source to the timer. The prescaler clock source (ϕT0) can be selected from fperiph/4, fperiph/8 and fperiph/16 by programming the PRCK[1:0] field of the SYSCR0 located within the CG.

fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The TBTPRUN bit in the TBTRUN register allows the enabling and disabling of the prescaler for the TMRC. A write of 1 to this bit starts the prescaler. A write of 0 to this bit clears and halts the prescaler. Table 13.1 shows prescaler output clock resolutions.

Table 13.1 Prescaler Output Clock Resolutions

@fc = 40.5 MHz

Peripheral Clock Source FPSEL	Clock Gear Value GEAR[1:0]	Prescaler Clock Source PRCK[1:0]	Prescaler Output Clock Resolution			
			ϕT1	ϕT2	ϕT4	ϕT8
0(fgear)	00(fc)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
		01(fperiph/8)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)
		10(fperiph/4)	fc/2^3(0.20 μs)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)
	01(fc/2)	00(fperiph/16)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)
		01(fperiph/8)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
		10(fperiph/4)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)
	10(fc/4)	00(fperiph/16)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)
		01(fperiph/8)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)
		10(fperiph/4)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
	11(fc/8)	00(fperiph/16)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
		01(fperiph/8)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)
		10(fperiph/4)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)
1(fc)	00(fc)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
		01(fperiph/8)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)
		10(fperiph/4)	fc/2^3(0.20 μs)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)
	01(fc/2)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
		01(fperiph/8)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)
		10(fperiph/4)	fc/2^3(0.20 μs)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)
	10(fc/4)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
		01(fperiph/8)	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)
		10(fperiph/4)	—	fc/2^4(0.40 μs)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)
	11(fc/8)	00(fperiph/16)	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)
		01(fperiph/8)	—	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)	fc/2^7(3.16 μs)
		10(fperiph/4)	—	—	fc/2^5(0.79 μs)	fc/2^6(1.58 μs)

@fc = 40.5 MHz

Peripheral Clock Source FPSEL	Clock Gear Value GEAR[1:0]	Prescaler Clock Source PRCK[1:0]	Prescaler Output Clock Resolution			
			φT16	φT32	φT64	φT128
0(fgear)	00(fc)	00(fperiph/16)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
		01(fperiph/8)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
		10(fperiph/4)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μ μs)	fc/2^{10}(25.3 μs)
	01(fc/2)	00(fperiph/16)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μ μs)	fc/2^{13}(202 μs)
		01(fperiph/8)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
		10(fperiph/4)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
	10(fc/4)	00(fperiph/16)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)	fc/2^{13}(202 μs)	fc/2^{14}(405 μs)
		01(fperiph/8)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)	fc/2^{13}(202 μs)
		10(fperiph/4)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
	11(fc/8)	00(fperiph/16)	fc/2^{12}(101 μs)	fc/2^{13}(202 μs)	fc/2^{14}(405 μs)	fc/2^{15}(809 μs)
		01(fperiph/8)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)	fc/2^{13}(202 μs)	fc/2^{14}(405 μs)
		10(fperiph/4)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)	fc/2^{13}(202 μs)
1(fc)	00(fc)	00(fperiph/16)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
		01(fperiph/8)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
		10(fperiph/4)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)
	01(fc/2)	00(fperiph/16)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
		01(fperiph/8)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
		10(fperiph/4)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)
	10(fc/4)	00(fperiph/16)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
		01(fperiph/8)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
		10(fperiph/4)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)
	11(fc/8)	00(fperiph/16)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)	fc/2^{12}(101 μs)
		01(fperiph/8)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)	fc/2^{11}(50.6 μs)
		10(fperiph/4)	fc/2^7(3.16 μs)	fc/2^8(6.32 μs)	fc/2^9(12.6 μs)	fc/2^{10}(25.3 μs)

Note 1: The prescaler's output clock φTn must be selected so that the relationship φTn < fsys/2 is satisfied.
Note 2: Do not change the clock gear value while the timer is running.
Note 3: The - character means "Setting prohibited."

13.2.2 Noise Eliminator

The noise eliminator removes noise components from the external clock source (TBTIN) and capture trigger input (TCnIN) for the time base timer (TBT). It can also output the input signals as is, without eliminating noise.

13.2.3 32-Bit Time Base Timer (TBT)

The TMRC contains a 32-bit binary counter, which counts up on the rising edge of a clock selected by the TBTCLK[3:0] field in the TBT Control Register (TBTCR).

The clock input to the TBT is either one of eight prescaler outputs (ϕT1, ϕT2, ϕT4, ϕT8, ϕT16, ϕT32, ϕT64, ϕT128) or the external clock applied to the TBTIN pin.

The TBTRUN bit in the TBTRUN register is used to start the TBT and to stop and clear the TBT. Upon reset, the TBT is cleared and stopped. When started, the TBT acts as a free-running counter. An overflow interrupt (INTTBT) is generated upon a counter overflow and clears the counter to 0, after which the counter restarts counting. INTTBT is grouped with other interrupt sources and controlled in the TCG1ST and TCG1IM registers, in the same way as INTCAPn described in Section 13.2.5.

13.2.4 Edge Detection Circuit

This circuit samples the external capture input (TCnIN) and detects its edges. The CPnEG[1:0] field in the Capture Control Register (CAPnCR) defines the edge detection polarity: rising edge, falling edge, both edges or no capture. Figure 13.2 shows the relationship between the capture input and the output from the edge detection circuit (capture source output).

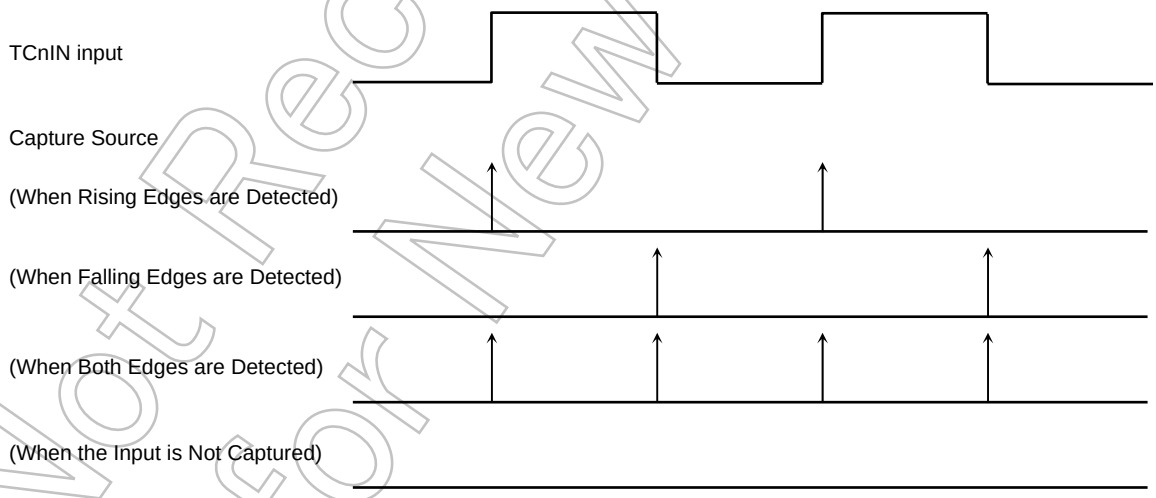


Figure 13.2 Capture Input and Capture Source Output (Output from the Edge Detection Circuit)

13.2.5 32-Bit Capture Registers

The TMRC contains 32-bit registers to which the TBT counter value is captured in response to a capture trigger source. A capture interrupt (INTCAPn) occurs when the counter value is captured. INTCAP0 to INTCAP3 are handled as a group so that any of the four interrupt sources are sent to the INTC as the same interrupt source. The interrupt service routine can determine the actual source by reading the status register (TCG0ST). Unnecessary interrupt requests can be masked by setting corresponding bits in the interrupt mask register (TCG0IM). Any capture trigger does not cause the counter value to be captured while the capture register is being read.

13.2.6 32-Bit Compare Registers

The TMRC contains eight 32-bit registers (TCCMP0-TCCMP7) in which compare values are stored. When the TBT value reaches the value stored in a compare register, the comparator activates the corresponding match-detected signal. The CMPEN[1:0] field in the Compare Control Register (CMPCTL) allows the enabling and disabling of comparison.

TCCMPn can be written with either a word-store instruction, a series of two halfword-store instructions or a series of four byte-store instructions. When halfword-store or byte-store instructions are used, the low-order byte must be stored first, followed by the high-order byte.

Each of the compare registers (TCCMPn) is double-buffered with Register Buffer n. The double-buffering function can be enabled and disabled through the programming of the CMPRDEn bit in the CMPCTL register: 0 = disable, 1 = enable.

If double-buffering is enabled, the TCCMPn latches a new compare value from Register Buffer n. This takes place when a match is detected between the TBT and the TCCMPn.

Upon reset, the contents of the TCCMPn are undefined; thus, they must be loaded with valid values before the timer can be used. A reset clears the CMPCTL.CMPRDEn bit to 0, disabling the double-buffering function. To use this function, the CMPCTL.CMPRDEn bit must be set to 1 after loading the TCCMPn with compare values. When CMPCTL.CMPRDEn = 1, the next compare value can be written to Register Buffer n.

The TCCMPn and the corresponding register buffer are mapped to the same address. When CMPCTL.CMPRDEn = 0, a compare value is written to both the TCCMPn and the register buffer; when CMPCTL.CMPRDEn = 1, a compare value is written only to the register buffer. Therefore, the double-buffering function should be disabled when writing an initial compare value to a compare register.

13.3 Register Description

TMRC Control Register

TCCR (0xFFFF_F403)

	7	6	5	4	3	2	1	0
Bit Symbol	TCEN	I2TBT						
Read/Write	R/W							
Reset Value	0	0						
Function	TMRC operation 0: Disable 1: Enable	IDLE 0: Off 1: On						

I2TBT: Enables or disables the operation of the TMRC in IDLE mode.

TCEN: Enables or disables the operation of the TMRC. If the TMRC is disabled, no clock pulses are supplied to the TMRC registers other than the TCCR, so that power consumption in the system can be reduced (only the TCCR can be read or written). To use the TMRC, set the TCEN bit to 1 before configuring other registers of the TMRC. Once the TMRC operates, all settings in its registers are held if it is disabled.

Note: Bits 0 to 6 of the TCCR are read as 0.

TBTRUN Register

TBTRUN (0xFFFF_F402)

	7	6	5	4	3	2	1	0
Bit Symbol						TBTCAP	TBTPRUN	TBTRUN
Read/Write						W	R/W	
Reset Value					0	0	0	0
Function					Must be written as 0.	TBT counter soft capture 0: Don't care 1: Soft capture	Timer run/stop control 0: Stop & clear 1: Run	

TBTRUN: Runs or stops the TBT.

TBTPRUN: Runs or stops the TBT prescaler.

TBTCAP: Setting this bit to 1 causes the TBT counter value to be captured into the capture register (TBTCAPn).

Figure 13.3 TMRC Registers

TBT Control Register

TBTCR (0xFFFF_F401)

	7	6	5	4	3	2	1	0
Bit Symbol	TBTNF				TBTCLK3	TBTCLK2	TBTCLK1	TBTCLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	TBTIN input noise elimination 0: Not eliminated 1: Eliminated	Must be written as 0.			TBT clock source 0000: ϕT1 0001: ϕT2 0010: ϕT4 0011: ϕT8 0100: ϕT16 0101: ϕT32 0110: ϕT64 0111: ϕT128 1xxx: TBTIN input			

TBTCLK[3:0]: Selects the TBT clock source. When TBTCLK[3:0] = 0000 to 0111, a prescaler output is used. When TBTCLK[3:0] = 1xxx, a clock input through the TBTIN pin is used.

TBTNF: Controls whether noise will be eliminated from the signal input through the TBTIN pin. When TBTNF = 0, the TBTIN input is directly used as the TBT clock source. When TBTNF = 1, high and low levels on TBTIN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

TBT Capture Register (TBTCAP)

TBTCAP3 (0xFFFF_F404)

	31	30	29	28	27	26	25	24
Bit Symbol	CAP031	CAP030	CAP029	CAP028	CAP027	CAP026	CAP025	CAP024
Read/Write	R							
Reset Value								
Function	Capture data							

TBTCAP2 (0xFFFF_F405)

	23	22	21	20	19	18	17	16
Bit Symbol	CAP023	CAP022	CAP021	CAP020	CAP019	CAP018	CAP017	CAP016
Read/Write	R							
Reset Value								
Function	Capture data							

TBTCAP1 (0xFFFF_F406)

	15	14	13	12	11	10	9	8
Bit Symbol	CAP015	CAP014	CAP013	CAP012	CAP011	CAP010	CAP09	CAP08
Read/Write	R							
Reset Value								
Function	Capture data							

TBTCAP0 (0xFFFF_F407)

	7	6	5	4	3	2	1	0
Bit Symbol	CAP07	CAP06	CAP05	CAP04	CAP03	CAP02	CAP01	CAP00
Read/Write	R							
Reset Value								
Function	Capture data							

Figure 13.4 TMRC Registers

TMRC Capture 0 Control Register

CAP0CR (0xFFFF_F413)

	7	6	5	4	3	2	1	0
Bit Symbol	TC0NF						CP0EG1	CP0EG0
Read/Write	R/W						R/W	
Reset Value	0						0	0
Function	TC0IN input noise elimination 0: Not eliminated 1: Eliminated						TC0IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP0EG[1:0]: Selects the edge to be detected on the TC0IN pin for the TCCAP0. When CP0EG[1:0] = 00, capture is disabled for the TCCAP0.

TC0NF: Controls whether noise will be eliminated from the signal input through the TC0IN pin. When TC0NF = 0, the TC0IN input is directly used as the TCCAP0 trigger input. When TC0NF = 1, high and low levels on TC0IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP0CR are read as 0.

Figure 13.5 TMRC Registers

TMRC Capture 0 Register (TCCAP0)

		31	30	29	28	27	26	25	24
TCCAP0HH (0xFFFF_F414)	Bit Symbol	CAP031	CAP030	CAP029	CAP028	CAP027	CAP026	CAP025	CAP024
	Read/Write	R							
	Reset Value								
	Function	Capture 0 data							
		23	22	21	20	19	18	17	16
TCCAP0HL (0xFFFF_F415)	Bit Symbol	CAP023	CAP022	CAP021	CAP020	CAP019	CAP018	CAP017	CAP016
	Read/Write	R							
	Reset Value								
	Function	Capture 0 data							
		15	14	13	12	11	10	9	8
TCCAP0LH (0xFFFF_F416)	Bit Symbol	CAP015	CAP014	CAP013	CAP012	CAP011	CAP010	CAP09	CAP08
	Read/Write	R							
	Reset Value								
	Function	Capture 0 data							
		7	6	5	4	3	2	1	0
TCCAP0LL (0xFFFF_F417)	Bit Symbol	CAP07	CAP06	CAP05	CAP04	CAP03	CAP02	CAP01	CAP00
	Read/Write	R							
	Reset Value								
	Function	Capture 0 data							

Note 1: Upon reset, the contents of the TCCAP0 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

TMRCG0 Interrupt Mask Register

		7	6	5	4	3	2	1	0
TCG0IM (0xFFFF_F40B)	Bit Symbol					TCIM3	TCIM2	TCIM1	TCIM0
	Read/Write					R/W			
	Reset Value					0	0	0	0
	Function					1: Masks INTCAP3.	1: Masks INTCAP2.	1: Masks INTCAP1.	1: Masks INTCAP0.

Note: Bits 4, 5, 6 and 7 of the TCG0IM are read as 0.

TMRCG0 Status Register

		7	6	5	4	3	2	1	0
TCG0ST (0xFFFF_F40A)	Bit Symbol					INTCAP3	INTCAP2	INTCAP1	INTCAP0
	Read/Write					R			
	Reset Value					0	0	0	0
	Function					0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

Note 1: Reading the TCG0ST register results in bits 0, 1, 2 and 3 being cleared.
Note 2: Bits 4, 5, 6 and 7 of the TCG0ST are read as 0.

Figure 13.6 TMRC Registers

TMRC Capture 1 Control Register

		7	6	5	4	3	2	1	0
CAP1CR (0xFFFF_F41B)	Bit Symbol	TC1NF						CP1EG1	CP1EG0
	Read/Write	R/W						R/W	
	Reset Value	0						0	0
	Function	TC1IN input noise elimination 0: Not eliminated 1: Eliminated						TC1IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP1EG[1:0]: Selects the edge to be detected on the TC1IN pin for the TCCAP1. When CP1EG[1:0] = 00, capture is disabled for the TCCAP1.

TC1NF: Controls whether noise will be eliminated from the signal input through the TC1IN pin. When TC1NF = 0, the TC1IN input is directly used as the TCCAP1 trigger input. When TC1NF = 1, high and low levels on TC1IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP1CR are read as 0.

TMRC Capture 1 Register (TCCAP1)

		31	30	29	28	27	26	25	24
TCCAP1HH (0xFFFF_F41C)	Bit Symbol	CAP131	CAP130	CAP129	CAP128	CAP127	CAP126	CAP125	CAP124
	Read/Write	R							
	Reset Value								
	Function	Capture 1 data							
		23	22	21	20	19	18	17	16
TCCAP1HL (0xFFFF_F41D)	Bit Symbol	CAP123	CAP122	CAP121	CAP120	CAP119	CAP118	CAP117	CAP116
	Read/Write	R							
	Reset Value								
	Function	Capture 1 data							
		15	14	13	12	11	10	9	8
TCCAP1LH (0xFFFF_F41E)	Bit Symbol	CAP115	CAP114	CAP113	CAP112	CAP111	CAP110	CAP19	CAP18
	Read/Write	R							
	Reset Value								
	Function	Capture 1 data							
		7	6	5	4	3	2	1	0
TCCAP1LL (0xFFFF_F41F)	Bit Symbol	CAP17	CAP16	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10
	Read/Write	R							
	Reset Value								
	Function	Capture 1 data							

Note 1: Upon reset, the contents of the TCCAP1 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.7 TMRC Registers

TMRC Capture 2 Control Register

CAP2CR (0xFFFF_F423)

	7	6	5	4	3	2	1	0
Bit Symbol	TC2NF						CP2EG1	CP2EG0
Read/Write	R/W						R/W	
Reset Value	0						0	0
Function	TC2IN input noise elimination 0: Not eliminated 1: Eliminated						TC2IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP2EG[1:0]: Selects the edge to be detected on the TC2IN pin for the TCCAP2. When CP2EG[1:0] = 00, capture is disabled for the TCCAP2.

TC2NF: Controls whether noise will be eliminated from the signal input through the TC2IN pin. When TC2NF = 0, the TC2IN input is directly used as the TCCAP2 trigger input. When TC2NF = 1, high and low levels on TC2IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP2CR are read as 0.

TMRC Capture 2 Register (TCCAP2)

TCCAP2HH (0xFFFF_F424)

	31	30	29	28	27	26	25	24
Bit Symbol	CAP231	CAP230	CAP229	CAP228	CAP227	CAP226	CAP225	CAP224
Read/Write	R							
Reset Value								
Function	Capture 2 data							

TCCAP2HL (0xFFFF_F425)

	23	22	21	20	19	18	17	16
Bit Symbol	CAP223	CAP222	CAP221	CAP220	CAP219	CAP218	CAP217	CAP216
Read/Write	R							
Reset Value								
Function	Capture 2 data							

TCCAP2LH (0xFFFF_F426)

	15	14	13	12	11	10	9	8
Bit Symbol	CAP215	CAP214	CAP213	CAP212	CAP211	CAP210	CAP29	CAP28
Read/Write	R							
Reset Value								
Function	Capture 2 data							

TCCAP2LL (0xFFFF_F427)

	7	6	5	4	3	2	1	0
Bit Symbol	CAP27	CAP26	CAP25	CAP24	CAP23	CAP22	CAP21	CAP20
Read/Write	R							
Reset Value								
Function	Capture 2 data							

Note 1: Upon reset, the contents of the TCCAP2 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.8 TMRC Registers

TMRC Capture 3 Control Register

CAP3CR (0xFFFF_F42B)

	7	6	5	4	3	2	1	0
Bit Symbol	TC3NF						CP3EG1	CP3EG0
Read/Write	R/W						R/W	
Reset Value	0						0	0
Function	TC3IN input noise elimination 0: Not eliminated 1: Eliminated						TC3IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP3EG[1:0]: Selects the edge to be detected on the TC3IN pin for the TCCAP3. When CP3EG[1:0] = 00, capture is disabled for the TCCAP3.

TC3NF: Controls whether noise will be eliminated from the signal input through the TC3IN pin. When TC3NF = 0, the TC3IN input is directly used as the TCCAP3 trigger input. When TC3NF = 1, high and low levels on TC3IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP3CR are read as 0.

TMRC Capture 3 Register (TCCAP3)

TCCAP3HH (0xFFFF_F42C) / TCCAP3HL (0xFFFF_F42D) / TCCAP3LH (0xFFFF_F42E) / TCCAP3LL (0xFFFF_F42F)

	31	30	29	28	27	26	25	24
Bit Symbol	CAP331	CAP330	CAP329	CAP328	CAP327	CAP326	CAP325	CAP324
Read/Write	R							
Reset Value								
Function	Capture 3 data							
	23	22	21	20	19	18	17	16
Bit Symbol	CAP323	CAP322	CAP321	CAP320	CAP319	CAP318	CAP317	CAP316
Read/Write	R							
Reset Value								
Function	Capture 3 data							
	15	14	13	12	11	10	9	8
Bit Symbol	CAP315	CAP314	CAP313	CAP312	CAP311	CAP310	CAP39	CAP38
Read/Write	R							
Reset Value								
Function	Capture 3 data							
	7	6	5	4	3	2	1	0
Bit Symbol	CAP37	CAP36	CAP35	CAP34	CAP33	CAP32	CAP31	CAP30
Read/Write	R							
Reset Value								
Function	Capture 3 data							

Note 1: Upon reset, the contents of the TCCAP3 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.9 TMRC Registers

TMRC Capture 4 Control Register

CAP4CR (0xFFFF_F433)

	7	6	5	4	3	2	1	0
Bit Symbol	TC4NF						CP4EG1	CP4EG0
Read/Write	R/W						R/W	
Reset Value	0						0	0
Function	TC4IN input noise elimination 0: Not eliminated 1: Eliminated						TC4IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP4EG[1:0]: Selects the edge to be detected on the TC4IN pin for the TCCAP4. When CP4EG[1:0] = 00, capture is disabled for the TCCAP4.

TC4NF: Controls whether noise will be eliminated from the signal input through the TC4IN pin. When TC4NF = 0, the TC4IN input is directly used as the TCCAP4 trigger input. When TC4NF = 1, high and low levels on TC4IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP4CR are read as 0.

TMRC Capture 4 Register (TCCAP4)

TCCAP4HH (0xFFFF_F434)

	31	30	29	28	27	26	25	24
Bit Symbol	CAP431	CAP430	CAP429	CAP428	CAP427	CAP426	CAP425	CAP424
Read/Write	R							
Reset Value								
Function	Capture 4 data							

TCCAP4HL (0xFFFF_F435)

	23	22	21	20	19	18	17	16
Bit Symbol	CAP423	CAP422	CAP421	CAP420	CAP419	CAP418	CAP417	CAP416
Read/Write	R							
Reset Value								
Function	Capture 4 data							

TCCAP4LH (0xFFFF_F436)

	15	14	13	12	11	10	9	8
Bit Symbol	CAP415	CAP414	CAP413	CAP412	CAP411	CAP410	CAP49	CAP48
Read/Write	R							
Reset Value								
Function	Capture 4 data							

TCCAP4LL (0xFFFF_F437)

	7	6	5	4	3	2	1	0
Bit Symbol	CAP47	CAP46	CAP45	CAP44	CAP43	CAP42	CAP41	CAP40
Read/Write	R							
Reset Value								
Function	Capture 4 data							

Note 1: Upon reset, the contents of the TCCAP4 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.10 TMRC Registers

TMRCG1 Interrupt Mask Register

TCG1IM (0xFFFF_F409)

	7	6	5	4	3	2	1	0
Bit Symbol				TBTIM	TCIM7	TCIM6	TCIM5	TCIM4
Read/Write				R/W				
Reset Value				0	0	0	0	0
Function				1: Masks INTTBT.	1: Masks INTCAP7.	1: Masks INTCAP6.	1: Masks INTCAP5.	1: Masks INTCAP4.

Note: Bits 5, 6 and 7 of the TCG1IM are read as 0.

TMRCG1 Status Register

TCG1ST (0xFFFF_F408)

	7	6	5	4	3	2	1	0
Bit Symbol				INTTBT	INTCAP7	INTCAP6	INTCAP5	INTCAP4
Read/Write				R				
Reset Value				0	0	0	0	0
Function				0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated	0: No interrupt generated 1: Interrupt generated

Note 1: Reading the TCG1ST register results in bits 0, 1, 2, 3 and 4 being cleared.
Note 2: Bits 5, 6 and 7 of the TCG1ST are read as 0.

Figure 13.11 TMRC Registers

TMRC Capture 5 Control Register

CAP5CR (0xFFFF_F43B)

	7	6	5	4	3	2	1	0
Bit Symbol	TC5NF						CP5EG1	CP5EG0
Read/Write	R/W						R/W	
Reset Value	0						0	0
Function	TC5IN input noise elimination 0: Not eliminated 1: Eliminated						TC5IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP5EG[1:0]: Selects the edge to be detected on the TC5IN pin for the TCCAP5. When CP5EG[1:0] = 00, capture is disabled for the TCCAP5.

TC5NF: Controls whether noise will be eliminated from the signal input through the TC5IN pin. When TC5NF = 0, the TC5IN input is directly used as the TCCAP5 trigger input. When TC5NF = 1, high and low levels on TC5IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP5CR are read as 0.

TMRC Capture 5 Register (TCCAP5)

TCCAP5HH (0xFFFF_F43C), TCCAP5HL (0xFFFF_F43D), TCCAP5LH (0xFFFF_F43E), TCCAP5LL (0xFFFF_F43F)

	31	30	29	28	27	26	25	24
Bit Symbol	CAP531	CAP530	CAP529	CAP528	CAP527	CAP526	CAP525	CAP524
Read/Write	R							
Reset Value								
Function	Capture 5 data							
	23	22	21	20	19	18	17	16
Bit Symbol	CAP523	CAP522	CAP521	CAP520	CAP519	CAP518	CAP517	CAP516
Read/Write	R							
Reset Value								
Function	Capture 5 data							
	15	14	13	12	11	10	9	8
Bit Symbol	CAP515	CAP514	CAP513	CAP512	CAP511	CAP510	CAP59	CAP58
Read/Write	R							
Reset Value								
Function	Capture 5 data							
	7	6	5	4	3	2	1	0
Bit Symbol	CAP57	CAP56	CAP55	CAP54	CAP53	CAP52	CAP51	CAP50
Read/Write	R							
Reset Value								
Function	Capture 5 data							

Note 1: Upon reset, the contents of the TCCAP5 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.12 TMRC Registers

TMRC Capture 6 Control Register

		7	6	5	4	3	2	1	0
CAP6CR (0xFFFF_F443)	Bit Symbol	TC6NF						CP6EG1	CP6EG0
	Read/Write	R/W						R/W	
	Reset Value	0						0	0
	Function	TC6IN input noise elimination 0: Not eliminated 1: Eliminated						TC6IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP6EG[1:0]: Selects the edge to be detected on the TC6IN pin for the TCCAP6. When CP6EG[1:0] = 00, capture is disabled for the TCCAP6.

TC6NF: Controls whether noise will be eliminated from the signal input through the TC6IN pin. When TC6NF = 0, the TC6IN input is directly used as the TCCAP6 trigger input. When TC6NF = 1, high and low levels on TC6IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP6CR are read as 0.

TMRC Capture 6 Register (TCCAP6)

		31	30	29	28	27	26	25	24
TCCAP6HH (0xFFFF_F444)	Bit Symbol	CAP631	CAP630	CAP629	CAP628	CAP627	CAP626	CAP625	CAP624
	Read/Write	R							
	Reset Value								
	Function	Capture 6 data							
		23	22	21	20	19	18	17	16
TCCAP6HL (0xFFFF_F445)	Bit Symbol	CAP623	CAP622	CAP621	CAP620	CAP619	CAP618	CAP617	CAP616
	Read/Write	R							
	Reset Value								
	Function	Capture 6 data							
		15	14	13	12	11	10	9	8
TCCAP6LH (0xFFFF_F446)	Bit Symbol	CAP615	CAP614	CAP613	CAP612	CAP611	CAP610	CAP69	CAP68
	Read/Write	R							
	Reset Value								
	Function	Capture 6 data							
		7	6	5	4	3	2	1	0
TCCAP6LL (0xFFFF_F447)	Bit Symbol	CAP67	CAP66	CAP65	CAP64	CAP63	CAP62	CAP61	CAP60
	Read/Write	R							
	Reset Value								
	Function	Capture 6 data							

Note 1: Upon reset, the contents of the TCCAP6 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.13 TMRC Registers

TMRC Capture 7 Control Register

CAP7CR (0xFFFF_F44B)

	7	6	5	4	3	2	1	0
Bit Symbol	TC7NF						CP7EG1	CP7EG0
Read/Write	R/W						R/W	
Reset Value	0						0	0
Function	TC7IN input noise elimination 0: Not eliminated 1: Eliminated						TC7IN edge detection 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	

CP7EG[1:0]: Selects the edge to be detected on the TC7IN pin for the TCCAP7. When CP7EG[1:0] = 00, capture is disabled for the TCCAP7.

TC7NF: Controls whether noise will be eliminated from the signal input through the TC7IN pin. When TC7NF = 0, the TC7IN input is directly used as the TCCAP7 trigger input. When TC7NF = 1, high and low levels on TC7IN shorter than 4/fsys (99 ns @fperiph = fc = 40.5 MHz) are regarded as noise and eliminated from the input. The elimination threshold varies with the clock gear setting.

Note: Bits 2 to 6 of the CAP7CR are read as 0.

TMRC Capture 7 Register (TCCAP7)

TCCAP7HH (0xFFFF_F44C)

	31	30	29	28	27	26	25	24
Bit Symbol	CAP731	CAP730	CAP729	CAP728	CAP727	CAP726	CAP725	CAP724
Read/Write	R							
Reset Value								
Function	Capture 7 data							

TCCAP7HL (0xFFFF_F44D)

	23	22	21	20	19	18	17	16
Bit Symbol	CAP723	CAP722	CAP721	CAP720	CAP719	CAP718	CAP717	CAP716
Read/Write	R							
Reset Value								
Function	Capture 7 data							

TCCAP7LH (0xFFFF_F44E)

	15	14	13	12	11	10	9	8
Bit Symbol	CAP715	CAP714	CAP713	CAP712	CAP711	CAP710	CAP79	CAP78
Read/Write	R							
Reset Value								
Function	Capture 7 data							

TCCAP7LL (0xFFFF_F44F)

	7	6	5	4	3	2	1	0
Bit Symbol	CAP77	CAP76	CAP75	CAP74	CAP73	CAP72	CAP71	CAP70
Read/Write	R							
Reset Value								
Function	Capture 7 data							

Note 1: Upon reset, the contents of the TCCAP7 are undefined.
Note 2: The counter value is not captured while the capture register is being read.

Figure 13.14 TMRC Registers

TMRC Compare Control Register (CMPCTL)

		31	30	29	28	27	26	25	24
CMPCTL7 (0xFFFF_F474)	Bit Symbol		TCFFEN7	TCFFC71	TCFFC70			CMPRDE7	CMPEN7
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF7 toggle trigger 0: Disable 1: Enable	TCFF7 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 7 0: Disable 1: Enable	Compare 7 0: Disable 1: Enable
		23	22	21	20	19	18	17	16
CMPCTL6 (0xFFFF_F475)	Bit Symbol		TCFFEN6	TCFFC61	TCFFC60			CMPRDE6	CMPEN6
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF6 toggle trigger 0: Disable 1: Enable	TCFF6 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 6 0: Disable 1: Enable	Compare 6 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
CMPCTL5 (0xFFFF_F476)	Bit Symbol		TCFFEN5	TCFFC51	TCFFC50			CMPRDE5	CMPEN5
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF5 toggle trigger 0: Disable 1: Enable	TCFF5 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 5 0: Disable 1: Enable	Compare 5 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
CMPCTL4 (0xFFFF_F477)	Bit Symbol		TCFFEN4	TCFFC41	TCFFC40			CMPRDE4	CMPEN4
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF4 toggle trigger 0: Disable 1: Enable	TCFF4 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 4 0: Disable 1: Enable	Compare 4 0: Disable 1: Enable

TMRC Compare Control Register (CMPCTL)

		31	30	29	28	27	26	25	24
CMPCTL3 (0xFFFF_F470)	Bit Symbol		TCFFEN3	TCFFC31	TCFFC30			CMPRDE3	CMPEN3
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF3 toggle trigger 0: Disable 1: Enable	TCFF3 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 3 0: Disable 1: Enable	Compare 3 0: Disable 1: Enable
		23	22	21	20	19	18	17	16
CMPCTL2 (0xFFFF_F471)	Bit Symbol		TCFFEN2	TCFFC21	TCFFC20			CMPRDE2	CMPEN2
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF2 toggle trigger 0: Disable 1: Enable	TCFF2 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 2 0: Disable 1: Enable	Compare 2 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
CMPCTL1 (0xFFFF_F472)	Bit Symbol		TCFFEN1	TCFFC11	TCFFC10			CMPRDE1	CMPEN1
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF1 toggle trigger 0: Disable 1: Enable	TCFF1 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 1 0: Disable 1: Enable	Compare 1 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
CMPCTL0 (0xFFFF_F473)	Bit Symbol		TCFFEN0	TCFFC01	TCFFC00			CMPRDE0	CMPEN0
	Read/Write		R/W	W				R/W	
	Reset Value	0	0	1	1	0	0	0	0
	Function		TCFF0 toggle trigger 0: Disable 1: Enable	TCFF0 control 00: Toggle 01: Set 10: Clear 11: Don't care				Double-buffering 0 0: Disable 1: Enable	Compare 0 0: Disable 1: Enable

CMPENn: Enables or disables the detection of a match in comparison.

CMPRDEn: Enables or disables double-buffering for the compare register.

TCFFCn[1:0]: Controls the compare match output flip-flop.

TCFFENn: Enables or disables the toggling of the compare match output flip-flop.

Note: Bits 31, 27, 26, 23, 19, 18, 15, 11, 10, 7, 3 and 2 of the CMPTCL are read as 0.

Figure 13.15 TMRC Registers

TMRC Compare Register 0 (TCCMP0)

		31	30	29	28	27	26	25	24
TCCMP0HH (0xFFFF_F450)	Bit Symbol	CMP031	CMP030	CMP029	CMP028	CMP027	CMP026	CMP025	CMP024
	Read/Write	W							
	Reset Value								
	Function	Compare register 0 data							
		23	22	21	20	19	18	17	16
TCCMP0HL (0xFFFF_F451)	Bit Symbol	CMP023	CMP022	CMP021	CMP020	CMP019	CMP018	CMP017	CMP016
	Read/Write	W							
	Reset Value								
	Function	Compare register 0 data							
		15	14	13	12	11	10	9	8
TCCMP0LH (0xFFFF_F452)	Bit Symbol	CMP015	CMP014	CMP013	CMP012	CMP011	CMP010	CMP09	CMP08
	Read/Write	W							
	Reset Value								
	Function	Compare register 0 data							
		7	6	5	4	3	2	1	0
TCCMP0LL (0xFFFF_F453)	Bit Symbol	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00
	Read/Write	W							
	Reset Value								
	Function	Compare register 0 data							

Note: The TCCMP0 is a write-only register. Upon reset, its contents are undefined.

TMRC Compare Register 1 (TCCMP1)

		31	30	29	28	27	26	25	24
TCCMP1HH (0xFFFF_F454)	Bit Symbol	CMP131	CMP130	CMP129	CMP128	CMP127	CMP126	CMP125	CMP124
	Read/Write	W							
	Reset Value								
	Function	Compare register 1 data							
		23	22	21	20	19	18	17	16
TCCMP1HL (0xFFFF_F455)	Bit Symbol	CMP123	CMP122	CMP121	CMP120	CMP119	CMP118	CMP117	CMP116
	Read/Write	W							
	Reset Value								
	Function	Compare register 1 data							
		15	14	13	12	11	10	9	8
TCCMP1LH (0xFFFF_F456)	Bit Symbol	CMP115	CMP114	CMP113	CMP112	CMP111	CMP110	CMP19	CMP18
	Read/Write	W							
	Reset Value								
	Function	Compare register 1 data							
		7	6	5	4	3	2	1	0
TCCMP1LL (0xFFFF_F457)	Bit Symbol	CMP17	CMP16	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10
	Read/Write	W							
	Reset Value								
	Function	Compare register 1 data							

Note: The TCCMP1 is a write-only register. Upon reset, its contents are undefined.

Figure 13.16 TMRC Registers

TMRC Compare Register 2 (TCCMP2)

		31	30	29	28	27	26	25	24
TCCMP2HH (0xFFFF_F458)	Bit Symbol	CMP231	CMP230	CMP229	CMP228	CMP227	CMP226	CMP225	CMP224
	Read/Write	W							
	Reset Value								
	Function	Compare register 2 data							
		23	22	21	20	19	18	17	16
TCCMP2HL (0xFFFF_F459)	Bit Symbol	CMP223	CMP222	CMP221	CMP220	CMP219	CMP218	CMP217	CMP216
	Read/Write	W							
	Reset Value								
	Function	Compare register 2 data							
		15	14	13	12	11	10	9	8
TCCMP2LH (0xFFFF_F45A)	Bit Symbol	CMP215	CMP214	CMP213	CMP212	CMP211	CMP210	CMP29	CMP28
	Read/Write	W							
	Reset Value								
	Function	Compare register 2 data							
		7	6	5	4	3	2	1	0
TCCMP2LL (0xFFFF_F45B)	Bit Symbol	CMP27	CMP26	CMP25	CMP24	CMP23	CMP22	CMP21	CMP20
	Read/Write	W							
	Reset Value								
	Function	Compare register 2 data							

Note: The TCCMP2 is a write-only register. Upon reset, its contents are undefined.

TMRC Compare Register 3 (TCCMP3)

		31	30	29	28	27	26	25	24
TCCMP3HH (0xFFFF_F45C)	Bit Symbol	CMP331	CMP330	CMP329	CMP328	CMP327	CMP326	CMP325	CMP324
	Read/Write	W							
	Reset Value								
	Function	Compare register 3 data							
		23	22	21	20	19	18	17	16
TCCMP3HL (0xFFFF_F45D)	Bit Symbol	CMP323	CMP322	CMP321	CMP320	CMP319	CMP318	CMP317	CMP316
	Read/Write	W							
	Reset Value								
	Function	Compare register 3 data							
		15	14	13	12	11	10	9	8
TCCMP3LH (0xFFFF_F45E)	Bit Symbol	CMP315	CMP314	CMP313	CMP312	CMP311	CMP310	CMP39	CMP38
	Read/Write	W							
	Reset Value								
	Function	Compare register 3 data							
		7	6	5	4	3	2	1	0
TCCMP3LL (0xFFFF_F45F)	Bit Symbol	CMP37	CMP36	CMP35	CMP34	CMP33	CMP32	CMP31	CMP30
	Read/Write	W							
	Reset Value								
	Function	Compare register 3 data							

Note: The TCCMP3 is a write-only register. Upon reset, its contents are undefined.

Figure 13.17 TMRC Registers

TMRC Compare Register 4 (TCCMP4)

		31	30	29	28	27	26	25	24
TCCMP4HH (0xFFFF_F460)	Bit Symbol	CMP431	CMP430	CMP429	CMP428	CMP427	CMP426	CMP425	CMP424
	Read/Write	W							
	Reset Value								
	Function	Compare register 4 data							
		23	22	21	20	19	18	17	16
TCCMP4HL (0xFFFF_F461)	Bit Symbol	CMP423	CMP422	CMP421	CMP420	CMP419	CMP418	CMP417	CMP416
	Read/Write	W							
	Reset Value								
	Function	Compare register 4 data							
		15	14	13	12	11	10	9	8
TCCMP4LH (0xFFFF_F462)	Bit Symbol	CMP415	CMP414	CMP413	CMP412	CMP411	CMP410	CMP49	CMP48
	Read/Write	W							
	Reset Value								
	Function	Compare register 4 data							
		7	6	5	4	3	2	1	0
TCCMP4LL (0xFFFF_F463)	Bit Symbol	CMP47	CMP46	CMP45	CMP44	CMP43	CMP42	CMP41	CMP40
	Read/Write	W							
	Reset Value								
	Function	Compare register 4 data							

Note: The TCCMP4 is a write-only register. Upon reset, its contents are undefined.

TMRC Compare Register 5 (TCCMP5)

		31	30	29	28	27	26	25	24
TCCMP5HH (0xFFFF_F464)	Bit Symbol	CMP531	CMP530	CMP529	CMP528	CMP527	CMP526	CMP525	CMP524
	Read/Write	W							
	Reset Value								
	Function	Compare register 5 data							
		23	22	21	20	19	18	17	16
TCCMP5HL (0xFFFF_F465)	Bit Symbol	CMP523	CMP522	CMP521	CMP520	CMP519	CMP518	CMP517	CMP516
	Read/Write	W							
	Reset Value								
	Function	Compare register 5 data							
		15	14	13	12	11	10	9	8
TCCMP5LH (0xFFFF_F466)	Bit Symbol	CMP515	CMP514	CMP513	CMP512	CMP511	CMP510	CMP59	CMP58
	Read/Write	W							
	Reset Value								
	Function	Compare register 5 data							
		7	6	5	4	3	2	1	0
TCCMP5LL (0xFFFF_F467)	Bit Symbol	CMP57	CMP56	CMP55	CMP54	CMP53	CMP52	CMP51	CMP50
	Read/Write	W							
	Reset Value								
	Function	Compare register 5 data							

Note: The TCCMP5 is a write-only register. Upon reset, its contents are undefined.

Figure 13.18 TMRC Registers

TMRC Compare Register 6 (TCCMP6)

		31	30	29	28	27	26	25	24
TCCMP6HH (0xFFFF_F468)	Bit Symbol	CMP631	CMP630	CMP629	CMP628	CMP627	CMP626	CMP625	CMP624
	Read/Write	W							
	Reset Value								
	Function	Compare register 6 data							
		23	22	21	20	19	18	17	16
TCCMP6HL (0xFFFF_F469)	Bit Symbol	CMP623	CMP622	CMP621	CMP620	CMP619	CMP618	CMP617	CMP616
	Read/Write	W							
	Reset Value								
	Function	Compare register 6 data							
		15	14	13	12	11	10	9	8
TCCMP6LH (0xFFFF_F46A)	Bit Symbol	CMP615	CMP614	CMP613	CMP612	CMP611	CMP610	CMP69	CMP68
	Read/Write	W							
	Reset Value								
	Function	Compare register 6 data							
		7	6	5	4	3	2	1	0
TCCMP6LL (0xFFFF_F46B)	Bit Symbol	CMP67	CMP66	CMP65	CMP64	CMP63	CMP62	CMP61	CMP60
	Read/Write	W							
	Reset Value								
	Function	Compare register 6 data							

Note: The TCCMP6 is a write-only register. Upon reset, its contents are undefined.

TMRC Compare Register 7 (TCCMP7)

		31	30	29	28	27	26	25	24
TCCMP7HH (0xFFFF_F46C)	Bit Symbol	CMP731	CMP730	CMP729	CMP728	CMP727	CMP726	CMP725	CMP724
	Read/Write	W							
	Reset Value								
	Function	Compare register 7 data							
		23	22	21	20	19	18	17	16
TCCMP7HL (0xFFFF_F46D)	Bit Symbol	CMP723	CMP722	CMP721	CMP720	CMP719	CMP718	CMP717	CMP716
	Read/Write	W							
	Reset Value								
	Function	Compare register 7 data							
		15	14	13	12	11	10	9	8
TCCMP7LH (0xFFFF_F46E)	Bit Symbol	CMP715	CMP714	CMP713	CMP712	CMP711	CMP710	CMP79	CMP78
	Read/Write	W							
	Reset Value								
	Function	Compare register 7 data							
		7	6	5	4	3	2	1	0
TCCMP7LL (0xFFFF_F46F)	Bit Symbol	CM77	CMP76	CMP75	CMP74	CMP73	CMP72	CMP71	CMP70
	Read/Write	W							
	Reset Value								
	Function	Compare register 7 data							

Note: The TCCMP7 is a write-only register. Upon reset, its contents are undefined.

Figure 13.19 TMRC Registers

14. Serial I/O (SIO)

The TMP1962 serial I/O contains seven channels (SIO0-SIO6). Each serial channel provides Universal Asynchronous Receiver/Transmitter (UART) mode and synchronous I/O Interface mode.

- I/O Interface mode — Mode 0: Transmits/receives a serial clock (SCLK) as well as data streams for a synchronous clock mode of operation.
- UART mode — Mode 1: 7 data bits; Mode 2: 8 data bits; Mode 3: 9 data bits

In Mode 1 and Mode 2, each frame can include a parity bit. In Mode 3, an SIO channel operates in a wakeup mode for multidrop applications in which a master station is connected to several slave stations through a serial link. Figure 14.2 shows a block diagram of the SIO0.

The main components of an SIO channel are a clock prescaler, a serial clock generator, a receive buffer, a receive controller, a transmit buffer and a transmit controller. Each SIO channel is independently programmable, and functionally equivalent. In the following sections, any references to the SIO0 also apply to the other channels.

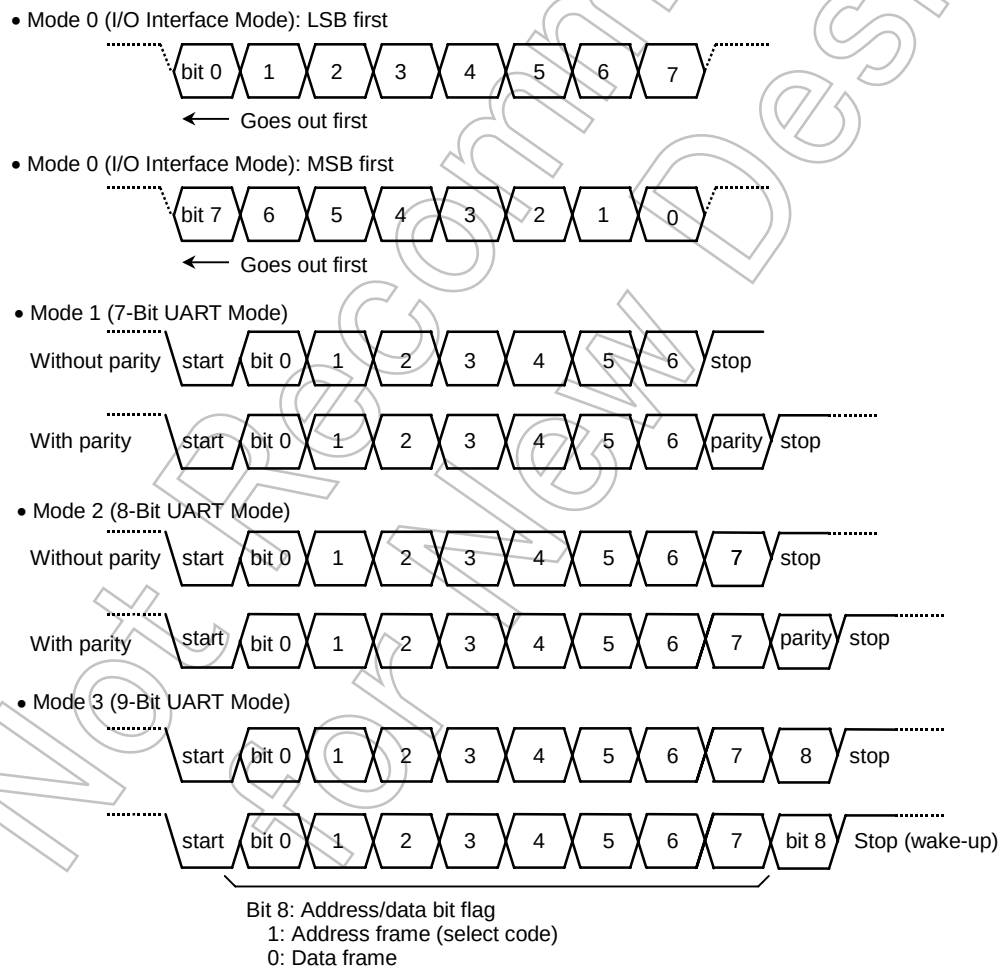


Figure 14.1 Data Formats

14.1 Block Diagram (Channel 0)

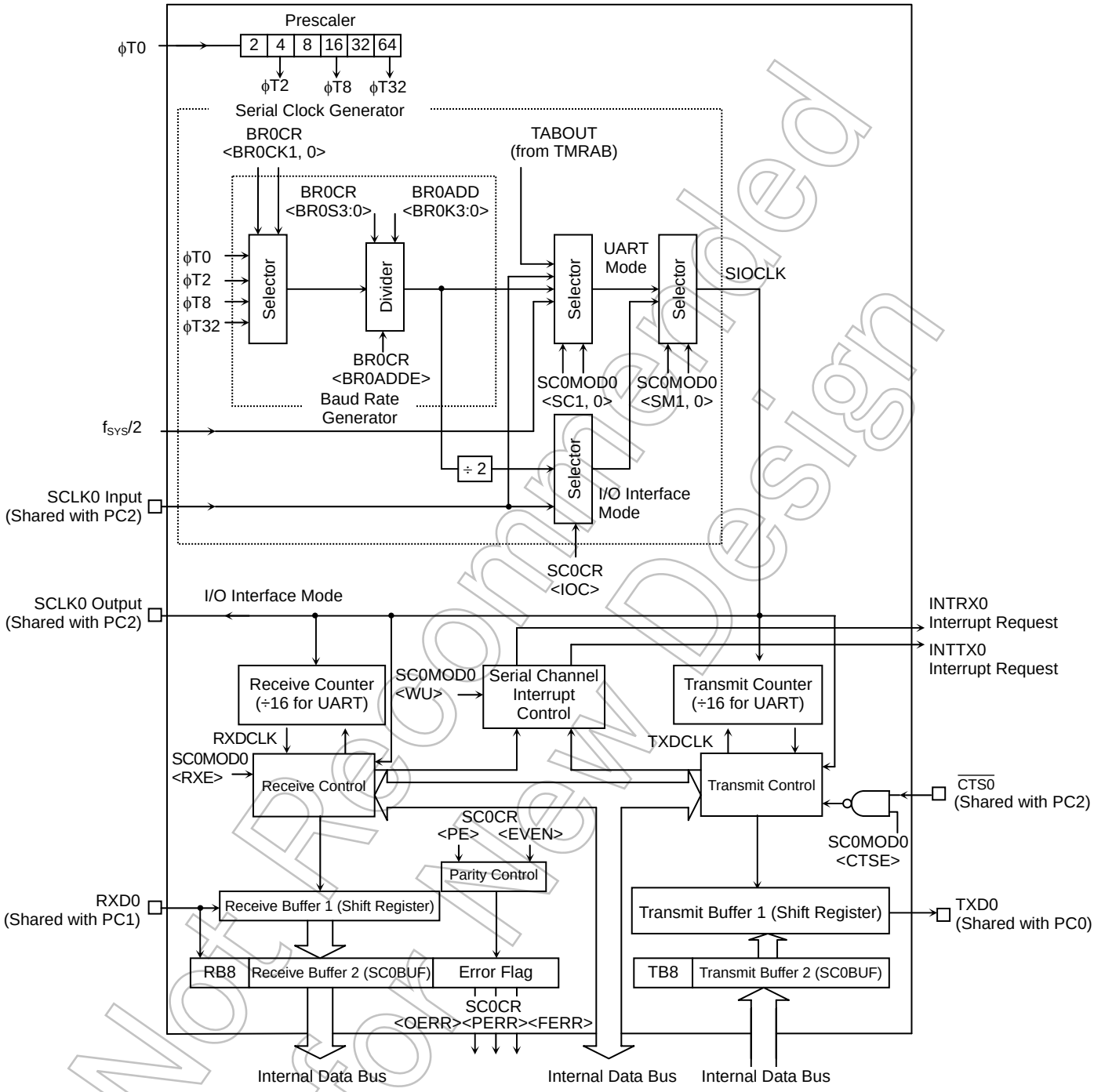


Figure 14.2 SIO Block Diagram

14.2 SIO Components (Channel 0)

14.2.1 Prescaler

The SIO0 has a 6-bit prescaler that slows the rate of a clocking source to the serial clock generator. The prescaler clock source (ϕT0) can be selected from fperiph/4, fperiph/8 and fperiph/16 by programming the PRCK[1:0] field of the SYSCR located within the CG. fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The serial clock is selectable from several clocks; the prescaler is only enabled when the baud rate generator output clock is selected as a serial clock. Table 14.1 shows prescaler output clock resolutions.

Table 14.1 Prescaler Output Clock Resolutions

@ = 40.5 MHz

Peripheral Clock Select FPSEL	Clock Gear Value GEAR[1:0]	Prescaler Clock Source PRCK[1:0]	Prescaler Output Clock Resolution			
			ϕT0	ϕT2	ϕT8	ϕT32
0 (fgear)	00 (fc)	00 (fperiph/16)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
		01 (fperiph/8)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)
		10 (fperiph/4)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)
	01 (fc/2)	00 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)	$fc/2^{11}$ (50.6μs)
		01 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
		10 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)
	10 (fc/4)	00 (fperiph/16)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)	$fc/2^{12}$ (101 μs)
		01 (fperiph/8)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)	$fc/2^{11}$ (50.6μs)
		10 (fperiph/4)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
	11 (fc/8)	00 (fperiph/16)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)	$fc/2^{11}$ (50.6μs)	$fc/2^{13}$ (202μs)
		01 (fperiph/8)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)	$fc/2^{12}$ (101μs)
		10 (fperiph/4)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μ μs)	$fc/2^9$ (12.6μs)	$fc/2^{11}$ (50.6μs)
1 (fc)	00 (fc)	00 (fperiph/16)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
		01 (fperiph/8)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)
		10 (fperiph/4)	$fc/2^2$ (0.1 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)
	01 (fc/2)	00 (fperiph/16)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
		01 (fperiph/8)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)
		10 (fperiph/4)	—	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)
	10 (fc/4)	00 (fperiph/16)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
		01 (fperiph/8)	—	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)
		10 (fperiph/4)	—	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)
	11 (fc/8)	00 (fperiph/16)	—	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)	$fc/2^{10}$ (25.3 μs)
		01 (fperiph/8)	—	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.6μs)
		10 (fperiph/4)	—	—	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.3 μs)

Note 1: The prescaler's output clock ϕTn must be selected so that the relationship ϕTn < fsys/2 is satisfied.
Note 2: Do not change the clock gear value while the SIO0 is operating.
Note 3: The - character means "Setting prohibited.

Prescaler output taps can be divide-by-1 (ϕT0), divide-by-4 (ϕT2), divide-by-16 (ϕT8) and divide-by-64 (ϕT32).

14.2.2 Baud rate generator

The frequency used to transmit and receive data through the SIO0 is derived from the baud rate generator. The clock source for the baud rate generator can be selected from the 6-bit prescaler outputs (ϕT0, ϕT2, ϕT8, ϕT32) through the programming of the BR0CK[1:0] field in the BR0CR.

The baud rate generator contains a clock divider that can divide the selected clock by 1, N + (m/16), or 16 (where N is an integer between 2 and 15, and m is an integer between 0 and 15). The clock divisor is programmed into the BR0ADDE and BR0S[3:0] bits in the BR0CR and the BR0K[3:0] bits in the BR0ADD.

- UART mode

  (1) When BR0CR.BR0ADDE = 0

  When the BR0CR.BR0ADDE bit is cleared, the BR0ADD.BR0K[3:0] field has no meaning or effect. In this case, the baud rate generator input clock is divided down by a value of N (1 to 16) programmed in the BR0CR.BR0S[3:0] field.

  (2) When BR0CR.BR0ADDE = 1

  Setting the BR0CR.BR0ADDE bit enables the N + (16 - K)/16 clock division function. The baud rate generator input clock is divided down according to the value of N (2 to 15) programmed in the BR0CR.BR0S[3:0] field and the value of K (1 to 15) programmed in the BR0ADD.BR0K[3:0] field.

Note: Setting N to 1 or 16 disables the N + (16 - K)/16 clock division function. When N = 1 or 16, the BR0CR.BR0ADDE bit must be cleared.

- I/O Interface mode

  I/O Interface mode cannot utilize the N + (16 - K)/16 clock division function. The BR0CR.BR0ADDE must be cleared, so the baud rate generator input clock is divided down by a value of N (1 to 16) programmed in the BR0CR.BR0S[3:0] field.

- Baud rate calculations

  (1) UART mode

  Baud rate = baud rate generator input clock / baud rate generator divisor ÷ 16

  When the clock input to the baud rate generator is 10.125-MHz ϕT0, the maximum baud rate is 632.8 kbps.

  The baud rate generator can be bypassed if the user wants to use the fsys/2 clock as a serial clock. In this case, the maximum baud rate is 1.266 Mbps @fsys = 40.5 MHz.

(2) I/O Interface mode

Baud rate = baud rate generator input clock / baud rate generator divisor ÷ 2

When the clock input to the baud rate generator is 10.125-MHz ϕT0, the maximum baud rate is 5.06 Mbps (with no clock division by the baud rate generator) if double-buffering is used, or 2.53 Mbps (with the clock divided by 2 by the baud rate generator) if double-buffering is not used.

- Calculation examples

(1) Integral clock division (divide-by-N)

fperiph = 40.5-MHz fc

ϕT0 = fperiph/16

Baud rate generator input clock: ϕT2

Clock divisor N (BR0CR.BR0S[3:0]) = 4

BR0CR.BR0ADDE = 0

Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	x1 (fc)
	Prescaler clock:	f_{periph}/16 ($f_{periph} = f_{sys}$)

The baud rate in UART mode is determined as follows:

Baud rate = (fc/64)/4 ÷ 16

= 40.5 x 10^6 ÷ 64 ÷ 4 ÷ 16 = 9888 (bps)

Note: Clearing the BR0CR.BR0ADDE bit to 0 disables the N + (16 - K)/16 clock division function. At this time, the BR0ADD.BR0K[3:0] field is ignored.

(2) N + (16 - K)/16 clock division (UART mode only)

fperiph = 19.2-MHz fc

ϕT0 = fperiph/16

Baud rate generator input clock: ϕT2

N (BR0CR.BR0S[3:0]) = 4

K (BR0ADD.BR0K[3:0]) = 14

BR0CR.BR0ADDE = 1

Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	x1 (fc)
	Prescaler clock:	f_{periph}/4 ($f_{periph} = f_{sys}$)

The baud rate is determined as follows:

Baud Rate = (fc/64)/(4 + (16-14)/16) ÷ 16

= 40.5 x 10^6 ÷ 64 ÷ (4 + 2/16) ÷ 16 = 9588 (bps)

The SIO0 can use an external clock as a serial clock, bypassing the baud rate generator. When an external clock is used, the baud rate is determined as shown below.

- Using an external clock as a serial clock

(1) UART mode

Baud rate = external clock input ÷ 16

The external clock period must be greater than or equal to 4/fsys. Therefore, when fsys =40.5 MHz, the maximum baud rate is 632.8 kbps (40.5 ÷ 4 ÷ 16).

(2) I/O Interface mode

Baud rate = external clock input clock

When double-buffering is used, the external clock period must be greater than 12/fsys. Therefore, when fsys = 40.5 MHz, the maximum baud rate is 3.375 Mbps (40.5 ÷ 12).

When double-buffering is not used, the external clock period must be greater than 16/fsys. Therefore, when fsys = 40.5 MHz, the maximum baud rate is 2.53 Mbps (40.5 ÷ 16).

Table 14.2 and Table 14.3 show the UART baud rates obtained with various combinations of clock inputs and clock divisor values.

Table 14.2 UART Baud Rate Selection

(When the baud rate generator is used and BR0CR.BR0ADDE = 0) Unit: kbps

fc [MHz]	Baud Rate Generator Input Clock Divisor N (Programmed in BR0CR.BR0S[3:0])	ϕT0 (fc/4)	ϕT2 (fc/16)	ϕT8 (fc/64)	ϕT32 (fc/256)
19.6608	1	307.200	76.800	19.200	4.800
↑	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	0	19.200	4.800	1.200	0.300
24.576	5	76.800	19.200	4.800	1.200
↑	A	38.400	9.600	2.400	0.600
29.4912	1	460.800	115.200	28.800	7.200
↑	2	230.400	57.600	14.400	3.600
↑	3	153.600	38.400	9.600	2.400
↑	4	115.200	28.800	7.200	1.800
↑	6	76.800	19.200	4.800	1.200
↑	C	38.400	9.600	2.400	0.600

Note: This table assumes: fsys = fc, clock gear = fc/1, prescaler clock source = $f_{periph}/4$

Table 14.3 UART Baud Rate Selection

(When the TMRAB timer trigger output (internal TABOUT) is used and the TMRAB input clock is ϕT1)

Unit: kbps

fc TA0REG	29.4912 MHz	24.576 MHz	24 MHz	19.6608 MHz	16 MHz	12.288 MHz
1H	230.4	192	187.5	153.6	125	96
2H	115.2	96	93.75	76.8	62.5	48
3H	76.8	64	62.5	51.2	41.67	32
4H	57.6	48	46.88	38.4	31.25	24
5H	46.08	38.4	37.5	30.72	25	19.2
6H	38.4	32	31.25	25.6	20.83	16
8H	28.8	24	23.44	19.2	15.63	12
AH	23.04	19.2	18.75	15.36	12.5	9.6
10H	14.4	12	11.72	9.6	7.81	6
14H	11.52	9.6	9.38	7.68	6.25	4.8

When the 8-bit timer TMRAB is used to generate a serial clock, the baud rate is determined by the following equation:

$$\text{Baud rate} = \frac{\text{clock frequency selected by SYSCR0.PRCK[1:0]}}{\text{TABREG} \times \underline{2} \times 16}$$

└ When the TMRAB clock source is ϕT1

Note 1: I/O Interface mode cannot utilize the trigger output signal (internal) from the 8-bit timer TMRAB as a serial clock.

Note 2: This table assumes: fsys = fc, clock gear = fc/1, prescaler clock source = fperiph/4

14.2.3 Serial Clock Generator

This block generates a basic clock that controls the transmit and receive circuit.

- I/O Interface mode

  If the SCLK0 pin is configured as an output by clearing the SC0CR.IOC bit to 0, the output clock from the baud rate generator is divided by two to generate the basic clock. If the SCLK0 pin is configured as an input by setting the SC0CR.IOC bit to 1, the external SCLK0 clock is used as the basic clock; the SC0CR.SCLKS bit determines the active clock edge.

- UART mode

  The basic clock (SIOCLK) is selected from a clock produced by the baud rate generator, the system clock (f_{SYS}/2), the internal output signal from the 8-bit timer TMRAB, and the external SCLK0 clock, according to the setting of the SC0MOD0.SC[1:0] field.

14.2.4 Receive Counter

The receive counter is a 4-bit binary up-counter used in UART mode. This counter is clocked by SIOCLK. The receiver utilizes 16 clocks for each received bit, and oversamples each bit three times around their center (with 7th to 9th clocks). The value of a bit is determined by voting logic which takes the value of the majority of three samples.

14.2.5 Receive Controller

- I/O Interface mode

  If the SCLK0 pin is configured as an output by clearing the SC0CR.IOC bit to 0, the receive controller samples the RXD0 input at the rising edge of the shift clock driven out from the SCLK0 pin. If the SCLK0 pin is configured as an input by setting the SC0CR.IOC bit to 1, the receive controller samples the RXD0 input at either the rising or falling edge of the SCLK0 clock, as programmed in the SC0CR.SCLKS bit.

- UART mode

  The receive controller contains the start bit detection logic. Once a valid start bit is detected, the receive controller begins sampling the incoming data streams.

14.2.6 Receive Buffer

The receive buffer is double-buffered to prevent overrun errors. Received data is serially shifted bit by bit into Receive Buffer 1. When a whole frame is loaded into Receive Buffer 1, it is transferred to Receive Buffer 2 (SC0BUF), and a receive-done interrupt (INTRX0) is generated. At this time, the Receive Buffer Full flag (SC0MOD2.RBFLL) is set to 1, indicating that Receive Buffer 2 contains valid data.

The CPU reads a frame from Receive Buffer 2 (SC0BUF), causing the Receive Buffer Full flag (SC0MOD2.RBFLL) to be cleared to 0. Receive Buffer 1 can accept a new frame through the RXD0 pin before the CPU picks up the previous frame in Receive Buffer 2.

If the SCLK0 pin is configured as an output in I/O Interface mode, Receive Buffer 2 (SC0BUF) can be enabled or disabled by programming the WBUF bit in the SC0MOD2. Disabling Receive Buffer 2 (double-buffering) enables handshaking during data transfer; the SIO0 stops outputting the SCLK0 clock every time a single frame has transmitted. In this case, the CPU reads a frame from Receive Buffer 1, causing the output of the SCLK0 clock to be restarted. If Receive Buffer 2 (double-buffering) is enabled, a received frame is transferred from Receive Buffer 1 to Receive Buffer 2. Once a next frame is received, resulting in both Receive Buffers 1 and 2 containing valid data, the SIO0 stops outputting the SCLK0 clock. When the CPU reads a frame from Receive Buffer 2, the frame stored in Receive Buffer 1 is transferred to Receive Buffer 2, causing a receive-done interrupt (INTRX0) to occur and the SIO0 to restart outputting the SCLK0 clock. Consequently, no overrun error occurs if the SCLK0 pin is configured as an output in I/O Interface mode, regardless of the setting of the SC0MOD2.WBUF bit.

Note: In this mode, the OEER flag in the SC0CR has no meaning; it is read as undefined. When exiting SCLK output mode, first read the SC0CR to initialize this flag.

In other operating modes, Receive Buffer 2 is always enabled to improve performance during continuous transfer. However, the CPU must read Receive Buffer 2 before Receive Buffer 1 is filled with a new frame. Otherwise, an overrun error occurs, causing the frame previously in Receive Buffer 1 to be lost. Even in that case, the contents of Receive Buffer 2 and the SC0CR.RB8 bit are preserved.

The SC0CR.RB8 bit holds the parity bit for an 8-bit UART frame and the most significant bit for a 9-bit UART frame.

In 9-bit UART mode, the receiver wake-up feature allows the slave station in a multidrop system to wake up whenever an address frame is received. Setting the SC0MOD0.WU bit enables the wake-up feature. The receiver generates the INTRX0 interrupt only when the SC0CR.RB8 bit is set to 1.

14.2.7 Transmit Counter

The transmit counter is a 4-bit binary up-counter used in UART mode. Like the receive counter, the transmit counter is also clocked by SIOCLK. The transmitter generates a transmit clock (TXDCLK) pulse every 16 SIOCLK pulses.

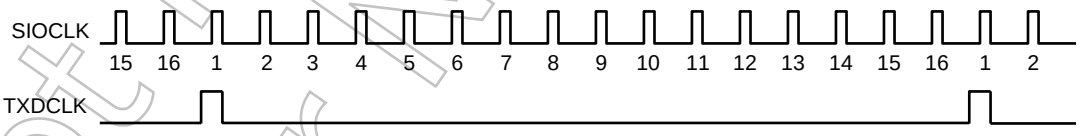


Figure 14.3 Transmit Clock Generation

14.2.8 Transmit Controller

- I/O Interface mode

  If the SCLK0 pin is configured as an output by clearing the SC0CR.IOC bit to 0, the transmit controller shifts out each bit in the transmit buffer to the TXD0 pin at the rising edge of the shift clock driven out on the SCLK0 pin. If the SCLK0 pin is configured as an input by setting the SC0CR.IOC bit to 1, the transmit controller shifts out each bit in the transmit buffer to the TXD0 pin at either the rising or falling edge of the SCLK0 input, as programmed in the SC0CR.SCLKS bit.

- UART mode

  Once the CPU loads a frame into the transmit buffer, the transmit controller begins transmission at the next rising edge of TXDCLK, producing a transmit shift clock (TXDSFT).

  Handshaking

  The SIO0 has the clear-to-send (CTS) pin. If the CTS operation is enabled, the CTS input must be low in order for the frame to be transmitted. This feature can be used for flow control to prevent overrun in the receiver. The SC0MOD.CTSE bit enables and disables the CTS operation.

  If the CTS pin goes high in the middle of a transmission, the transmit controller stops transmission upon completion of the current frame until CTS again goes low. If so enabled, the transmit controller generates the INTTX0 interrupt to notify the CPU that the transmit buffer is empty. After the CPU loads the next frame into the transmit buffer, the transmit controller remains in idle state until it detects CTS going low.

  Although the SIO0 does not have an RTS pin, any general-purpose port pin can serve as the RTS pin. The receiving device uses the RTS output to control the CTS input of the transmitting device. Once the receiving device has received a frame, RTS should be set to high in the receive-done interrupt handler to temporarily stop the transmitting device from sending the next frame. This way, the user can easily implement a two-way handshake protocol.

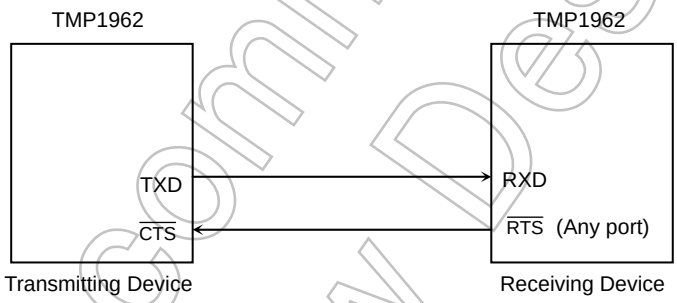
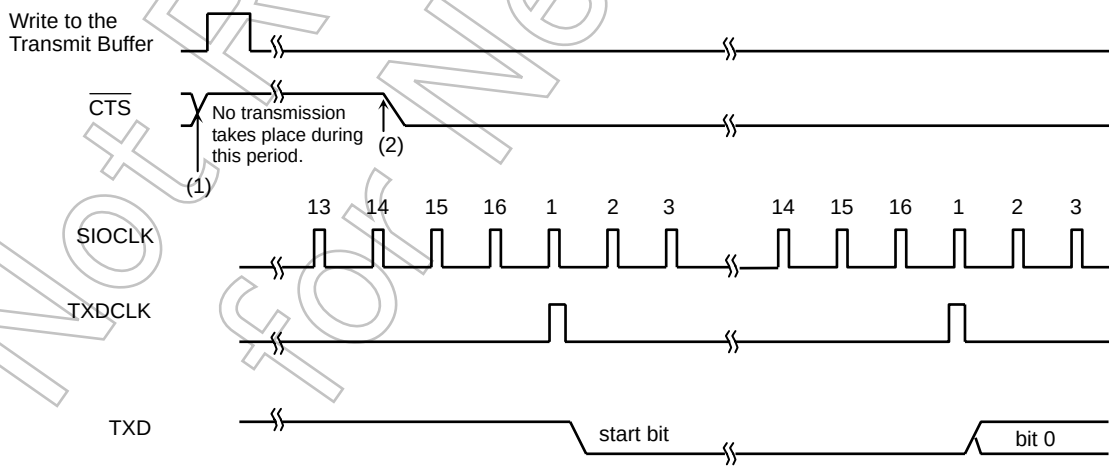


Figure 14.4 Handshaking Signals



Note: (1) When CTS goes high in the middle of transmission, the transmitter stops transmission after the current frame has been sent.

(2) The transmitter starts transmission at the first falling edge of the TXDCLK clock after the CTS signal goes low.

Figure 14.5 Clear-To-Send ($\overline{CTS}$) Signal Timing

14.2.9 Transmit Buffer

The transmit buffer is double-buffered. Double-buffering can be enabled or disabled by programming the WBUF bit in the SC0MOD2. If double-buffering is enabled, a frame is first written to Transmit Buffer 2 (SC0BUF) and then transferred to Transmit Buffer 1 (shift register), causing the INTTX interrupt to occur and the Transfer Buffer Empty flag (SC0MOD2.TBEMP) to be set. This flag indicates that Transfer Buffer 2 is empty and a next transmit frame can be written. Writing a next frame to Transmit Buffer 2 clears the TBEMP flag.

When the SCLK0 pin is configured as an input in I/O Interface mode, an underrun error occurs upon the completion of transmitting a frame from Transmit Buffer 1, if a next frame is not written to Transfer Buffer 2 before the clock pulse for the next frame is input. An underrun error is indicated by the parity/underrun flag (PERR) in the SC0CR. When the SCLK0 pin is configured as an output in I/O Interface mode, the SIO0 stops outputting the SCLK0 clock after transmitting a frame which has been transferred from Transmit Buffer 2 to Transmit Buffer 1. In this mode, therefore, no underrun error occurs.

Note: When the SCLK0 pin is configured as an output in I/O Interface mode, the PEER flag in the SC0CR has no meaning; it is read as undefined. When exiting SCLK output mode, first read the SC0CR to initialize this flag.

If double-buffering is disabled, the CPU writes a transmit frame to Transmit Buffer 1. The INTTX interrupt is generated upon the completion of transmission.

If handshaking is required, Transmit Buffer 2 must be disabled by clearing the WBUF bit in the SC0MOD2. For continuous transmission without handshaking, Transmit Buffer 2 can be enabled, by setting the WBUF bit, to improve performance.

14.2.10 Parity Controller

For transmit operations, setting the SC0CR.PE bit enables parity generation in 7- and 8-bit UART modes. The SC0CR.EVEN bit selects either even or odd parity.

If enabled, the parity controller automatically generates parity for the frame in the transmit buffer (SC0BUF). In 7-bit UART mode, the TB7 bit in the SC0BUF holds the parity bit. In 8-bit UART mode, the TB8 bit in the SC0MOD holds the parity bit. The parity bit is set after the frame has been transmitted. The SC0CR.PE and SC0CR.EVEN bits must be programmed prior to a write to the transmit buffer.

For receive operations, the parity controller automatically computes the expected parity when a frame in Receive Buffer 1 is transferred to Receive Buffer 2 (SC0BUF). The received parity bit is compared to the SC0BUF.RB7 bit in 7-bit UART mode and to the SC0CR.RB8 bit in 8-bit UART mode. If a frame is received with incorrect parity, the SC0CR.PERR bit is set.

In I/O Interface mode, the SC0CR.PERR bit indicates an underrun error rather than a parity error.

14.2.11 Error Flags

The SIO0 has the following error flag bits that indicate the status of the received frame for improved data reception reliability.

(1) Overrun error (OERR): Bit 4 of the SC0CR

In UART and I/O Interface modes, an overrun error is reported with the OERR bit set to 1 if all bits of a new frame are received before the CPU reads the current frame from the receive buffer. Reading the flag causes it to be cleared. When the SCLK0 pin is configured as an output in I/O Interface mode, however, no overrun error occurs so that the OEER flag has no meaning and is read as undefined.

(2) Parity error/underrun error (PERR): Bit 3 of the SC0CR

In UART mode, this flag indicates whether a parity error has occurred. A parity error is reported when the parity bit attached to a received frame does not match the expected parity computed from the frame. Reading the flag causes it to be cleared.

In I/O Interface mode, this flag indicates whether an underrun error has occurred, only when double-buffering (Transmit Buffer 2) is enabled (SC0MOD2.WBUF = 1) with the SCLK0 pin configured as an input. An underrun error is reported upon the completion of transmitting a frame from Transmit Buffer 1, if a next frame is not written to Transfer Buffer 2 before the clock pulse for the next frame is input. When the SCLK0 pin is configured as an output, no underrun error occurs so that the PEER flag has no meaning and is read as undefined. Reading the flag causes it to be cleared.

(3) Framing error (FERR): Bit 2 of the SC0CR

In UART mode, this flag indicates whether a framing error has occurred. A framing error is reported when a 0 is detected where a stop bit was expected. (The middle three of the 16 samples are used to determine the bit value.) Reading the flag causes it to be cleared. During reception, only a single stop bit is detected regardless of the setting of the SBLEN bit in Serial Mode Control Register 2 (SC0MOD2).

Operating Mode	Error Flag	Function
UART	OERR	Overrun error flag
	PERR	Parity error flag
	FERR	Framing error flag
I/O Interface (SCLK Input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O Interface (SCLK Output)	OERR	Undefined
	PERR	Undefined
	FERR	Fixed to 0

14.2.12 Bit Transfer Sequence

The DRCHG bit in Serial Mode Control Register 2 (SC0MOD2) determines whether the most significant bit (MSB) or least significant bit (LSB) is transmitted first in I/O Interface mode. The setting of the DRCHG bit cannot be modified while the SIO is transferring data.

14.2.13 Stop Bit Length

Bit 4 (SBLEN) in the SC0MOD2 register determines the number of stop bits (1 or 2) used in UART mode.

14.2.14 Status Flag

Bit 8 (RBFLL) in the SC0MOD2 register indicates whether Receive Buffer 2 is full (contains data) when double-buffering is enabled (SC0MOD2.WBUF = 1). It is set to 1 once a received frame is transferred from Receive Buffer 1 to Receive Buffer 2. The RBFLL bit is cleared to 0 when the CPU or DMAC reads data from Receive Buffer 2. When WBUF = 0, the RBFLL bit has no meaning; it should not be used as a status flag. Bit 7 (TBEMP) in the SC0MOD2 register indicates whether Transmit Buffer 2 is empty when double-buffering is enabled (SC0MOD2.WBUF = 1). It is set to 1 once a transmit frame is transferred from Transmit Buffer 2 to Transmit Buffer 1 (shift register). The TBEMP bit is cleared to 0 when the CPU or DMAC stores data in Transmit Buffer 2. When WBUF = 0, the TBEMP bit has no meaning; it should not be used as a status flag.

14.2.15 Transmit/Receive Buffer Configuration

		WBUF = 0	WBUF = 1
UART	Transmit	Single	Double
	Receive	Double	Double
I/O Interface (SCLK Input)	Transmit	Single	Double
	Receive	Double	Double
I/O Interface (SCLK Output)	Transmit	Single	Double
	Receive	Single	Double

14.2.16 Signal Generation Timing

(1) UART mode

Receive operation

Mode	9 Data Bits	8 Data Bits with Parity	8 Data Bits with No Parity, 7 Data Bits with Parity, 7 Data Bits with No Parity
Interrupt	Middle of the first stop bit	Middle of the first stop bit	Middle of the first stop bit
Framing Error	Middle of the stop bit	Middle of the stop bit	Middle of the stop bit
Parity Error	—	Middle of the last bit (i.e., parity bit)	Middle of the last bit (i.e., parity bit)
Overrun Error	Middle of the stop bit	Middle of the stop bit	Middle of the stop bit

Transmit operation

Mode	9 Data Bits	8 Data Bits with Parity	8 Data Bits with No Parity, 7 Data Bits with Parity, 7 Data Bits with No Parity
Interrupt (WBUF = 0)	Immediately before the stop bit is shifted out	Immediately before the stop bit is shifted out	Immediately before the stop bit is shifted out
Interrupt (WBUF = 1)	Immediately after the frame is transferred to Transmit Buffer 1 (i.e., immediately before the stop bit is shifted out)	Immediately after the frame is transferred to Transmit Buffer 1 (i.e., immediately before the stop bit is shifted out)	Immediately after the frame is transferred to Transmit Buffer 1 (i.e., immediately before the stop bit is shifted out)

(2) I/O Interface mode

Receive operation

Interrupt (WBUF = 0)	SCLK Output Mode	Immediately after the rising edge of the last SCLK pulse
	SCLK Input Mode	Immediately after the rising or falling edge of the last SCLK pulse, as programmed
Interrupt WBUF = 1)	SCLK Output Mode	Immediately after the rising edge of the last SCLK pulse (i.e., immediately after the frame is transferred to Receive Buffer 2) or immediately after the frame is read from Receive Buffer 2
	SCLK Input Mode	Immediately after rising or falling edge of the last SCLK pulse, as programmed (i.e., immediately after the frame is transferred to Receive Buffer 2)
Overrun Error	SCLK Input Mode	Immediately after the rising or falling edge of the last SCLK pulse, as programmed

Transmit operation

Interrupt (WBUF = 0)	SCLK Output Mode	Immediately after the rising edge of the last SCLK pulse
	SCLK Input Mode	Immediately after the rising or falling edge of the last SCLK pulse, as programmed
Interrupt (WBUF = 1)	SCLK Output Mode	Immediately after the rising edge of the last SCLK pulse or immediately after the frame is transferred to Transmit Buffer 1
	SCLK Input Mode	Immediately after rising or falling edge of the last SCLK pulse, as programmed or immediately after the frame is transferred to Transmit Buffer 1
Underrun Error	SCLK Input Mode	Immediately after the rising or falling edge of the next SCLK pulse, as programmed

Note 1: Do not modify any control register during transmit or receive operations.
Note 2: Do not disable receive operations by clearing the SC0MOD0.RXE bit while any data is being received.

14.3 Register Description (Channel 0)

SC0MOD0 (0xFFFF_F261)

	7	6	5	4	3	2	1	0
Bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Bit 8 of a transmitted character	0: Disables CTS operation 1: Enables CTS operation	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: I/O Interface mode 01: 7-Bit UART mode 10: 8-Bit UART mode 11: 9-Bit UART mode		Serial clock (for UART) 00: TA6TRG (timer) 01: Baud rate generator 10: Internal fSYS/2 clock 11: External clock (SCLK0 input)	

Note: In I/O Interface mode, the Serial Control Register (SC0CR) is used to select the clock source.

→ Wake-up function

	9-Bit UART Mode	Other Modes
0	Interrupt on every received frame	don't care
1	Interrupt only when RB8 = 1	

→ Handshake ($\overline{\text{CTS}}$) control

0	Disable (Accepts data streams at all times)
1	Enable

Note: First, ensure RXE is cleared to 0. Then, configure the mode registers (SC0MOD0, SC0MOD1 and SC0MOD2). Finally, set RXE to 1.

Figure 14.6 Serial Mode Control Register 0 (SC0MOD0, for SIO0)

SC0MOD1 (0xFFFF_F266)

	7	6	5	4	3	2	1	0
Bit Symbol	I2S0	FDPX0	SIOEN					
Read/Write	R/W	R/W	R/W					
Reset Value	0	0	0					
Function	IDLE 0: Off 1: On	Syn-chronous 0: Half-duplex 1: Full-duplex	SIO operation 0: Disable 1: Enable					

SIOEN: Enables or disables the supply of clock pulses to the SIO module, except for registers.

Note: When configuring the SC0MOD1 register, first set bit 5 (SIOEN) to 1 before programming other bits (I2SO and FDPX0).

Figure 14.7 Serial Mode Control Register 1 (SC0MOD1, for SIO0)

SC0MOD2 (0xFFFF_F265)

	7	6	5	4	3	2	1	0
Bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R/W						W	W
Reset Value	1	0	0	0	0	0	0	0
Function	Transmit buffer empty flag 0: Full 1: Empty	Receive buffer empty flag 0: Empty 1: Full	Trans-mission-in-progress flag 0: Stopped 1: n progress	Number of stop bits 0: 1 1: 2	Bit sequence 0: LSB first 1: MSB first	Double-buffering 0: Disable 1: Enable	Software reset A write of 10 followed by a write of 01	

SWRST[1:0]: A write of 10 followed by a write of 01 to this field resets the module, thus initializing the RXE bit in the SC0MOD0, the TBEMP, RBFLL and TXRUN bits in the SC0MOD2, the OERR, PERR and FERR bits in the SC0CR, and the internal circuits.

WBUF: Enables or disables double-buffering for transmit (SCLK output or input) or receive (SCLK output) operation in I/O Interface mode or transmit operation in UART mode. For any other operation, double-buffering is always enabled.

DRCHG: Specifies the bit transfer sequence in I/O Interface mode. In UART mode, the LSB is always transferred first.

TXRUN: A status flag indicating whether transmit shift operation is in progress. When this bit is set to 1, transmit operation is in progress. When this bit is cleared to 0, transmit operation is completed (if TBEMP = 1) or the transmit buffer contains a next frame and is ready for transmission (if TBEMP = 0).

RBFLL: A flag indicating whether Receive Buffer 2 is full. The RBFLL bit is set to 1 once a received frame is transferred from Receive Buffer 1 to Receive Buffer 2. It is cleared when the CPU or DMAC reads the frame. If double-buffering is disabled, the RBFLL bit has no meaning.

TBEMP: A flag indicating whether Transmit Buffer 2 is empty. The TBEMP bit is set to 1 once a frame is transferred from Transmit Buffer 2 to Transmit Buffer 1. It is cleared when the CPU or DMAC writes a next frame to Transmit Buffer 2. If double-buffering is disabled, the TBEMP bit has no meaning.

SBLEN: Specifies the number of transmit stop bits in UART mode. For receive operation, a single stop bit is used regardless of the setting of this bit.

Note: If the module needs to be reset while it is transmitting data, two consecutive software reset sequences (i.e., 10, 01, 10, 01) must be programmed.

Figure 14.8 Serial Mode Control Register 2 (SC0MOD2, for SIO0)

SC0CR (0xFFFF_F262)

	7	6	5	4	3	2	1	0
Bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared when read)			R/W	
Reset Value		0	0	0	0	0	0	0
Function	Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred.			0: SCLK0 ⎍ 1: SCLK0 ⎍	0: Baud rate generator 1: SCLK0 input
				Overrun	Parity/ Underrun	Framing		

Input clock in I/O Interface mode

0	Baud rate generator
1	SCLK0 input

Active edge for the SCLK0 input

0	Data is transmitted/received on the SCLK0 rising edge. ⎍
1	Data is transmitted/received on the SCLK0 falling edge. ⎍

Framing error flag
Parity error/underrun error flag
Overrun error flag
} These bits are cleared to 0 when read.

Parity type

0	Odd parity
1	Even parity

Note 1: The SCLKS bit must be set to 0 for the SCLK0 output.
Note 2: All error flags are cleared to 0 when read.

Figure 14.9 Serial Control Register (SC0CR, for SIO0)

BR0CR (0xFFFF_F260)

	7	6	5	4	3	2	1	0
Bit Symbol	—	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Must be written as 0.	N + (16 - K)/16 function 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Clock divisor value N			

Clock source for baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

BR0ADD (0xFFFF_F267)

	7	6	5	4	3	2	1	0
Bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	R/W							
Reset Value					0	0	0	0
Function	Value of K in N + (16 - K)/16							

Clock divisor value for baud rate generator

	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0
BR0ADD <BR0K3:0> \ BR0CR <BR0S3:0>	0000(N = 16) to 0001(N = 1)	0010(N = 2) to 1111(N = 15)	0001(N = 1) (ONLY UART) to 1111 (N = 15) 0000 (N = 16)
0000	Disabled	Disabled	
0001(K = 1) to 1111(K = 15)	Disabled	Divided by N + (16 - K)/16	Divided by N

Note 1: The baud rate generator divisor cannot be set to 1 in UART mode if the N + (16 - K)/16 clock division function is enabled. In I/O Interface mode, the baud rate generator divisor can be set to 1 only when double-buffering is enabled.

Note 2: To use the N + (16 - K)/16 clock division function, the value of K must be programmed in the BR0ADD.BR0K[3:0] field before setting BR0CR.BR0ADDE to 1. However, the N + (16 - K)/16 clock division function is not usable when BR0CR.BR0S[3:0] = 0000 (N = 16) or 0001 (N = 1).

Note 3: The N + (16 - K)/16 clock division function can only be used in UART mode. In I/O Interface mode, this must be disabled by clearing BR0CR.BR0ADDE to 0.

Figure 14.10 Baud Rate Generator Control Registers (BR0CR and BR0ADD, for SIO0)

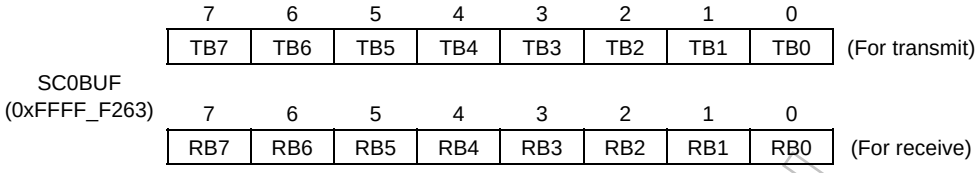


Figure 14.11 Serial Transmit/Receive Buffer Register (SC0BUF, for SIO0)

14.4 Operating Modes

14.4.1 Mode 0 (I/O Interface Mode)

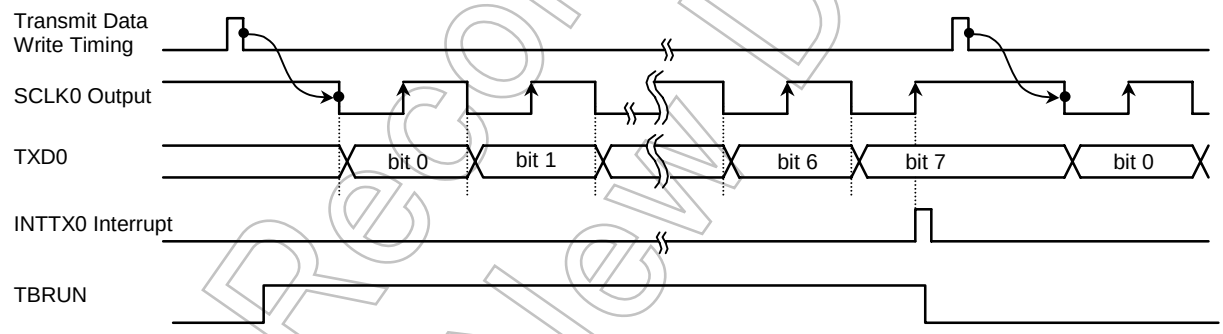
Mode 0 utilizes a synchronization clock (SCLK), which can be configured for either Output mode in which the SCLK clock is driven out from the TMP1962 or Input mode in which the SCLK clock is supplied externally.

(1) Transmit operations

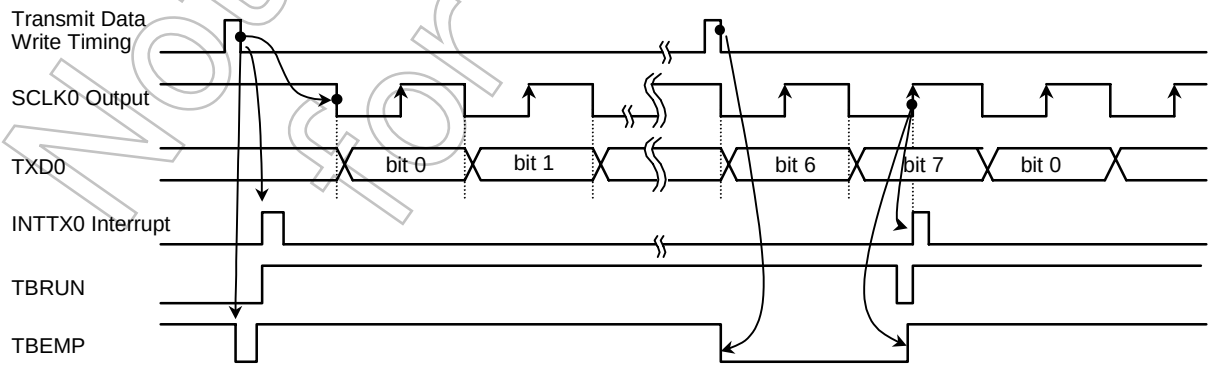
SCLK Output mode

When transmit double-buffering is disabled (SC0MOD2.WBUF = 0) in SCLK Output mode, each time the CPU writes a frame to the transmit buffer, the eight bits of the frame is shifted out on the TXD0 pin, and the synchronization clock is driven out from the SCLK0 pin. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated.

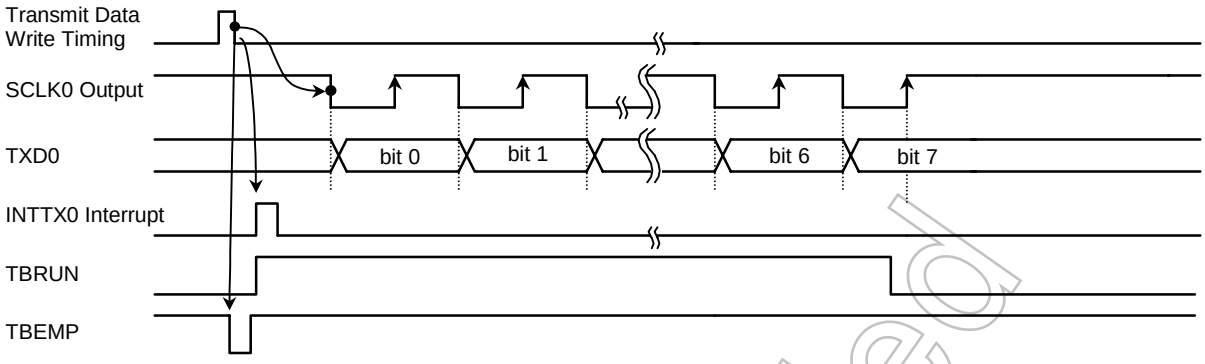
When transmit double-buffering is enabled (SC0MOD2.WBUF = 1), a frame is transferred from Transmit Buffer 2 to Transmit Buffer 1 (shift register) once the CPU writes the frame to Transmit Buffer 2 when the SIO0 is not transmitting any data or once the last frame in Transmit Buffer 1 has been sent. At this time, the transmit buffer empty flag (SC0MOD2.TBEMP) is set to 1 and the INTTX0 is generated. If there is no data to be transferred from Transmit Buffer 2 to Transmit Buffer 1, however, SCLK0 output is stopped without generating the INTTX0 interrupt.



When WBUF = 0 (double-buffering disabled)



When WBUF = 1 (double-buffering enabled) and Transmit Buffer 2 contains data



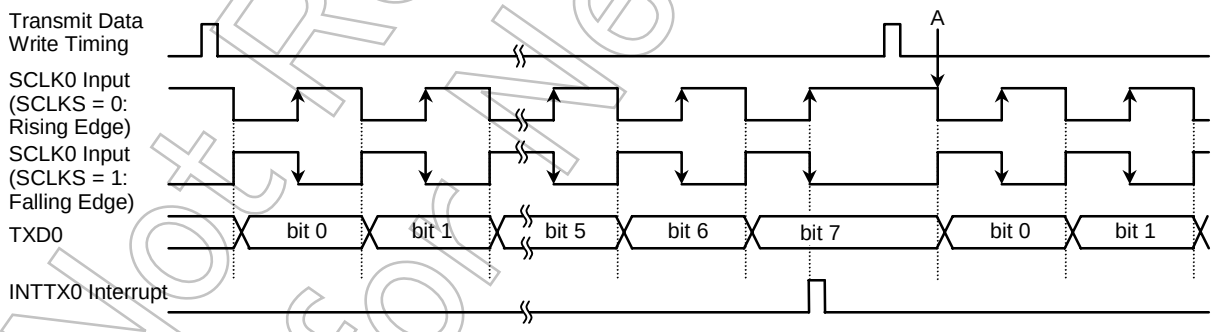
When WBUF = 1 (double-buffering enabled) but Transmit Buffer 2 does not contain data

Figure 14.12 Transmit Operation in I/O Interface Mode (SCLK0 Output Mode)

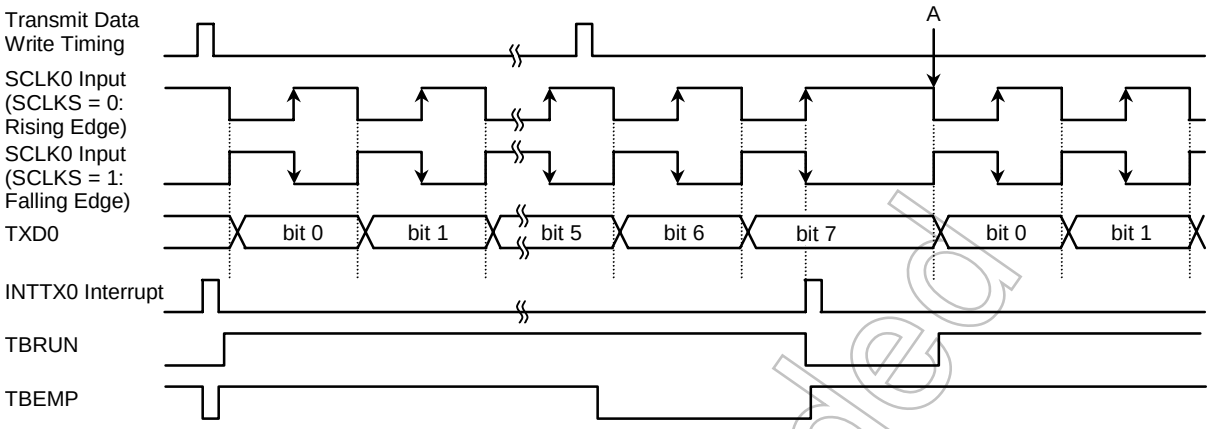
SCLK Input mode

When transmit double-buffering is disabled (SC0MOD2.WBUF = 0) in SCLK Input mode, the CPU must write a frame to the transmit buffer before the SCLK0 input is activated. The eight bits of a frame in the transmit buffer are shifted out on the TXD0 pin, synchronous to the programmed edge of the SCLK0 input. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated. The CPU must load the next frame into the transmit buffer by point A (shown in the figure below).

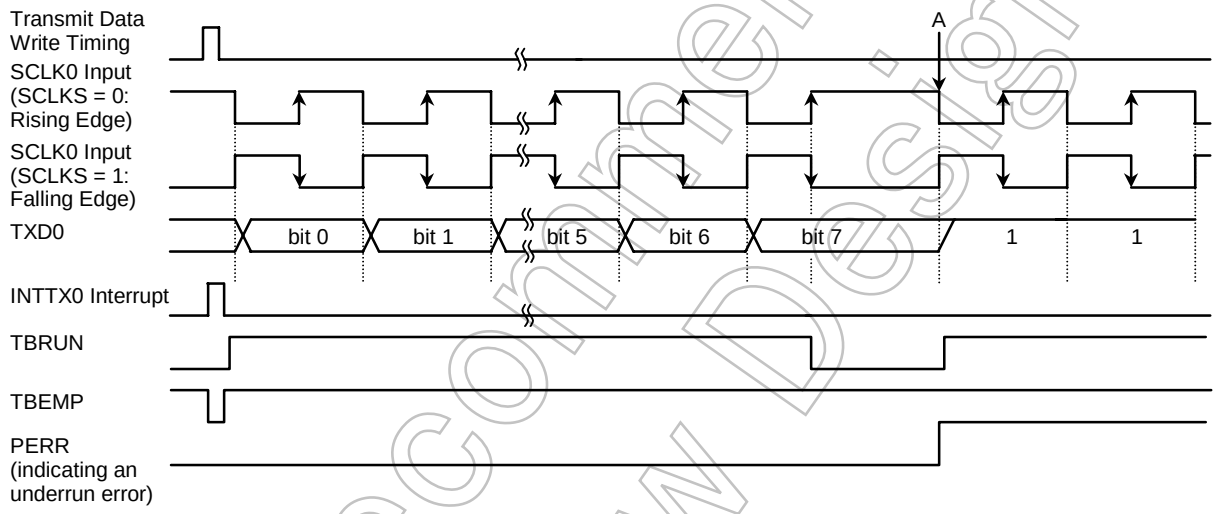
When transmit double-buffering is enabled (SC0MOD2.WBUF = 1), a frame is transferred from Transmit Buffer 2 to Transmit Buffer 1 (shift register) once the CPU writes the frame to Transmit Buffer 2 before the SCLK0 input is activated or once the last frame in Transmit Buffer 1 has been sent. At this time, the transmit buffer empty flag (SC0MOD2.TBEMP) is set to 1 and the INTTX0 interrupt is generated. If the SCLK0 input is activated before a frame is written to Transmit Buffer 2, however, the SIO0 assumes an underrun error and sends eight bits of dummy data (FFh) although the internal bit counter starts counting.



When WBUF = 0 (double-buffering disabled)



When WBUF = 1 (double-buffering enabled) and Transmit Buffer 2 contains data



When WBUF = 1 (double-buffering enabled) but Transmit Buffer 2 does not contain data

Figure 14.13 Transmit Operation in I/O Interface Mode (SCLK0 Input Mode)

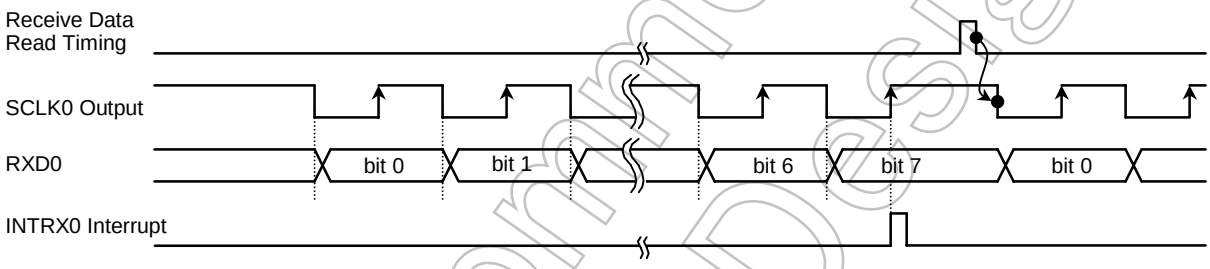
(2) Receive operations

SCLK Output mode

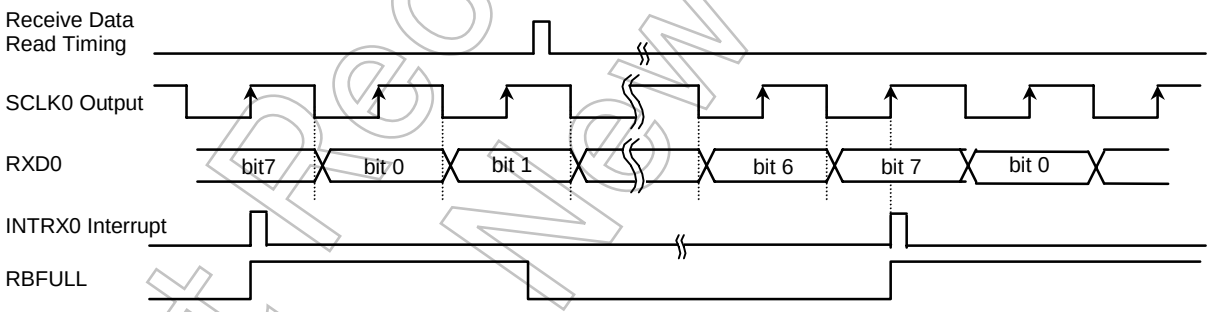
When receive double-buffering is disabled (SC0MOD2.WBUF = 0) in SCLK Output mode, each time the CPU picks up the frame in Receive Buffer 1, the synchronization clock is driven out from the SCLK0 pin to shift the next frame into Receive Buffer 1. When a whole 8-bit frame has been loaded into Receive Buffer 1, the INTRX0 interrupt is generated.

The SCLK output is initiated by setting the SC0MOD0.RXE bit to 1. When receive double-buffering is enabled (SC0MOD2.WBUF = 1), the frame received first is transferred to Receive Buffer 2 and then a next frame is received into Receive Buffer 1. Once a frame is transferred from Receive Buffer 1 to Receive Buffer 2, the receive buffer full flag (SC0MOD2.RBFLL) is set to 1 and the INTRX0 interrupt is generated.

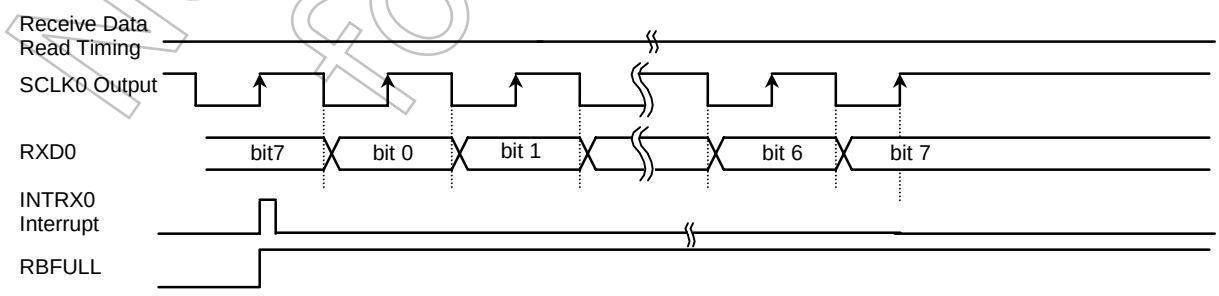
After a frame has been transferred to Receive Buffer 2, the CPU or DMAC should read it before all eight bits of a next frame are received. Otherwise, the INTRX0 interrupt is not generated and SCLK0 output is stopped. In that state, when the CPU or DMAC reads the frame from Receive Buffer 2, the next frame is transferred from Receive Buffer 1 to Receive Buffer 2, generating the INTRX0 interrupt to restart receive operation.



When WBUF = 0 (double-buffering disabled)



When WBUF = 1 (double-buffering enabled) and Receive Buffer 2 is read



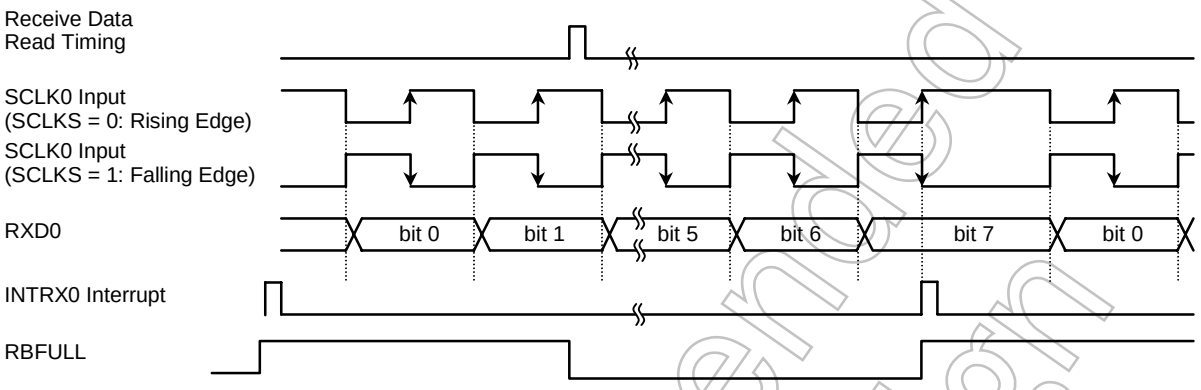
When WBUF = 1 (double-buffering enabled) but Receive Buffer 2 is not read

Figure 14.14 Receive Operation in I/O Interface Mode (SCLK0 Output Mode)

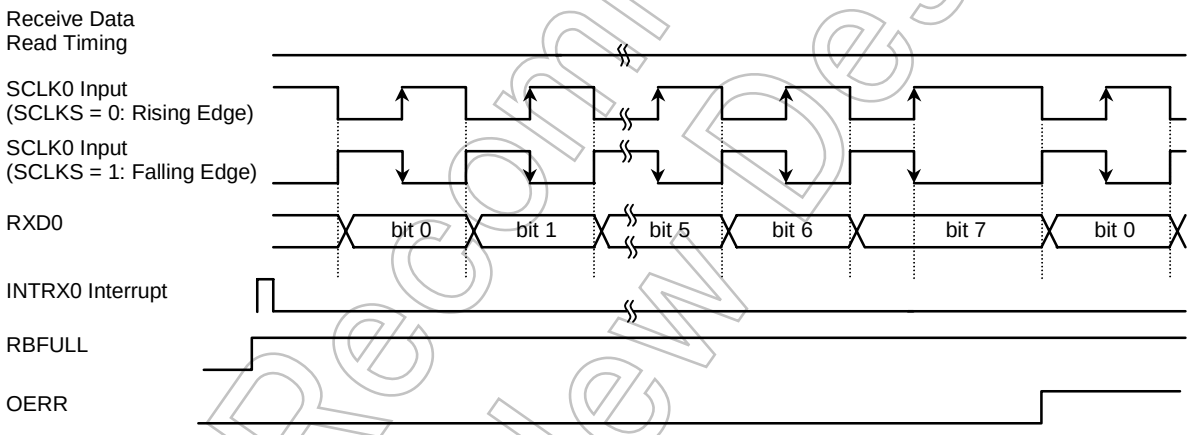
SCLK Input mode

In SCLK Input mode, receive double-buffering is always enabled. A received frame is transferred to Receive Buffer 2 so that a next frame can be received continuously into Receive Buffer 1.

The INTRX0 interrupt is generated every time a frame is transferred from Receive Buffer 1 to Receive Buffer 2.



When Receive Buffer 2 is read



When Receive Buffer 2 is not read

Figure 14.15 Receive Operation in I/O Interface Mode (SCLK0 Input Mode)

Note: Regardless of whether SCLK0 is in Input mode or Output mode, the receiver must be enabled by setting the SC0MOD0.RXE bit to 1 in order to perform receive operations.

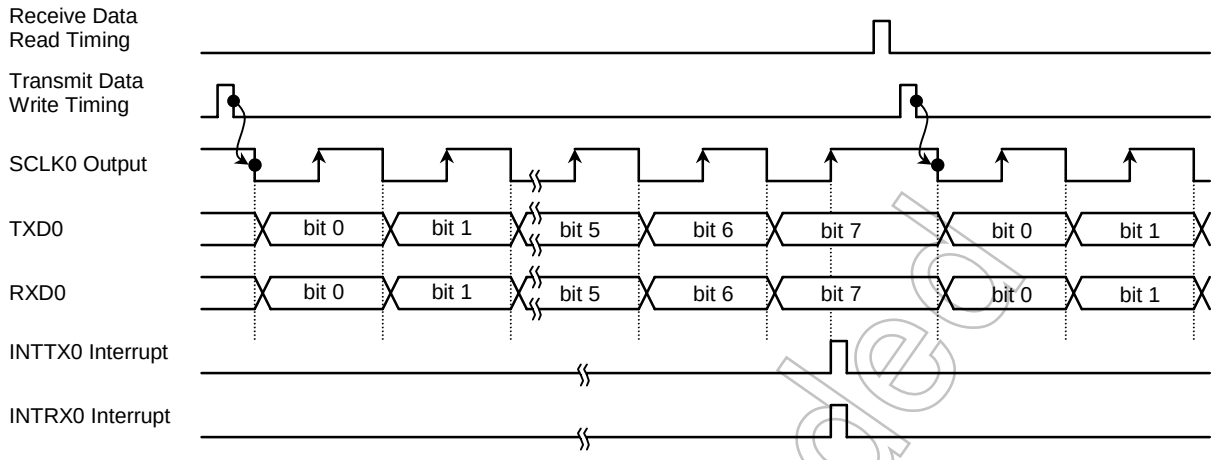
(3) Full-duplex transmit/receive operations

Setting bit 6 (FDPX0) in Serial Control Register 1 (SC0MOD1) to 1 enables full-duplex communication.

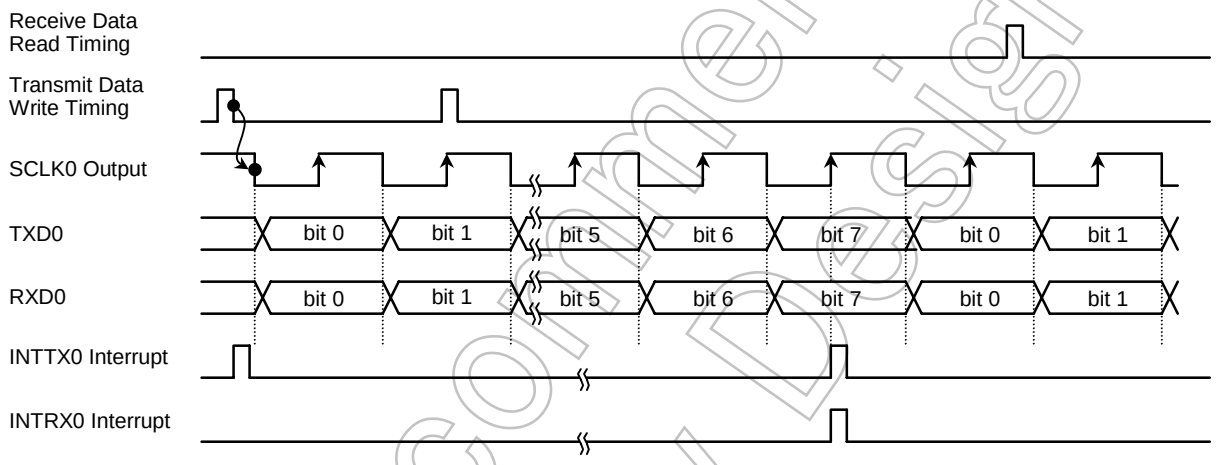
SCLK Output mode

When transmit/receive double-buffering is disabled (SC0MOD2.WBUF = 0) in SCLK Output mode, each time the CPU writes a frame to the transmit buffer, the synchronization clock is driven out from the SCLK0 pin to shift an 8-bit frame into Receive Buffer 1, generating the INTRX0 interrupt. At the same time, the frame written to the transmit buffer is shifted out on the TXD0 pin. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated and SCLK0 output is stopped. When the CPU subsequently picks up the frame in the receive buffer and writes a next frame to the transmit buffer, next transmit/receive operation starts, regardless of whether the CPU first reads the receive buffer or it first writes data to the transmit buffer.

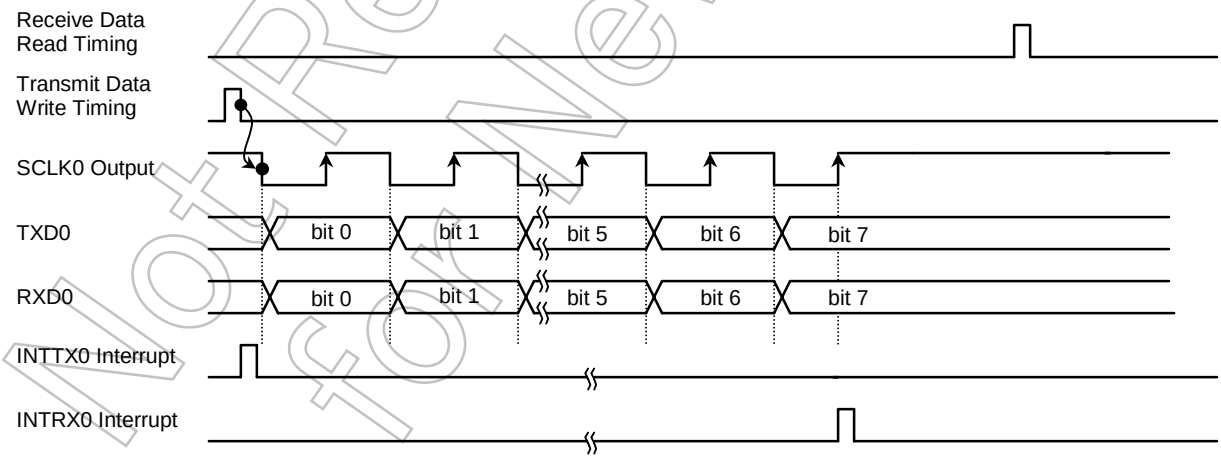
When transmit/receive double-buffering is enabled (SC0MOD2.WBUF = 1), each time the CPU writes a frame to Transmit Buffer 2, the synchronization clock is driven out from the SCLK0 pin to shift an 8-bit frame into Receive Buffer 1; it is then transferred to Receive Buffer 2, generating the INTRX0 interrupt. At the same time, the frame stored in Transmit Buffer 1 is shifted out on the TXD0 pin. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated and the next frame is transferred from Transmit Buffer 2 to Transmit Buffer 1. During the above sequence, SCLK0 output is stopped if Transmit Buffer 2 does not contain data (SC0MOD2.TBEMP = 1) or if Receive Buffer 2 still contains data (SC0MOD2.RBFLL = 1). When the CPU subsequently picks up the frame in Receive Buffer 2 and writes a next frame to Transmit Buffer 2, SCLK0 output is restarted so that next transmit/receive operation starts.



When WBUF = 0 (double-buffering disabled)



When WBUF = 1 (double-buffering enabled) and no error occurs



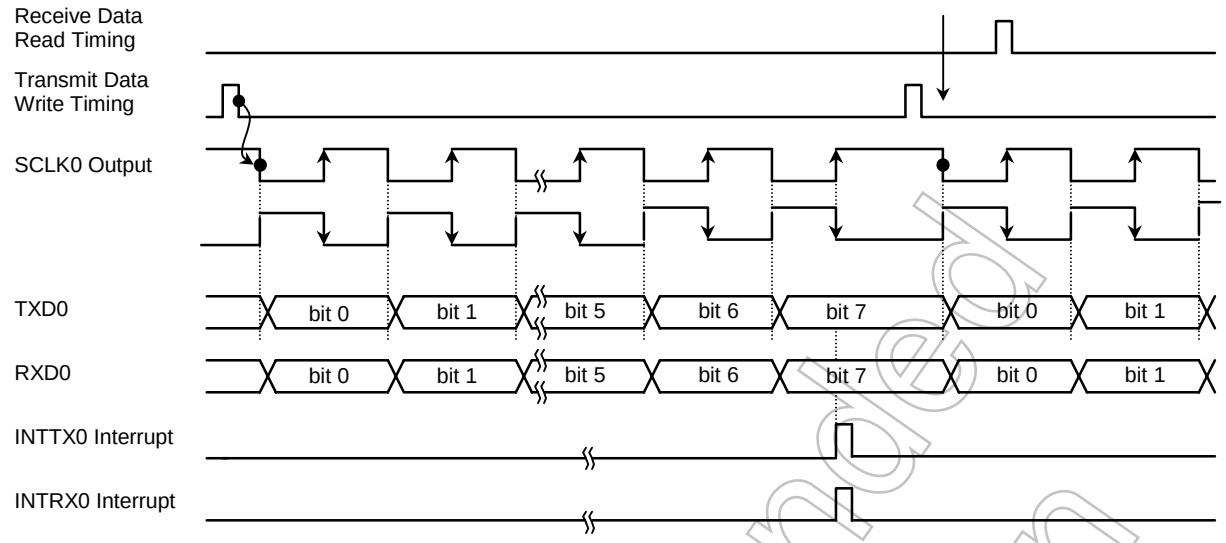
When WBUF = 1 (double-buffering enabled) and an error occurs

Figure 14.16 Full-Duplex Transmit/Receive Operation in I/O Interface Mode (SCLK0 Output Mode)

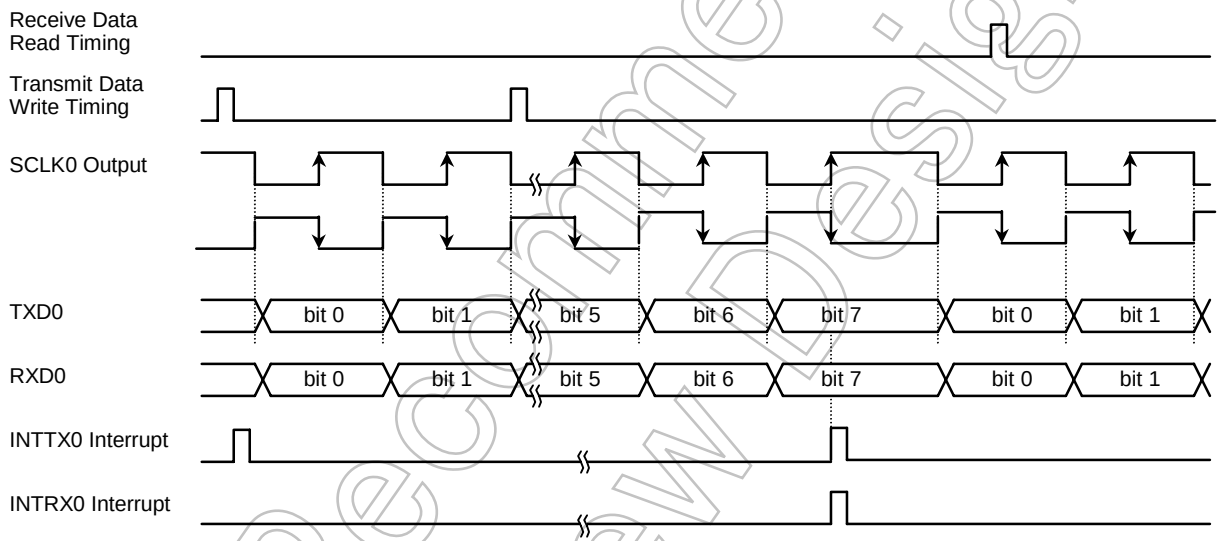
SCLK Input mode

When transmit double-buffering is disabled (SC0MOD2.WBUF = 0) in SCLK Input mode (receive double-buffering is always enabled in this mode), the CPU must write a frame to the transmit buffer before the SCLK0 input is activated. The eight bits of a frame in the transmit buffer are shifted out on the TXD0 pin, and the eight bits of a received frame are shifted into Receive Buffer 1, synchronous to the programmed edge of the SCLK0 input. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated. When all the bits have been received, the frame is transferred from Receive Buffer 1 to Receive Buffer 2, generating the INTRX0 interrupt. The CPU must load the next frame into the transmit buffer by point A (shown in the figure below). The CPU must also pick up the frame in Receive Buffer 2 before a next frame has been received.

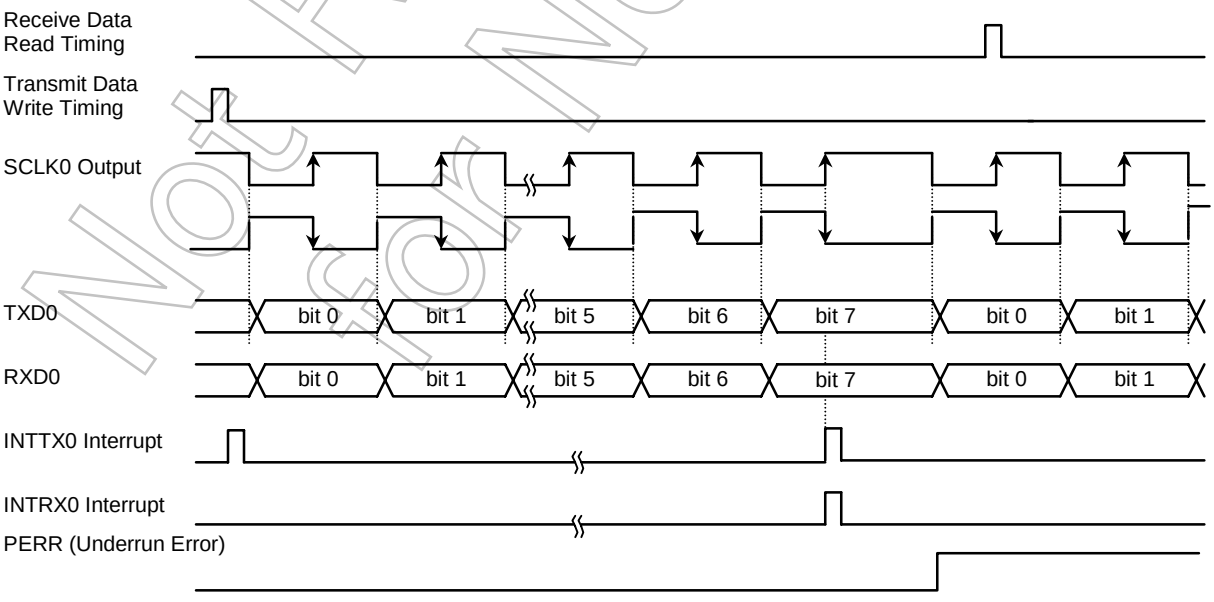
When transmit/receive double-buffering is enabled (SC0MOD2.WBUF = 1), a frame is transferred from Transmit Buffer 2 to Transmit Buffer 1 once the last frame in Transmit Buffer 1 has been sent. At this time, the INTTX0 interrupt is generated. When the 8-bit frame, received in parallel with transmission, has been shifted into Receive Buffer 1, it is transferred to Receive Buffer 2, generating the INTRX0 interrupt. When the SLCK0 is subsequently activated, the frame stored in Transmit Buffer 1 is shifted out while a next frame is received into Receive Buffer 1. If the CPU does not read the frame from Receive Buffer 2 before the last bit of a next frame is received, an overrun error occurs. If the CPU does not write a frame to Transmit Buffer 2 before the SCLK0 input is subsequently activated, an underrun error occurs.



When WBUF = 0 (double-buffering disabled)



When WBUF = 1 (double-buffering enabled) and no error occurs



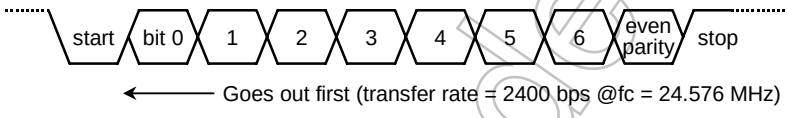
When WBUF = 1 (double-buffering enabled) and an error occurs

Figure 14.17 Full-Duplex Transmit/Receive Operation in I/O Interface Mode (SCLK0 Input Mode)

14.4.2 Mode 1 (7-Bit UART Mode)

Setting the SM[1:0] field in the SC0MOD0 to 01 puts the SIO0 in 7-Bit UART mode. In this mode of operation, the parity bit can be added to the transmitted frame, and the receiver can perform a parity check on incoming data. Parity can be enabled and disabled through the programming of the PE bit in the SC0CR. When PE = 1, the SCR0CR.EVEN bit selects even or odd parity. The SBLEN bit in the SC0MOD2 specifies the number of stop bits.

Example: Transmitting 7-bit UART frames with an even-parity bit



← Goes out first (transfer rate = 2400 bps @fc = 24.576 MHz)

Clocking conditions:
- System clock: High-speed (fc)
- High-speed clock gear: x1 (fc)
- Prescaler clock: $f_{periph}/4$ ($f_{periph} = f_{sys}$)

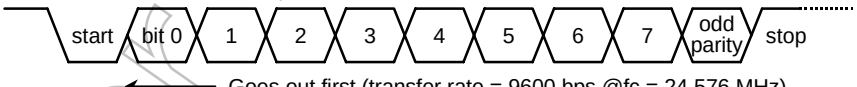
	7	6	5	4	3	2	1	0	
PGCR	← –	–	–	–	–	–	1	–	Configures the PG1 pin as TXD0.
PGFC	← –	–	–	–	–	–	1	–	
SC0MOD	← X	0	–	X	0	1	0	1	Selects 7-Bit UART mode.
SC0CR	← X	1	1	X	X	X	0	0	Selects even parity.
BR0CR	← 0	0	1	0	1	0	1	0	Sets the transfer rate to 2400 bps.
IMC4LH	← –	–	1	1	0	1	0	0	Enables the INTTX0 interrupt and sets its priority level to 4.
SC0BUF	← *	*	*	*	*	*	*	*	Loads the transmit buffer with a frame.

Note: X = Don't care, - = No change

14.4.3 Mode 2 (8-Bit UART Mode)

Setting the SM[1:0] field in the SC0MOD0 to 10 puts the SIO0 in 8-Bit UART mode. In this mode of operation, the parity bit can be added to the transmitted frame, and the receiver can perform a parity check on incoming data. Parity can be enabled and disabled through the programming of the PE bit in the SC0CR. When PE = 1, the SCR0CR.EVEN bit selects even or odd parity.

Example: Transmitting 8-bit UART frames with an odd-parity bit



← Goes out first (transfer rate = 9600 bps @fc = 24.576 MHz)

Clocking conditions:
- System clock: High-speed (fc)
- High-speed clock gear: x1 (fc)
- Prescaler clock: $f_{periph}/4$ ($f_{periph} = f_{sys}$)

- Settings in the main routine

		7	6	5	4	3	2	1	0	
PGCR	←	–	–	–	–	–	0	–	–	Configures the PG2 pin as RXD0.
PGFC	←	–	–	–	–	–	1	–	–	
SC0MOD	←	–	0	0	X	1	0	0	1	Selects 8-Bit UART mode.
SC0CR	←	X	0	1	X	X	X	0	0	Selects odd parity.
BR0CR	←	0	0	0	1	0	1	0	1	Sets the transfer rate to 9600 bps.
IMC4LL	←	–	–	1	1	0	1	0	0	Enables the INTRX0 interrupt and sets its priority level to 4.
SC0MOD	←	–	–	1	X	–	–	–	–	Enables reception.

- Example of interrupt routine processing

INTCLR	←	X X 0 1 0 0 0 0	Clears the interrupt request.
Reg.	←	SC0CR AND 0x1C	Checks for errors.
if Reg.	≠	0 then Error	
Reg.	←	SC0BUF	Reads received data.
End of interrupt processing			

Note: X = Don't care, - = No change

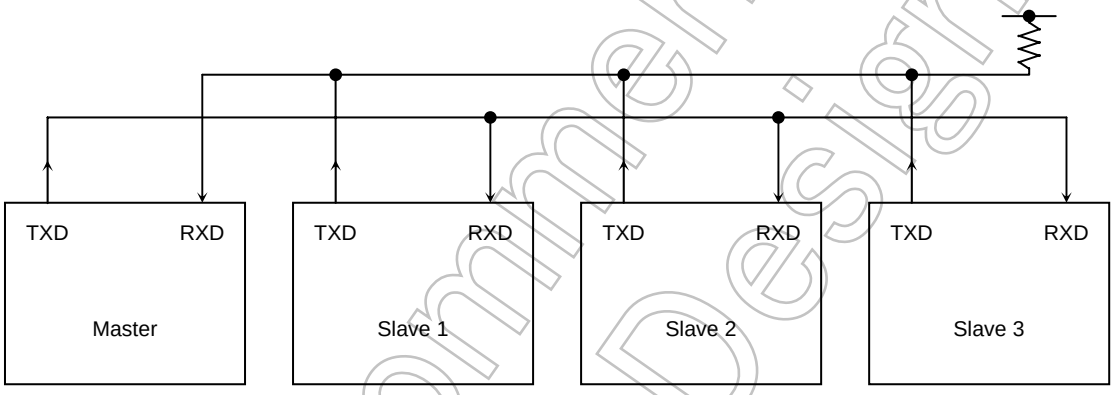
14.4.4 Mode 3 (9-Bit UART Mode)

Setting the SM[1:0] field in the SC0MOD0 to 11 puts the SIO0 in 9-Bit UART mode. In this mode, a parity bit cannot be used; thus, parity should be disabled by clearing the SC0CR.PE bit to 0.

For transmit operations, the most-significant bit (9th bit) is stored in bit 7 (TB8) in the SC0MOD0. For receive operations, the most-significant bit is stored in bit 7 (RB8) in the SC0CR. Reads and writes of the transmit/receive frame must be done with the most-significant bit first, followed by the SC0BUF. The SBLEN bit in the SC0MOD2 specifies the number of stop bits.

Wake-up Feature

In 9-Bit UART mode, the receiver wake-up feature allows the slave station in a multidrop system to wake up whenever an address frame is received. Setting the SC0MOD0.WU bit enables the wake-up feature. When the SC0CR.RB8 bit has received an address/data flag bit set to 1, the receiver generates the INTRX0 interrupt.



Note: The slave controller's TXD pin must be configured as an open-drain output by programming the ODE register.

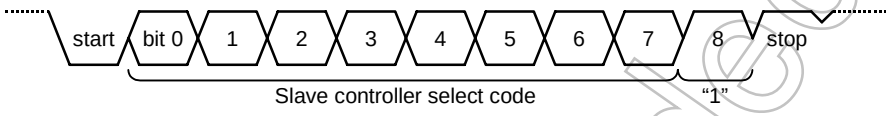
Figure 14.18 Serial Link Using the Wake-Up Function

Protocol

(1) Put all the master and slave controllers in 9-Bit UART mode.

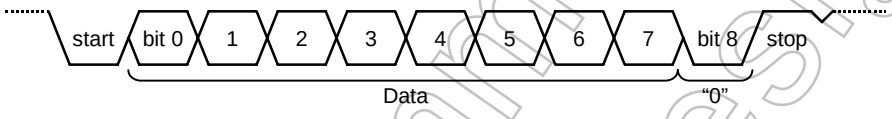
(2) Enables the receiver in each slave controller by setting the SC0MOD0.WU bit to 1.

(3) The master controller transmits an 8-bit address frame (i.e, select code) that identifies a slave controller. The address character has the most-significant bit (bit 8) set to 1.



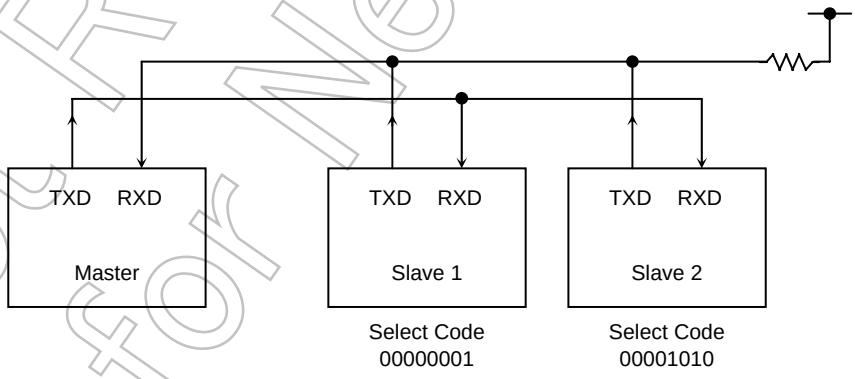
(4) Each slave controller compares the received address to its station address and clears the WU bit if they match.

(5) The master controller transmits a block of data to the selected slave controller (with SC0MOD0.WU bit cleared). Data frames have the most-significant bit (bit 8) cleared to 0.



(6) Slave controllers not addressed continue to monitor the data stream, but discard any frames with the most-significant bit (RB8) cleared, and thus do not generate receive-done interrupts (INTRX0). The addressed slave controller with its WU bit cleared can transmit data to the master controller to notify that it has successfully received the message.

Example: Connecting a master station with two slave stations through a serial link using the f_{SYS}/2 clock as a serial clock



- Master controller settings

Main routine

PGCR	← – – – – – 0 1 –	Configures the PG1 pin as TXD0 and the PG2 pin as RXD0.
PGFC	← – – – – – 1 1 –	
IMC4LL	← – – 1 1 0 1 0 1	Enables INTRX0 and sets its interrupt level to 5.
IMC4LH	← – – 1 1 0 1 0 0	Enables INTTX0 and sets its interrupt level to 4.
SC0MOD0	← 1 0 1 0 1 1 1 0	Selects 9-Bit UART mode and selects $f_{SYS}/2$ as a serial clock.
SC0BUF	← 0 0 0 0 0 0 0 1	Loads the select code for slave 1.

Interrupt routine (INTTX0)

INTCLR	← X X 0 1 0 0 0 1	Clears the interrupt request.
SC0MOD0	← 0 – – – – – – –	Clears the TB8 bit to 0.
SC0BUF	← * * * * * * * *	Loads the transmit data.
End of interrupt processing		

- Slave controller settings

Main routine

PDCR	← – – – – – 0 1 –	Configures the PD0 pin as TXD (open-drain output) and the PD1 pin as RXD.
PDFC	← – – – – – 1 1 –	
PGODE	← – – – – – – 1 –	
IMC4LL	← – – 1 1 0 1 1 0	Enables INTTX0 and INTRX0.
IMC4LH	← – – 1 1 0 1 0 1	
SC0MOD0	← 0 0 1 1 1 1 1 0	Selects 9-Bit UART mode, selects $f_{SYS}/2$ as a serial clock and sets the WU bit to 1.

Interrupt routine (INTRX0)

INTCLR	← X X 0 1 0 0 0 0	Clears the interrupt request.
Reg.	← SC0BUF	
if Reg.	= Select code	
Then		
SC0MOD0	← – – – 0 – – – –	Clears the WU bit to 0.

15. Serial Bus Interface (SBI)

The TMP1962 contains a Serial Bus Interface (SBI) channel, which has the following two operating modes:

- I^2C Bus mode (with multi-master capability)
- Clock-Synchronous 8-Bit SIO mode

In I^2C Bus mode, the SBI is connected to external devices via two pins, PF0 (SDA) and PF1 (SCL). In Clock-Synchronous 8-Bit SIO mode, the SBI is connected to external devices via three pins, PF2 (SCK), PF2 (SO) and PF1 (SI).

The following table shows the programming required to put the SBI in each operating mode.

	PFODE <PFODE1:0>	PFCR <PF2C, PF1C, PF0C>	PFFC <PF2F, PF1F, PF0F>
I^2C Bus Mode	11	X11	011
Clock-Synchronous 8-Bit SIO Mode	XX	101 (clock output) 001 (clock input)	111

X: Don't care

15.1 Block Diagram

Figure 15.1 shows a block diagram of the SBI.

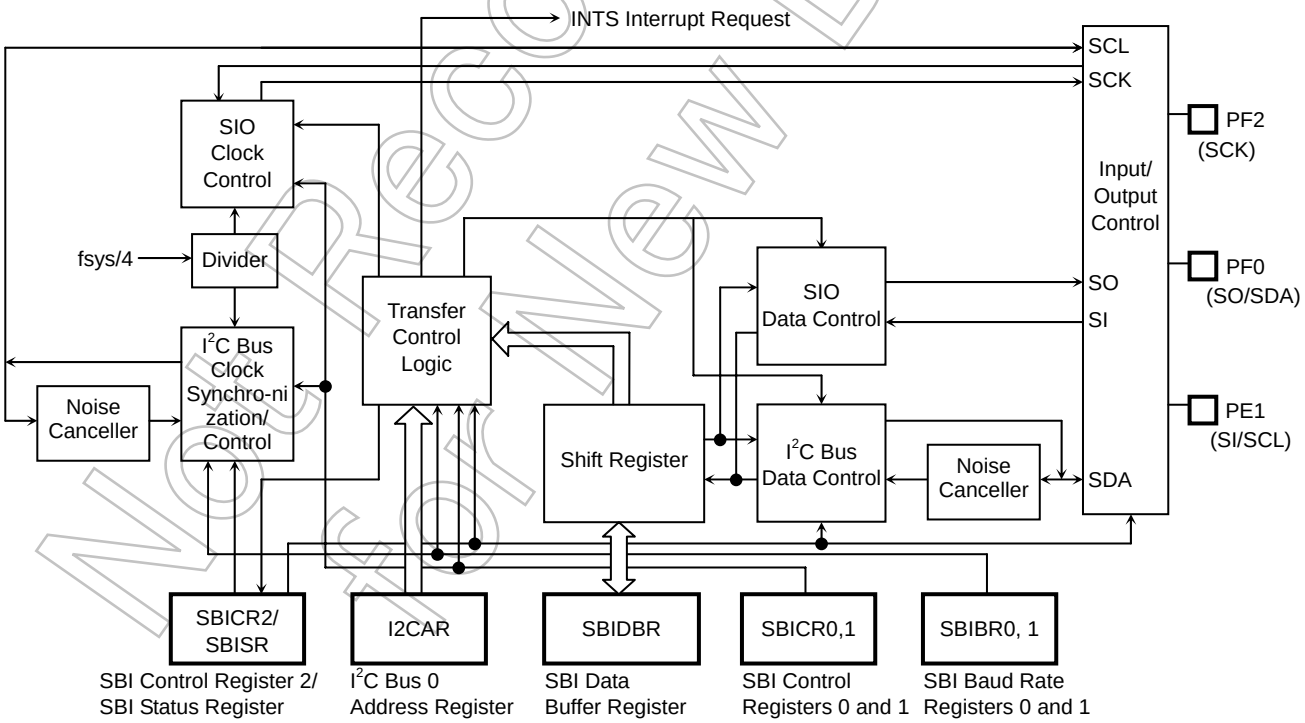


Figure 15.1 SBI Block Diagram

15.2 Registers

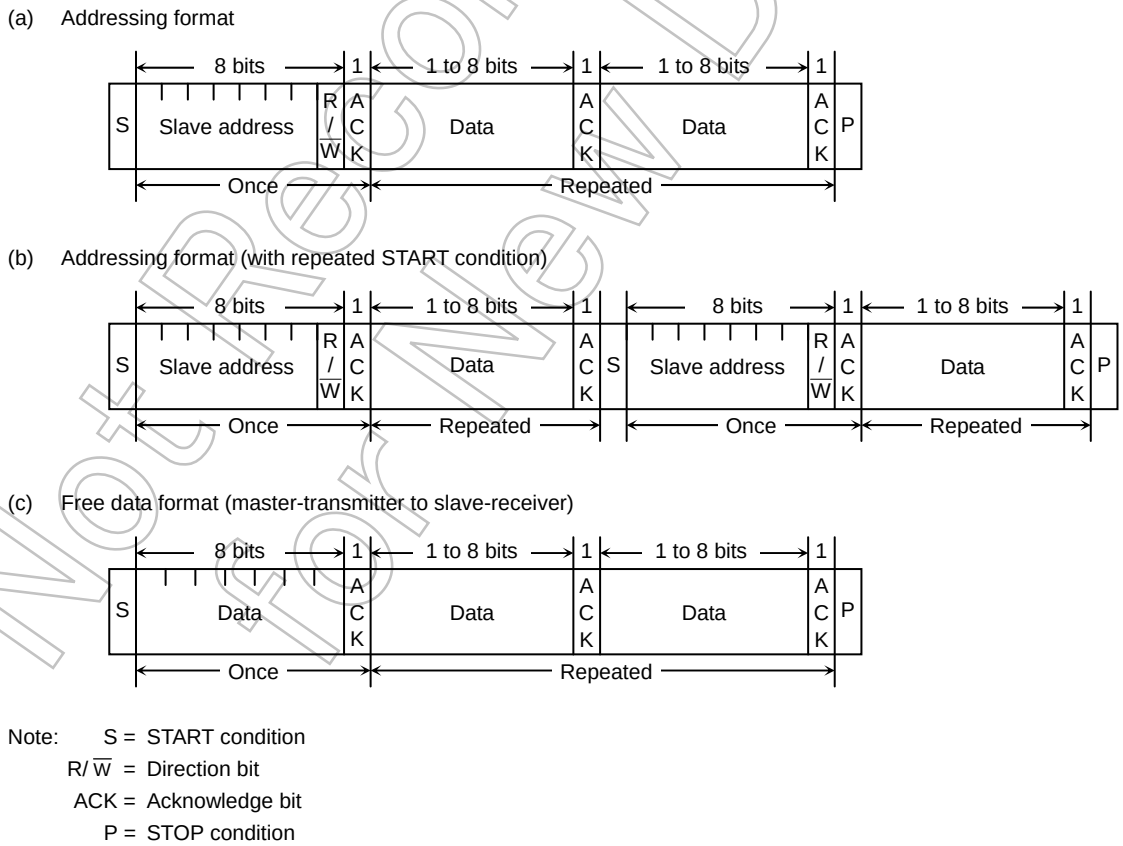
A listing of the registers used to control the SBI follows:

- Serial Bus Interface Control Register 0 (SBICR0)
- Serial Bus Interface Control Register 1 (SBICR1)
- Serial Bus Interface Control Register 2 (SBICR2)
- Serial Bus Interface Data Buffer Register (SBIDBR)
- I^2C Bus Address Register (I2CAR)
- Serial Bus Interface Status Register (SBISR)
- Serial Bus Interface Baud Rate Register 0 (SBIBR0)
- Serial Bus Interface Baud Rate Register 1 (SBIBR1)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to Section 15.4, "Description of the Registers Used in I2C Bus Mode," and Section 15.7, "Description of the Registers Used in Clock-Synchronous 8-Bit SIO Mode."

15.3 I^2C Bus Mode Data Formats

Figure 15.2 shows the serial bus interface data formats used in I^2C Bus mode.



Note: S = START condition
R/$\overline{W}$ = Direction bit
ACK = Acknowledge bit
P = STOP condition

Figure 15.2 I^2C-Bus Mode Data Formats

15.4 Description of the Registers Used in I²C Bus Mode

This section provides a summary of the registers which control I²C bus operation and provide I²C bus status information for bus access/monitoring.

Serial Bus Interface Control Register 0

SBICR0 (0xFFFF_F254)

	7	6	5	4	3	2	1	0
Bit Symbol	SBIEN							
Read/Write	R/W							
Reset Value	0							
Function	SBI operation 0: Disable 1: Enable							

SBIEN: Enables or disables the operation of the SBI. If the SBI is disabled, no clock pulses are supplied to the SBI registers other than the SBICR0, so that power consumption in the system can be reduced (only the SBICR0 can be read or written). To use the SBI, set the SBIEN bit to 1 before configuring other registers of the SBI. Once the SBI operates, all settings in its registers are held if it is disabled.

Note: Bits 0 to 6 of the SBICR0 are read as 0.

Figure 15.3 I²C Bus Mode Registers

Serial Bus Interface Control Register 1

SBICR1 (0xFFFF_F253)

	7	6	5	4	3	2	1	0
Bit Symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	W			R/W		W		R/W
Reset Value	0	0	0	0		0	0	1
Function	Number of bits per transfer (Note 1)			ACK clock pulse 0: No ACK 1: ACK		Internal SCL output clock frequency (Note 2) / Software reset monitor		

On writes: SCK[2:0] = Internal SCL output clock frequency

000	n=4	506 kHz
001	n=5	281 kHz
010	n=6	149 kHz
011	n=7	77 kHz
100	n=8	39 kHz
101	n=9	20 kHz
110	n=10	10 kHz
111		reserved

System clock: fsys (= 40.5 MHz)

Clock gear : fc/1

$$\text{Frequency} = \frac{fsys/4}{2^n + 4} \text{ [Hz]}$$

On reads: SWRMON = Software reset monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

Number of bits per transfer

<BC2:0>	ACK = 0		ACK = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Note 1: Clear the BC[2:0] field to 000 before switching the operating mode to Clock-Synchronous 8-Bit SIO mode.
Note 2: For details on the SCL bus clock frequency, refer to Section 15.5.3, "Serial Clock."

Figure 15.4 I^2C Bus Mode Registers

Serial Bus Interface Control Register 2

SBICR2 (0xFFFF_F250)

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 1)		W (Note 1)	
Reset Value	0	0	0	1	0	0	0	0
Function	Master/ slave 0: Slave 1: Master	Transmit/ receive 0: Receive 1: Transmit	START/ STOP generation 0: STOP condition 1: START condition	INTS interrupt clear 0: - 1: Interrupt clear	Operating mode (Note 2) 00: Port mode 01: SIO mode 10: I²C Bus mode 11: Reserved		Software reset A write of 10 followed by a write of 01	

Operating mode (Note 2)

00	Port mode (serial bus interface output disabled)
01	Clock-Synchronous 8-Bit SIO mode
10	I²C Bus mode
11	Reserved

Note 1: Reading this register causes it to function as a status register (SBISR).

Note 2: Ensure that the bus is free before switching the operating mode to Port mode. Ensure that the port is at logic high before switching from Port mode to I²C Bus or SIO mode.

Figure 15.5 I²C Bus Mode Registers

Table 15.1 Base Clock Resolutions

@fsys = 40.5 MHz

Clock Gear Value GEAR[1:0]	Base Clock Resolution
00 (fc)	fsys/2^2 (0.1 μs)
01 (fc/2)	fsys/2^3 (0.2 μs)
10 (fc/4)	fsys/2^4 (0.4 μs)
11 (fc/8)	fsys/2^5 (0.8 μs)

Serial Bus Interface Status Register

SBISR (0xFFFF_F250)

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
Reset Value	0	0	0	1	0	0	0	0
Function	Master/ slave 0: Slave 1: Master	Transmit/ receive 0: Receive 1: Transmit	I^2C bus status 0: Free 1: Busy	INTS interrupt status 0: The interrupt is asserted. 1: The interrupt is not asserted	Arbitration lost 0: - 1: Detected	Addressed as slave 0: - 1: Detected	General call 0: - 1: Detected	Last received bit 0: 0 1: 1

Last received bit

0	The last bit received was 0.
1	The last bit received was 1.

Addressed as slave

0	—
1	The address on the bus matches the slave address or general-call address

Arbitration lost

0	—
1	Arbitration was lost to another master.

Note: Writing to this register causes it to function as a control register (SBICR2).

Figure 15.6 I^2C Bus Mode Registers

Serial Bus Interface Baud Rate Register 0

SBIBR0 (0xFFFF_F257)

	7	6	5	4	3	2	1	0
Bit Symbol		I2SBI0						
Read/Write		R/W						W
Reset Value		0						0
Function		IDLE 0: Off 1: On						Must be written as 0.

SBI on/off in IDLE2 mode

0	Off
1	On

Serial Bus Interface Baud Rate Register 1

SBIBR1 (0xFFFF_F256)

	7	6	5	4	3	2	1	0
Bit Symbol	P4EN							
Read/Write	R/W							
Reset Value	0							
Function	Internal clock 0: Off 1: On							

Controls the internal baud rate generator

0	Off
1	On

Serial Bus Interface Data Buffer Register

SBIDBR (0xFFFF_F252)

	7	6	5	4	3	2	1	0
Bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive) / W (transmit)							
Reset Value	Undefined							

Note: In Transmitter mode, data must be written to this register, with bit 7 being the most-significant bit (MSB).

I^2C Bus Address Register

I2CAR (0xFFFF_F251)

	7	6	5	4	3	2	1	0
Bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	When the SBI is addressed as a slave, this field specifies a 7-bit I^2C-bus address to which the SBI responds.							Address Recognition mode

Address Recognition mode

0	Recognizes the slave address.
1	Does not recognize the slave address.

Figure 15.7 I^2C Bus Mode Registers

15.5 I^2C Bus Mode Configuration

15.5.1 Acknowledgment Mode

Setting the SBICR1.ACK bit selects Acknowledge mode. When operating as a master, the SBI generates a clock pulse for acknowledge automatically after each data. As a transmitter, the SBI releases the SDA line during this acknowledge cycle so that the receiver of the data transfer can drive the SDA line low to acknowledge receipt of the data. As a receiver, the SBI pulls the SDA line low during the acknowledge cycle after each data has been received.

Clearing the SBICR1.ACK bit selects Non-Acknowledge mode. When operating as a master, the SBI does not generate acknowledge clock pulses.

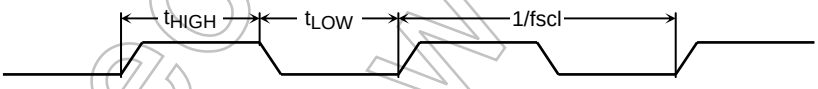
15.5.2 Number of Bits Per Transfer

The SBICR1.BC[2:0] field specifies the number of bits of the next data item to be transmitted or received. After a reset, this field is cleared to 000, causing a 7-bit slave address and the data direction (R/W) bit to be transferred in a packet of eight bits. At other times, the SBICR1.BC[2:0] field keeps a previously programmed value.

15.5.3 Serial Clock

(1) Clock source

The SBICR1.SCK[2:0] field controls the maximum frequency of the serial clock driven out on the SCL pin in Master mode, as illustrated below.



$t_{LOW} = 2^{n-1}/(fsys/4)$

$t_{HIGH} = 2^{n-1}/(fsys/4) + 4/(fsys/4)$

$fscl = 1/(t_{Low} + t_{HIGH})$

$= \dfrac{fsys/4}{2^n + 4}$

SBI0CR1 <SCK2:0>	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10

Figure 15.8 Clock Source

(2) Clock synchronization

Clock synchronization is performed using the wired-AND connection of all I^2C-bus components to the bus. If two or more masters try to transfer messages on the I^2C bus, the first to pull its clock line low wins the arbitration, overriding other masters producing a high on their clock lines.

Clock signals of two or more devices on the I^2C-bus are synchronized to ensure correct data transfers. Figure 15.9 shows a depiction of the clock synchronization mechanism for the I^2C bus with two masters.

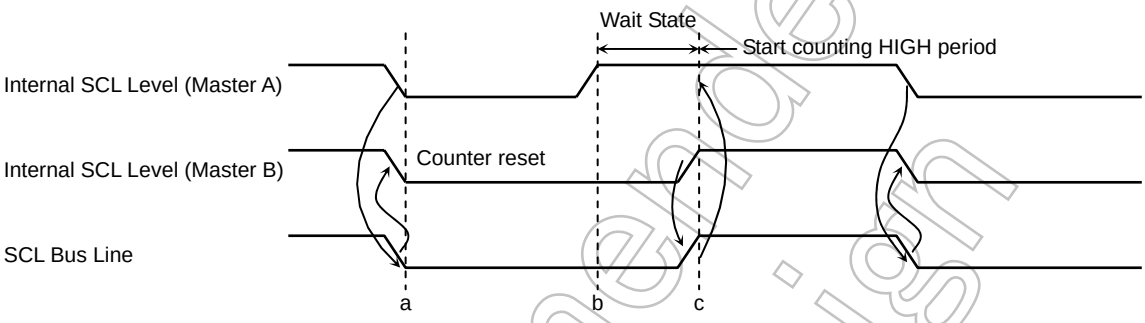


Figure 15.9 Clock Synchronization Example

At point a, Master A pulls its internal SCL level low, bringing the SCL bus line low. The high-to-low transition on the SCL bus line causes Master B to reset its high-level counter and pull its internal SCL level low.

Master A completes its low period at point b. However, the low-to-high transition on its internal SCL level does not change the state of the SCL bus line if Master B's internal SCL level is still within its low period. Therefore, Master A enters a high wait state, where it does not start counting off its high period. When Master B has counted off its low period at point c, its internal SCL level goes high, releasing the SCL bus line (high). There will then be no difference between the internal SCL levels and the state of the SCL bus line, and both Master A and Master B start counting off their high periods.

This way, a synchronized SCL clock is generated with its high period determined by the master with the shortest clock high period and its low period determined by the one with the longest clock low period.

15.5.4 Slave Addressing and Address Recognition Mode

When the SBI is configured to operate as a slave, the SA[6:0] field in the I2CAR must be loaded with the 7-bit I^2C-bus address to which the SBI is to respond. The ALS bit must be cleared for the SBI to recognize the incoming slave address.

15.5.5 Configuring the SBI as a Master or a Slave

Setting the SBICR2.MST bit configures the SBI as a master, and clearing it configures the SBI as a slave. This bit is cleared by hardware when a STOP condition has been detected and when arbitration for the I^2C bus has been lost.

15.5.6 Configuring the SBI as a Transmitter or a Receiver

The SBICR2.TRX bit is set or cleared by hardware to configure the SBI as a transmitter or a receiver.

As a slave, the SBI is put in either Slave-Receiver or Slave-Transmitter mode, depending on the value of the data direction (R/W) bit transmitted by the master. When the SBI is addressed as a slave, the TRX bit reflects the value of the R/W bit. The TRX bit is set or cleared on the following occasions:

- when transferring data using addressing format
- when the received slave address matches the value in the I2CCR
- when a general-call address is received; i.e., the eight bits following the START condition are all zeros.

As a master, the SBI is put in either Master-Transmitter or a Master-Receiver mode upon reception of an acknowledge from an addressed slave. The TRX bit changes to the opposite value of the R/W bit sent by the SBI. If the SBI does not receive an acknowledge from a slave, the TRX bit retains the previous value.

The TRX bit is cleared by hardware when a STOP condition has been detected and when arbitration for the I^2C bus has been lost.

15.5.7 Generating START and STOP Conditions

When the SBISR.BB bit is cleared, the bus is free. At this time, writing 1s to the MST, TRX, BB and PIN bits in the SBICR2 causes the SBI to generate a START condition on the bus and shift out 8-bit I^2C-bus data. Before generating a START condition, the ACK bit must be set to 1.

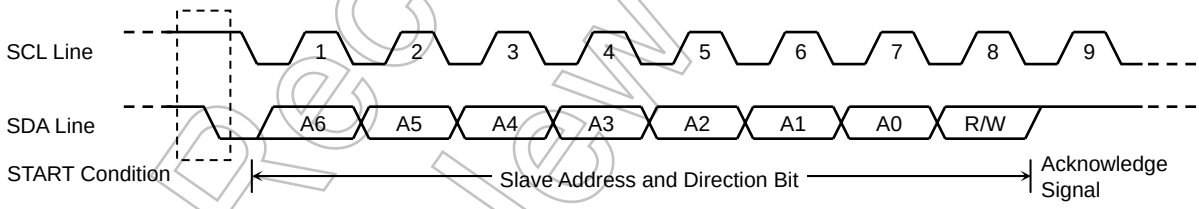


Figure 15.10 Generating a START Condition and a Slave Address

When the SBISR.BB bit is set, the bus is busy. When SBISR.BB = 1, writing 1s to the MST, TRX and PIN bits and a 0 to the BB bit causes the SBI to start a sequence for generating a STOP condition on the bus to abort the transfer. The MST, TRX, BB and PIN bits should not be altered until a STOP condition appears on the bus.

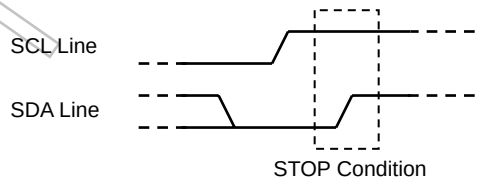


Figure 15.11 Generating a STOP Condition

The BB bit can be read to determine if the I^2C bus is in use. The BB bit is set when a START condition is detected and cleared when a STOP condition is detected.

15.5.8 Asserting and Deasserting Interrupt Requests

When an SBI interrupt (INTS) is generated, the PIN bit in the SBICR2 is cleared to 0. While the PIN bit is 0, the SBI pulls the SCL line low.

After transmission or reception of one data word on the I^2C bus, the PIN bit is automatically cleared. In Transmitter mode, the PIN bit is subsequently set to 1 each time the SBIDBR is written. In Receiver mode, the PIN bit is set to 1 each time the SBIDBR is read. It takes a period of t_{LOW} for the SCL line to be released after the PIN bit is set.

In Address Recognition mode (ALS = 0), the PIN bit is cleared when the SBI is addressed as a slave and the received slave address matches the value in the I2CCR or is all 0s (i.e., a general call). A write of 1 by software sets the PIN bit, but a write of 0 has no effect on this bit.

15.5.9 SBI Operating Modes

The SBIM[1:0] field in the SBICR2 is used to select an operating mode of the SBI. To configure the SBI for I^2C Bus mode, set the SBIM[1:0] field to 10. A switch to Port mode should only be attempted when the bus is free.

15.5.10 Lost-Arbitration Detection Monitor

The I^2C bus is a multi-master bus and has an arbitration procedure to ensure correct data transfers. A master may start a transfer only if the bus is free. A master that attempts to generate a START condition while the bus is busy loses bus arbitration, with no START condition occurring on the SDA and SCL lines. The I^2C-bus arbitration takes place on the SDA line.

Figure 15.12 shows the arbitration procedure for two masters. Up until point a, the internal data levels of Master A and Master B are the same. At point a Master B's internal data level makes a low-to-high transition while Master A's internal data level remains at logic low. However, the SDA bus line is held low because it is the wired-AND of the two data outputs. When the SCL bus clock goes high at point b, the addressed slave device reads the data transmitted by Master A (i.e., winning master). Master B loses arbitration and switches off its data output stage, releasing its SDA line (high), so that it does not affect the data transfer initiated by the winning master. In case two competing masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

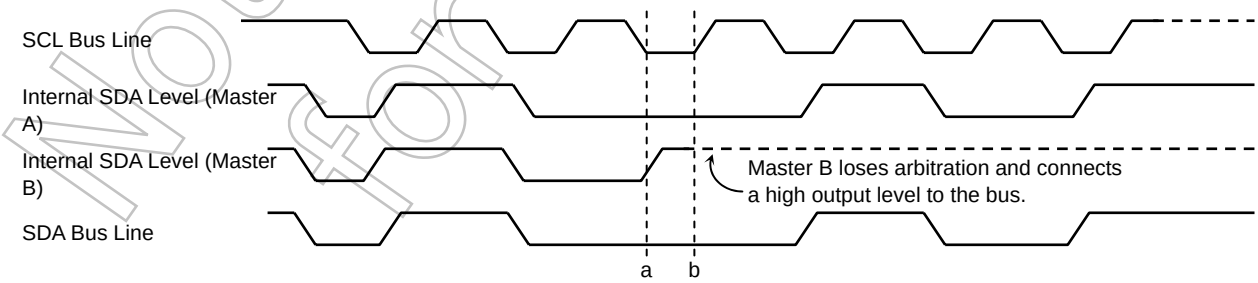


Figure 15.12 Arbitration Procedure of Two Masters

A master compares its internal data level to the actual level on the SDA line at the rising edge of the SCL clock. The master loses arbitration if there is a difference between these two values. The losing master sets the AL bit in the SBISR to 1, which causes the MST and TRX bits in the same register to be cleared. That is, the losing master switches to Slave-Receiver mode. The AL bit is subsequently cleared when data is written to or read from the SBIDBR and when the SBICR2 is programmed with new parameters.

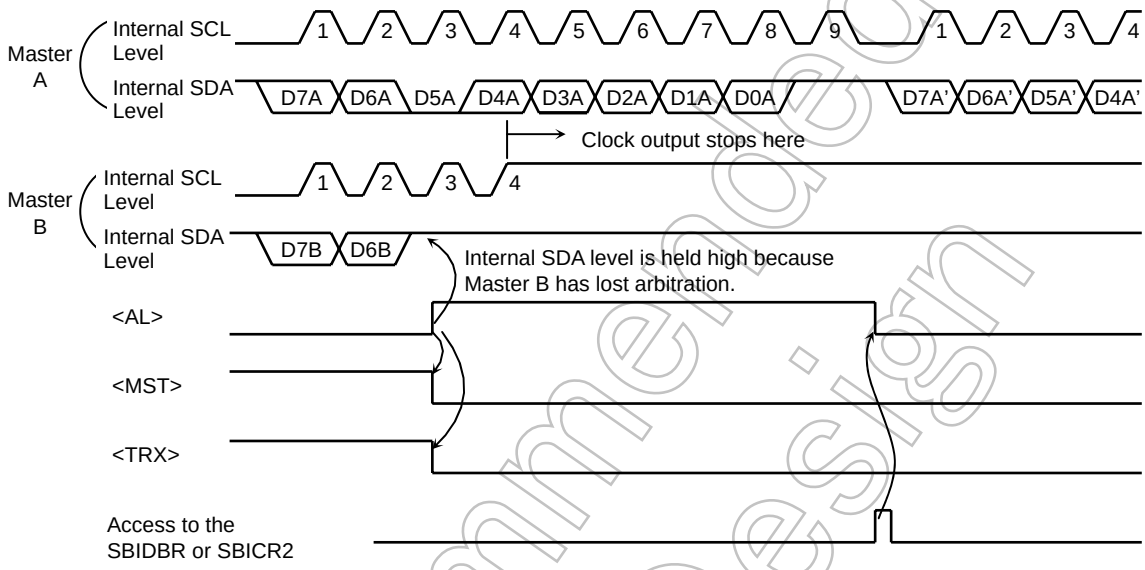


Figure 15.13 Master B Loses Arbitration (D7A = D7B, D6A = D6B)

15.5.11 Slave Address Match Monitor

When acting as a slave-receiver, the ALS bit in the I2CCR determines whether the SBI recognizes the incoming slave address or not. In Address Recognition mode (i.e., ALS = 0), the AAS bit in the SBISR is set when an incoming address over the I^2C bus matches the value in the I2CCR or when the general-call address has been received. When ALS = 1, the AAS bit is set when the first data word has been received. The AAS bit is cleared each time the SBIDBR is read or written.

15.5.12 General-Call Detection Monitor

When acting as a slave-receiver, the AD0 bit in the SBISR is set when a general-call address has been received. The general-call address is detected when the eight bits following a START condition are all zeros. The AD0 bit is cleared when a START or STOP condition is detected on the bus.

15.5.13 Last Received Bit Monitor

The LRB bit in the SBISR holds the value of the last bit received over the SDA line at the rising edge of the SCL clock. In Acknowledge mode, reading this bit immediately after generation of the INTS2 interrupt returns the value of the ACK signal.

15.5.14 Software Reset

The SBI provides a software reset, which permits recovery from system lockups caused by external noise. A software reset is performed by a write of 10 followed by a write of 01 to the SWRST[1:0] field in the SBICR2. After a software reset, all control and status register bits are initialized to their reset values. Upon resetting the SBI, the SWRST[1:0] field is automatically cleared to 00.

Note: A software reset causes the SBI operating mode to switch from I^2C Bus mode to Clock-Synchronous mode.

15.5.15 Serial Bus Interface Data Buffer Register (SBIDBR)

The SBIDBR is a data buffer interfacing to the I^2C bus. All read and write operations to/from the I^2C bus are done via this register.

When the SBI is acting as a master, loading this register with a slave address and a data direction bit causes a START condition to be generated.

15.5.16 I^2C Bus Address Register (I2CAR)

When the SBI is configured as a slave, the SA[6:0] field in the I2CAR must be loaded with the 7-bit I^2C-bus address to which the SBI is to respond. If the ALS bit in the I2CAR is cleared, the SBI recognizes a slave address transmitted by the master device, interpreting incoming frame structures as per addressing format. If the ALS bit is set, the SBI does not recognize a slave address and interprets all frame structures as per free data format.

15.5.17 Baud Rate Register (SBIBR1)

Before the I^2C bus can be used, the P4EN bit in the SBIBR1 must be set to enable the SBI internal baud rate generation logic.

15.5.18 IDLE Setting Register (SBIBR0)

The I2SBI bit in the SBIBR0 determines whether the SBI is shut down or not when the TMP1962 is put in IDLE standby mode. This register must be programmed before executing an instruction for entering a standby mode..

15.6 Programming Sequences in I^2C Bus Mode

15.6.1 SBI Initialization

First, program the P4EN bit in the SBIBR1, and the ACK and SCK[2:0] bits in the SBICR1. Set the SBIBR1.P4EN bit to 1 to enable the internal baud rate generation logic. Write 0s to bits 7-5 and bit 3 in the SBICR1.

Next, program the I2CAR. The SA[6:0] field in the I2CAR defines the chip's slave address, and the ALS bit (bit 0) selects an address recognition mode. (The ALS bit must be cleared when using the addressing format.)

Next, program the SBICR2 to initially configure the SBI in Slave-Receiver mode; i.e., clear the MST, TRX and BB bits to 0, set the PIN bit to 1 and set the SBIM[1:0] field to 10. Write 00 to the SWRST[1:0] field.

		7	6	5	4	3	2	1	0	
SBIBR1	←	1	0	0	0	0	0	0	0	Enables internal baud rate generator.
SBICR1	←	0	0	0	X	0	X	X	X	Disables generation of ACK and selects SCL clock frequency.
I2CAR	←	X	X	X	X	X	X	X	X	Loads a slave address and selects address recognition mode.
SBICR2	←	0	0	0	1	1	0	0	0	Configures the SBI in Slave-Receiver mode.

Note: X = Don't care

15.6.2 Generating a START Condition and a Slave Address

(1) Master mode

In Master mode, the following steps are required to generate a START condition and a slave address on the I^2C-bus.

First, ensure that the bus is free (i.e., SBICR2.BB = 0). Next, set the ACK bit in the SBICR1 to enable generation of acknowledge clock pulses. Then, load the SBIDBR with a slave address and a data direction bit to be transmitted via the I^2C bus.

When BB = 0, writing 1s to the MST, TRX, BB and PIN bits in the SBICR2 causes a START condition to be generated on the bus. Following a START condition, the SBI generates SCL clock pulses nine times: the SBI shifts out the contents of the SBIDBR with the first eight SCL clocks, and releases the SDA line during the last (i.e., ninth) SCL clock to receive an acknowledgement signal from the addressed slave.

The INTS interrupt request is generated on the falling edge of the ninth SCL clock pulse, and the PIN bit in the SBICR2 is cleared to 0. In Master mode, the SBI holds the SCL line low while the PIN bit is 0. Upon interrupt, the TRX bit either remains set or is cleared according to the value of the transmitted direction bit, provided an acknowledgement signal has been returned from the slave.

Settings in main routine

```
                    7 6 5 4 3 2 1 0
→ Reg.      ← SBISR
  Reg.      ← Reg. e 0x20
  if Reg.   ≠ 0x00                    Ensures that the bus is free.
└ Then
  SBICR1    ← X X X 1 0 X X X       Selects Acknowledgement mode.
  SBIDR1    ← X X X X X X X X       Loads the slave address and a data direction bit.
  SBICR2    ← 1 1 1 1 1 0 0 0       Generates a START condition.
```

INTS interrupt routine

```
INTCLR ← 0X14                       Clears the interrupt request.
Interrupt processing
End of interrupt
```

(2) Slave mode

In Slave mode, the following steps are required to receive a START condition and a slave address via the I^2C bus.

Upon detection of a START condition, the SBI clocks in a 7-bit slave address and a data direction bit transmitted by the master during the first eight SCL clock pulses. If the received slave address matches its own address in the I2CAR or is equal to the general-call address (00H), the SBI pulls the SDA line low during the last (i.e., ninth) SCL clock for acknowledgement.

The INTS interrupt request is generated on the falling edge of the ninth SCL clock pulse, and the PIN bit in the SBICR2 is cleared to 0. In Slave mode, the SBI holds the SCL line low while the PIN bit is 0.

Note: The user can only use a DMA transfer:

- **when there is only one master and only one slave on the I^2C bus; and**
- **continuous transmission or reception is possible.**

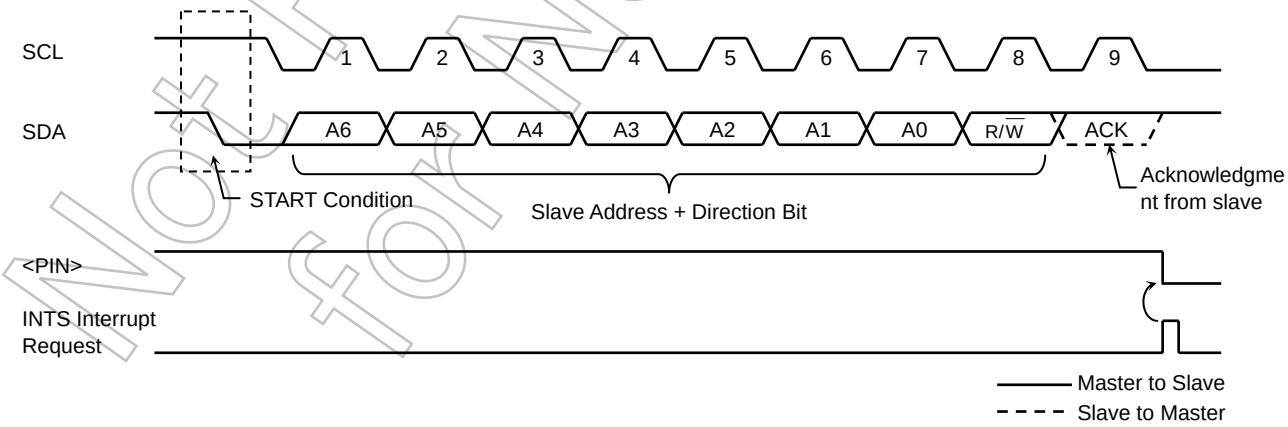


Figure 15.14 Generation of a START Condition and a Slave Address

15.6.3 Transferring a Data Word

Each time a data word has been transmitted or received, the INTS interrupt is generated. It is the responsibility of the INTS interrupt service routine to test the MST bit in the SBICR2 to determine whether the SBI is in Master or Slave mode.

(1) Master mode (SBICR2.MST = 1)

If the MST bit in the SBICR2 is set, then test the TRX bit in the same register to determine whether the SBI is in Master-Transmitter or Master-Receiver mode.

Master-Transmitter mode (SBICR2.TRX = 1)

Test the LRB bit in the SBISR. If the LRB bit is set, that means the slave-receiver requires no further data to be sent from the master-transmitter. The master-transmitter must then generate a STOP condition as described later to stop transmission.

If the LRB bit is cleared, that means the slave-receiver requires further data. If the number of bits per transfer is 8, then write the transmit data into the SBIDBR. When using other data length, program the BC[2:0] and ACK bits in the SBICR1, and then write the transmit data into the SBIDBR. When the SBIDBR is loaded, the PIN bit in the SBISR is set to 1, and the transmit data is shifted out from the SDA0 pin, clocked by the SCL clock. Once the transfer is complete, the INTS interrupt is generated, the PIN bit is cleared, and the SCL line is pulled low. To transmit further data, test the LRB bit again and repeat the above procedure.

INTS interrupt

if MST = 0
Then go to slave-mode processing
if TRX = 0
Then go to receiver-mode processing
if LRB = 0
Then go to processing for generating a STOP condition
SBICR1 ← X X X X 0 X X X Sets number of bits to be transmitted and specify whether ACK is required.
SBIDBR ← X X X X X X X X Loads the transmit data.
End of interrupt processing

Note: X = Don't care

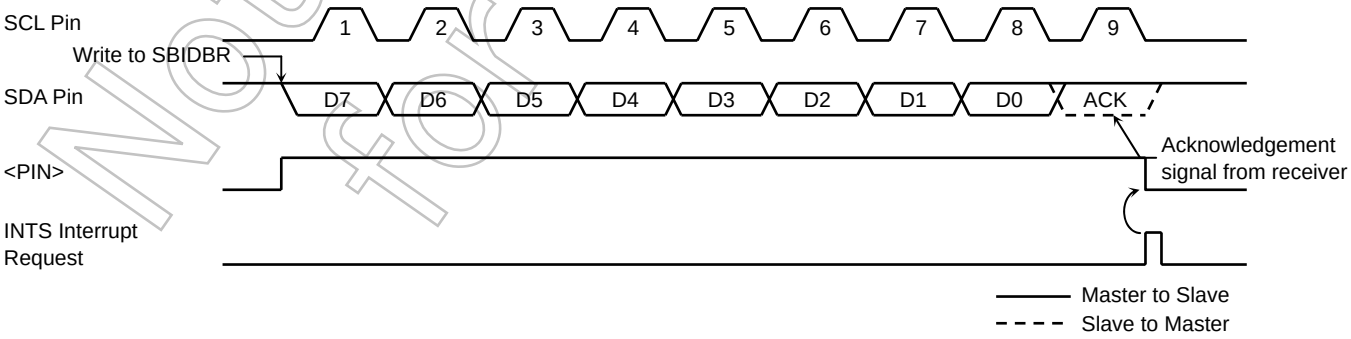


Figure 15.15 SBICR1.BC[2:0] = 000 and SBICR1.ACK = 1 (Master-Transmitter Mode)

Master-Receiver mode (SBICR2.TRX = 0)

If the number of bits per transfer is 8, read the SBIDBR. When using other data length, program the BC[2:0] and ACK bits in the SBICR1, and then read the SBIDBR. The first read of the SBIDBR is a dummy read because data has not yet been received. A dummy read returns an undefined value. Upon this read, the SCL line is released, the PIN bit in the SBISR is set, and the SCL clock is driven out to receive a data word into the SBIDBR. The master-transmitter generates an acknowledgement signal (i.e., a low level) on the SDA line following the last received bit.

Once the transfer is complete, the INTS interrupt is generated, the PIN bit is cleared, and the SCL line is pulled low. Each subsequent read from the SBIDBR is accompanied by an SCL clock pulse for a data word and an acknowledgement signal.

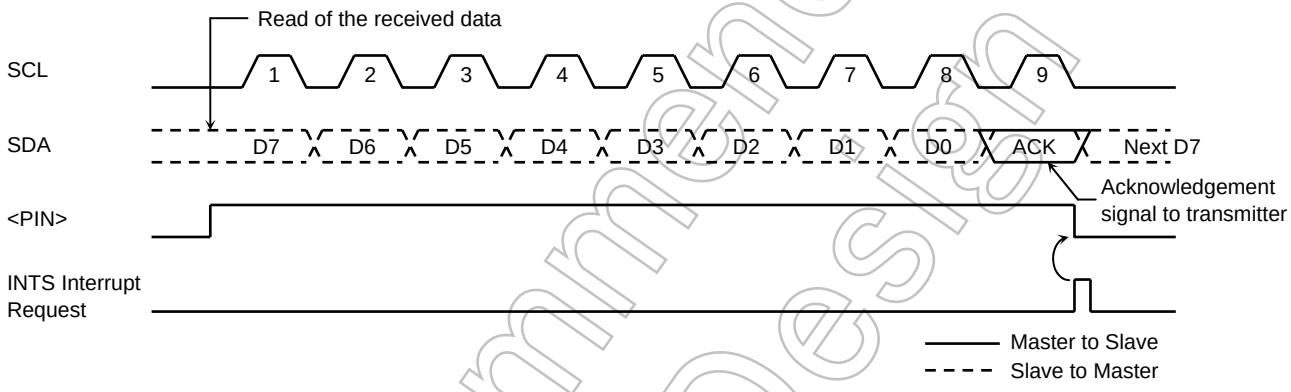


Figure 15.16 SBICR1.BC[2:0] = 000 and SBICR1.ACK = 1 (Master-Receiver Mode)

To prepare to terminate the data transfer, the master-receiver must clear the ACK bit in the SBICR1 immediately before the read of the second to last data word. This causes an acknowledge clock pulse to be suppressed on the last data word. When the transfer is complete, the INTS interrupt is generated. After interrupt processing, the INTS interrupt handler must set the BC[2:0] field in the SBICR1 to 001 and read the SBIDBR, so that a clock is generated on the SCL line once. With the ACK bit cleared, the master-receiver holds the SDA line high, which signals the end of transfer to the slave-transmitter.

Then, the SBI generates the INTS interrupt again, whereupon the INTS interrupt service routine must generate a STOP condition to stop communication via the I^2C bus.

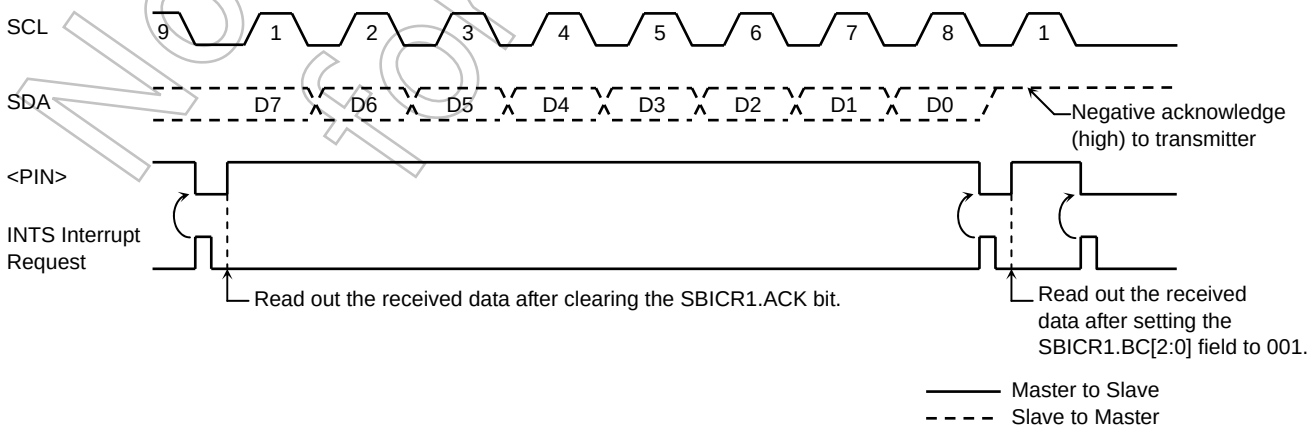


Figure 15.17 Terminating Data Transmission in Master-Receiver Mode

Example: When receiving N data words

INTS interrupt (after data transmission)

		7 6 5 4 3 2 1 0	
SBICR1	←	X X X X 0 X X X	Sets the number of bits to be received and specifies whether ACK is required.
Reg.	←	SBI0CBR	Dummy read
End of interrupt			

INTS interrupt (first to (N-2)th data reception)

		7 6 5 4 3 2 1 0	
Reg.	←	SBIDBR	Reads the first to (N-2)th data words.
End of interrupt			

INTS interrupt ((N-1)th data reception)

		7 6 5 4 3 2 1 0	
SBI0CR1	←	X X X 0 0 X X X	Disables generation of acknowledgement clock.
Reg.	←	SBIDBR	Reads the (N-1)th data word.
End of interrupt			

INTS interrupt (Nth data reception)

		7 6 5 4 3 2 1 0	
SBI0CR1	←	0 0 1 0 0 X X X	Generates a clock once.
Reg.	←	SBIDBR	Reads the Nth data word.
End of interrupt			

INTS interrupt (after completing data reception)

Processing for generating STOP condition	Terminates data transmission.
End of interrupt	

X = Don't care

(2) Slave mode (SBICR2.MST = 0)

If the MST bit in the SBICR2 is cleared, the SBI is in Slave mode. In Slave mode, the SBI generates the INTS interrupt on four occasions: 1) when the SBI has received any slave address; 2) when the SBI has received a general-call address; 3) when a data transfer has been completed in response to a received slave address that matches its own address in the I2CAR; and 4) when a data transfer has been completed in response to a general-call. Also, if the SBI, as a master, loses arbitration for the I^2C bus, it switches to Slave mode. If arbitration is lost during a data transfer, SCL continues to be generated until the data word is complete; then the INTS interrupt is generated. When the INTS interrupt occurs, the PIN bit in the SBISR is cleared, and the SCL line is pulled low. When the SBIDBR is read or written or when the PIN bit is set back to 1, the SCL line is released after a period of t_{LOW}.

Processing to be done in Slave mode varies, depending on whether or not the SBI has switched over to Slave mode as a result of lost arbitration.

Test the AL, TRX, AAS and AD0 bits in the SBISR to determine the processing required, as summarized in Table 15.2

Example: When the received slave address matches the SBI's own address and the data direction bit is 1

INTS interrupt

```
if TRX = 0
Then go to other processing
if AL = 1
Then go to other processing
if AAS = 0
Then go to other processing
SBICR1  ← X X X 1 0 X X X      Specifies the number of bits to be transmitted.
SBIDBR  ← X X X X 0 X X X      Loads the transmit data.
```

Note: X = Don't care

Table 15.2 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration was lost while the slave address was being transmitted, and the SBI received a slave address with the direction bit set transmitted by another master.	Set the SBICR1.BC[2:0] field to the number of bits in a data word and write the transmit data into the SBIDBR.
	0	1	0	In Slave-Receiver mode, the SBI received a slave address with the direction bit set transmitted by the master.	
		0	0	In Slave-Transmitter mode, the SBI has completed a transmission of one data word.	Test the SBISR.LRB bit. If the LRB bit is set, that means the master-receiver does not require further data. Set the SBICR2.PIN bit to 1 and clear the TRX bit to 0 to release the bus. If the LRB bit is cleared, that means the master-receiver requires further data. Set the SBICR1.BC[2:0] field to the number of bits in the data word and write the transmit data to the SBIDBR.
0	1	1	1/0	Arbitration was lost while a slave address was being transmitted, and the SBI received either a slave address with the direction bit cleared or a general-call address transmitted by another master.	Read the SBIDBR (a dummy read) to set the SBICR2.PIN bit to 1, or write a 1 to this bit.
		0	0	Arbitration was lost while a slave address or a data word was being transmitted, and the transfer terminated.	
	0	1	1/0	In Slave-Receiver mode, the SBI received either a slave address with the direction bit cleared or a general-call address transmitted by the master.	
		0	1/0	In Slave-Receiver mode, the SBI has completed a reception of a data word.	Set the SBICR1.BC[2:0] field to the number of bits in the data word and read the received data from the SBIDBR.

15.6.4 Generating a STOP Condition

When the SBISR.BB bit is set, setting the MST, TRX and PIN bits in the SBICR2 to 1 and clearing the BB bit in the same register causes the SBI to start a sequence for generating a STOP condition on the I^2C bus. Do not alter the contents of these bits until the STOP condition is present on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released (high) again; when SCL is high, the SBI drives the SDA pin high to generate a STOP condition.

```
                7 6 5 4 3 2 1 0
SBICR2   ← 1 1 0 1 1 0 0 0        Generates a STOP condition.
```

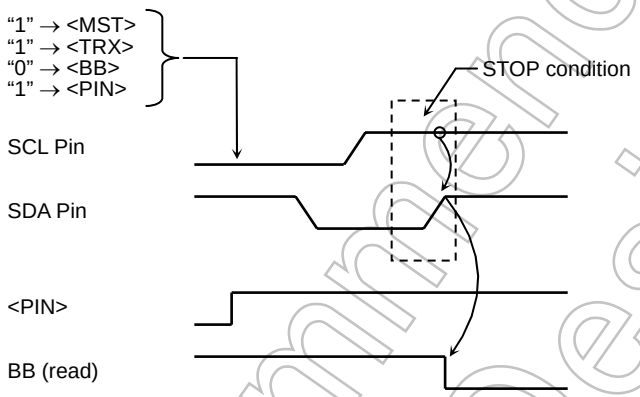


Figure 15.18 Generating a STOP Condition

15.6.5 Repeated START Condition

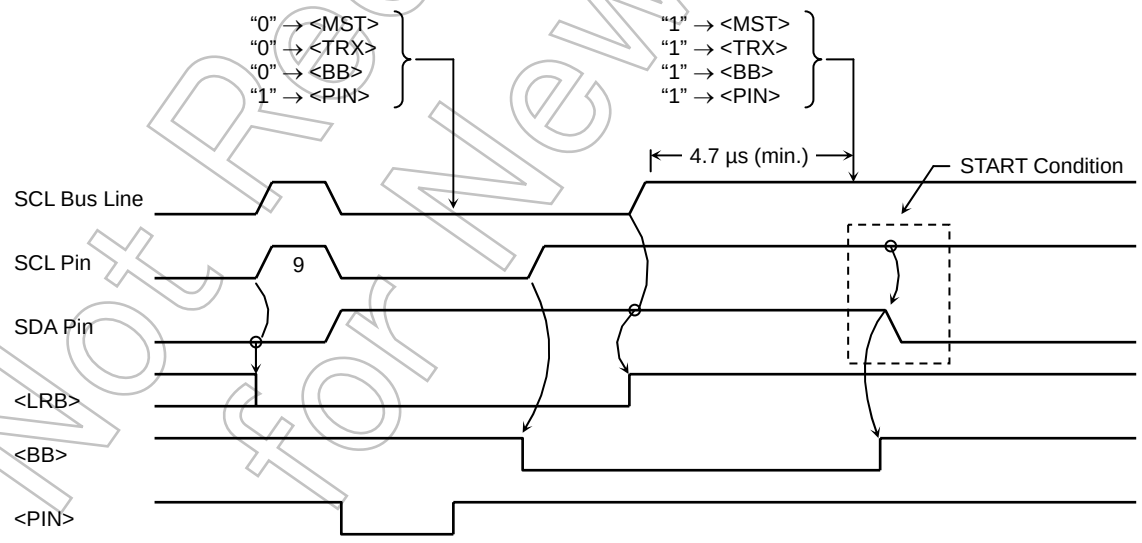
A data transfer is always terminated by a STOP condition. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition and address another slave or change the data direction without first generating a STOP condition. The following describes the steps required to generate a repeated START condition.

First, clear the MST, TRX and BB bits in the SBICR2 and set the PIN bit in the same register to release the bus. This causes the SDA pin to be held high and the SCL pin to be released. Because no STOP condition is generated on the bus, other devices think that the bus is busy. Then, poll the SBISR.BB bit until it is cleared to ensure that the SCL pin is released. Next, poll the LRB bit until it is set to ensure that no other device is pulling the SCL bus line low. Once the bus is determined to be free this way, use the steps described in Section 15.6.2 to generate a START condition.

To satisfy the minimum setup time of the START condition, at least 4.7-µs wait period (in normal mode) must be created by software after the bus becomes free.

```
                    7 6 5 4 3 2 1 0
SBICR2   ← 0 0 0 1 1 0 0 0   Releases the bus.
if SBISR<BB> ≠ 0             Checks that the SCL pin is released.
Then
if SBISR<LRB> ≠ 1            Checks that no other device is pulling the SCL line low.
Then
4.7 µs Wait
SBICR1   ← X X X 1 0 X X X   Selects Acknowledge mode.
SBIDBR   ← X X X X X X X X   Loads a slave address and the direction bit.
SBICR2   ← 1 1 1 1 1 0 0 0   Generates a START condition.

X = Don't care
```



Note: Ensure that MST = 1 before writing a 0 to MST. When MST = 0, writing a 0 to MST does not enable a repeated start.

Figure 15.19 Repeated START Condition

15.7 Description of Registers Used in Clock-Synchronous 8-Bit SIO Mode

This section provides a summary of the registers which control clock-synchronous 8-bit SIO operation and provides its status information for monitoring.

Serial Bus Interface Control Register 0

SBICR0 (0xFFFF_F254)

	7	6	5	4	3	2	1	0
Bit Symbol	SBIEN							
Read/Write	R/W							
Reset Value	0							
Function	SBI operation 0: Disable 1: Enable							

SBIEN: Enables or disables the operation of the SBI. If the SBI is disabled, no clock pulses are supplied to the SBI registers other than the SBICR0, so that power consumption in the system can be reduced (only the SBICR0 can be read or written). To use the SBI, set the SBIEN bit to 1 before configuring other registers of the SBI. Once the SBI operates, all settings in its registers are held if it is disabled.

Note: Bits 0 to 6 of the SBICR0 are read as 0.

Serial Bus Interface Control Register 1

SBICR1 (0xFFFF_F253)

	7	6	5	4	3	2	1	0
Bit Symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	W					W		R/W
Reset Value	0	0	0	0		0	0	1
Function	Start transfer 0: Stop 1: Start	Abort transfer 0: Continue 1: Abort	Transfer mode 00: Transmit mode 01: Reserved 10: Transmit/Receive mode 11: Receive mode			Serial clock frequency		

On writes: SCK[2:0] = Serial clock frequency

000	n = 3	1.27 MHz
001	n = 4	633 kHz
010	n = 5	316 kHz
011	n = 6	158 kHz
100	n = 7	79 kHz
101	n = 8	40 kHz
110	n = 9	20 kHz
111	—	External clock

System clock: fsys (= 40.5 MHz)
Clock gear: fc/1
Frequency = $\frac{fsys/4}{2^n}$ [Hz]

Note: Clear the SIOS bit and set the SIOINH bit before programming the transfer mode and serial clock frequency bits.

Serial Bus Interface Data Buffer Register

SBIDBR (0xFFFF_F252)

	7	6	5	4	3	2	1	0
Bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive)/W (transmit)							
Reset Value	Undefined							

Figure 15.20 SIO Mode Registers

Serial Bus Interface Control Register 2

SBICR2 (0xFFFF_F250)

	7	6	5	4	3	2	1	0
Bit Symbol					SBIM1	SBIM0		
Read/Write					W			
Reset Value					0	0		
Function					SBI Operating mode 00: Port mode 01: SIO mode 10: I^2C Bus mode 11: Reserved			

Serial Bus Interface Register

SBISR (0xFFFF_F250)

	7	6	5	4	3	2	1	0
Bit Symbol					SIOF	SEF		
Read/Write					R			
Reset Value					0	0		
Function					Serial transfer status monitor 0: Termi-nated 1: In progress	Shift operation status monitor 0: Terminated 1: In progress		

Serial Bus Interface Baud Rate Register 0

SBIBR0 (0xFFFF_F257)

	7	6	5	4	3	2	1	0
Bit Symbol		I2SBI						
Read/Write		R/W						W
Reset Value		0						
Function		IDLE 0: Off 1: On						Must be written as 0.

Serial Bus Interface Baud Rate Register 1

SBIBR1 (0xFFFF_F256)

	7	6	5	4	3	2	1	0
Bit Symbol	P4EN							
Read/Write	R/W							
Reset Value	0							
Function	Internal clock 0: Off 1: On							

Figure 15.21 SIO Mode Registers

15.7.1 Serial Clock

(1) Clock source

The clock source for SIO mode can be selected from internal and external clocks through the programming of the SCK[2:0] field in the SBICR1.

Internal clocks

One of the seven internal clocks can be used as a serial clock, which is driven onto the SCK pin. At the beginning of a transfer, the SCK clock will start out at logic high.

If software is slow and the reading of the received data or the writing of the transmit data cannot keep up with the serial clock rate, the SBI automatically inserts a wait period, as shown below. During this period, the serial clock is temporarily stopped to suspend a shift operation.

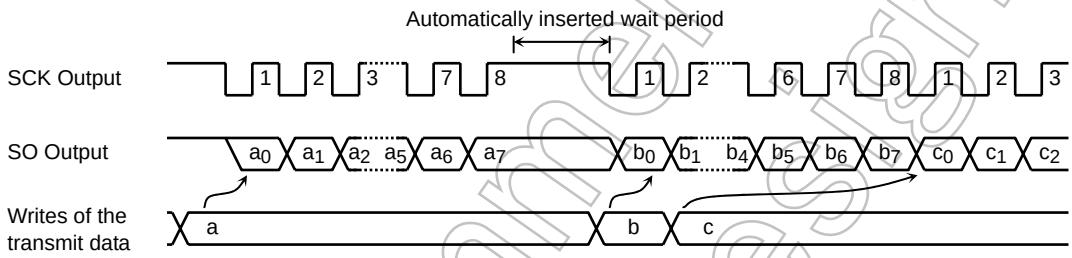


Figure 15.22 Automatic Wait Insertion

External clock (SBICR1.SCK[2:0] = 111)

If the SCK[2:0] field in the SBICR1 contains 111, the SBI uses an external clock supplied from the SCK pin as a serial clock. For proper shift operations, the clock high width and the clock low width must satisfy the following relationship.

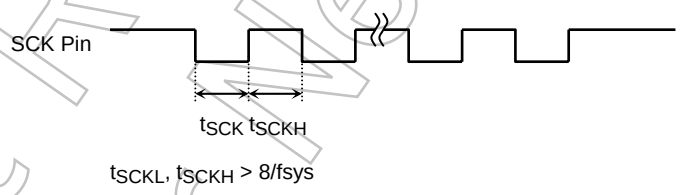


Figure 15.23 Maximum External Clock Frequency

(2) Shift edge types

In Transmit mode, leading-edge shift is used. In Receive mode, trailing-edge shift is used.

Leading-edge shift

Every bit of SIO data is shifted by the leading edge of the serial clock (falling edge of SCK).

Trailing-edge shift

Every bit of SIO data is shifted by the trailing edge of the serial clock (rising edge of SCK).

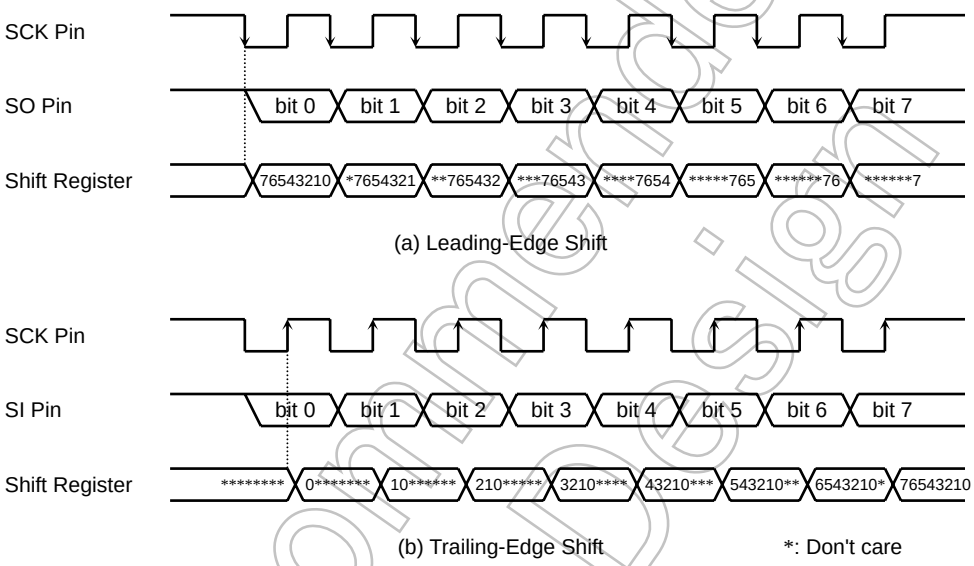


Figure 15.24 Shift Edge Types

15.7.2 Transfer Modes

The SBI supports three SIO transfer modes: Receive mode, Transmit mode and Transmit/Receive mode. The SIOM[1:0] field in the SBICR1 is used to select a transfer mode.

(1) 8-Bit Transmit mode

Configure the SIO interface in Transmit mode and write the transmit data into the SBIDBR.

Then setting the SIOS bit in the SBICR1 initiates a transmission. The contents of the SBIDBR are moved to an internal shift register and then shifted out on the SO pin, with the least-significant bit (LSB) first, synchronous to the serial clock. Once the transmit data is transferred to the shift register, the SBIDBR becomes empty, and the buffer-empty interrupt (INTSBI) is generated.

In Internal Clock mode, the SIO interface will be in wait state (SCK will stop) until the INTS interrupt service routine provides the next transmit data to the SBIDBR. Once the SBIDBR is loaded, the SIO interface will automatically get out of the wait state.

In External Clock mode, the INTS interrupt service routine must provide the next transmit data to the SBIDBR before the previous transmit data has been shifted out. Therefore, the data rate is a function of the maximum latency between when the INTS interrupt is generated and when the SBIDBR is loaded by the interrupt service routine.

At the beginning of a transmission, the value of the last bit of the previously transmitted byte appears on the SO pin between when the SBISR.SIOF bit is set and when SCK subsequently goes low.

Transmission can be terminated by the INTS interrupt service routine clearing the SIOS bit to 0 or setting the SIOINH bit to 1. If the SIOS bit is cleared, the remaining bits in the SBIDBR continue to be shifted out before transmission ends. In this case, software can check the SBISR.SIOF bit to determine whether transmission has come to an end (0 = end-of-transmission). If the SIOINH bit is set, the ongoing transmission is aborted immediately, and the SIOF bit is cleared at that point.

In External Clock mode, the SIOS bit must be cleared before the SIO interface begins shifting out the next transmit data. Otherwise, the SIO will stop after sending out dummy data.

```
                  7 6 5 4 3 2 1 0
SBICR1    ← 0 1 0 0 0 X X X      Selects transmit mode.

SBIDBR    ← X X X X X X X X      Writes the transmit data.
SBICR1    ← 1 0 0 0 0 X X X      Starts transmission.

INTS interrupt

SBIDBR    ← X X X X X X X X      Writes the next transmit data.
```

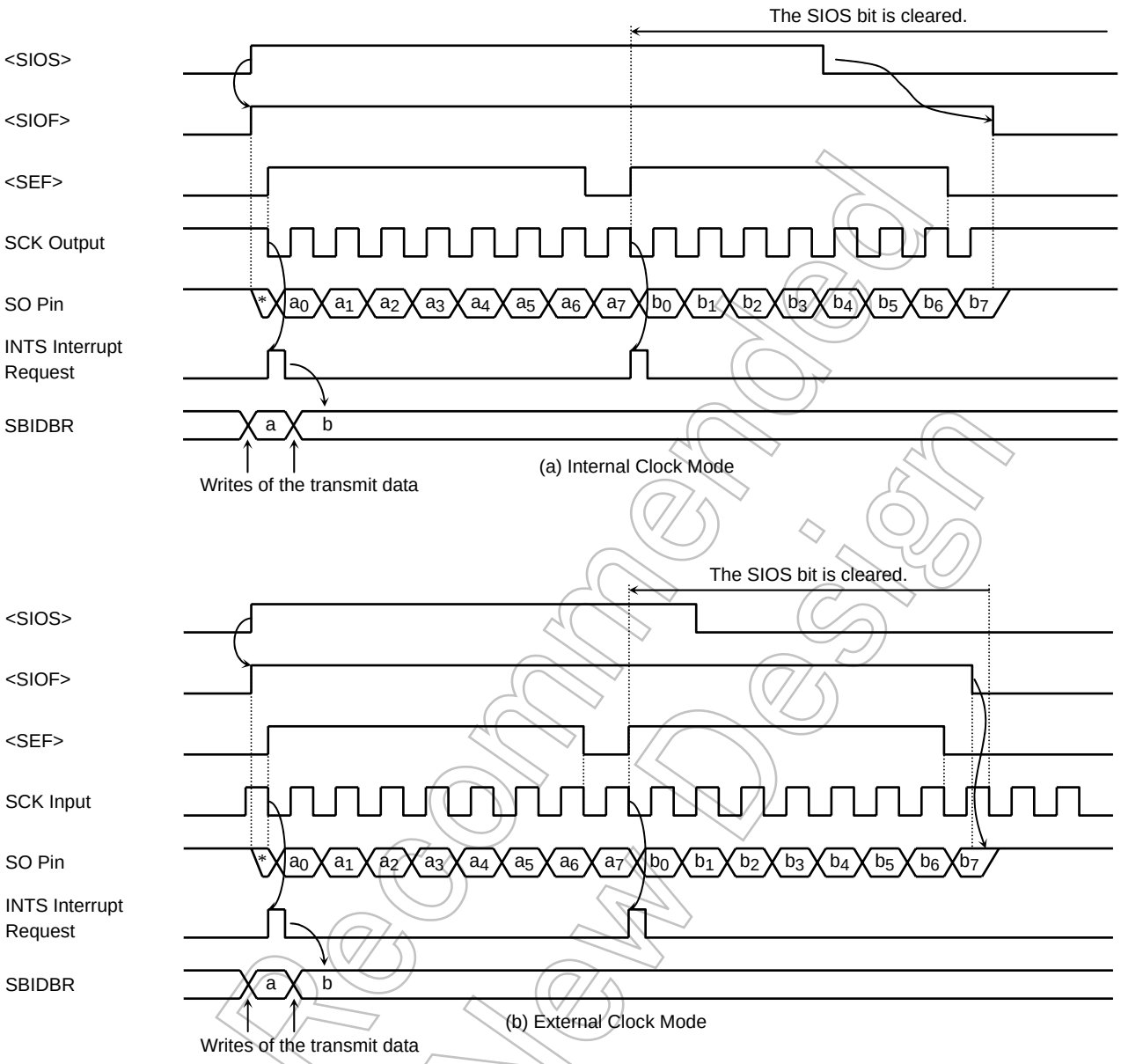


Figure 15.25 Transmit Mode

Example: MIP16 code to terminate transmission by SIOS (external clock mode)

```
        ADDIU   r3,     r0,     0x04
STEST1: LB      r2, (SBISR)                 ;  If SBISR<SEF> = 1 then loop
        AND     r2,     r3
        BNEZ    r2,     STEST1
        ADDIU   r3,     r0,     0x20
STEST2: LB      r2, (PA)                    ;  If SCK = 0 then loop
        AND     r2,     r3
        BEQZ    r2,     STEST2
        ADDIU   r3,     r0,     0y00000111
        STB     r3, (SBICR1)                ;  <SIOS> ← 0
```

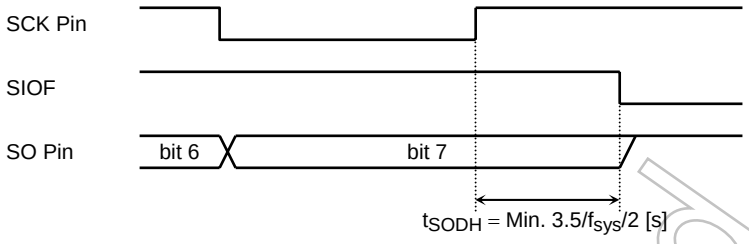


Figure 15.26 Retention Time of the Last Transmitted Bit

(2) 8-Bit Receive mode

Configure the SIO interface in Receive mode. Then setting the SIOS bit in the SBICR1 enables reception. The receive data is clocked into the internal shift register via the SI pin, with the least-significant bit (LSB) first, synchronous to the serial clock. Once the shift register is fully loaded, the received byte is transferred to the SBIDBR, and the buffer-full interrupt (INTS0) is generated. The INTS interrupt service routine must then pick up the received data from the SBIDBR.

In Internal Clock mode, the SIO interface will be in wait state (SCK will stop) until the INTS interrupt service routine reads the data from the SBIDBR.

In External Clock mode, shift operations continue, synchronous to the external clock. In this mode, the maximum data rate is a function of the maximum latency between when the INTS interrupt is generated and when the SBIDBR is read by the interrupt service routine.

Reception can be terminated by the INTS interrupt service routine clearing the SIOS bit to 0 or setting the SIOINH bit to 1. If the SIOS bit is cleared, reception continues until the shift register is fully loaded and transferred to the SBIDBR. In this case, software can check the SBISR.SIOF bit to determine whether reception has come to an end (0 = end-of-reception). If the SIOINH bit is set, the ongoing reception is aborted immediately, and the SIOF bit is cleared at that point. (The received data becomes invalid; there is no need to read it out.)

Note: The contents of the SBIDBR are not preserved after changing the transfer mode. Before changing the transfer mode, clear the SIOS bit to complete the ongoing reception and have the INTS interrupt service routine pick up the last received data.

```
                    7 6 5 4 3 2 1 0
SBICR1   ← 0 1 1 1 0 X X X          Selects receive mode.

SBICR1   ← 1 0 1 1 0 0 0 0          Starts reception.
```

INTS interrupt

```
Reg.     ← SBIDBR                   Reads the received data.
```

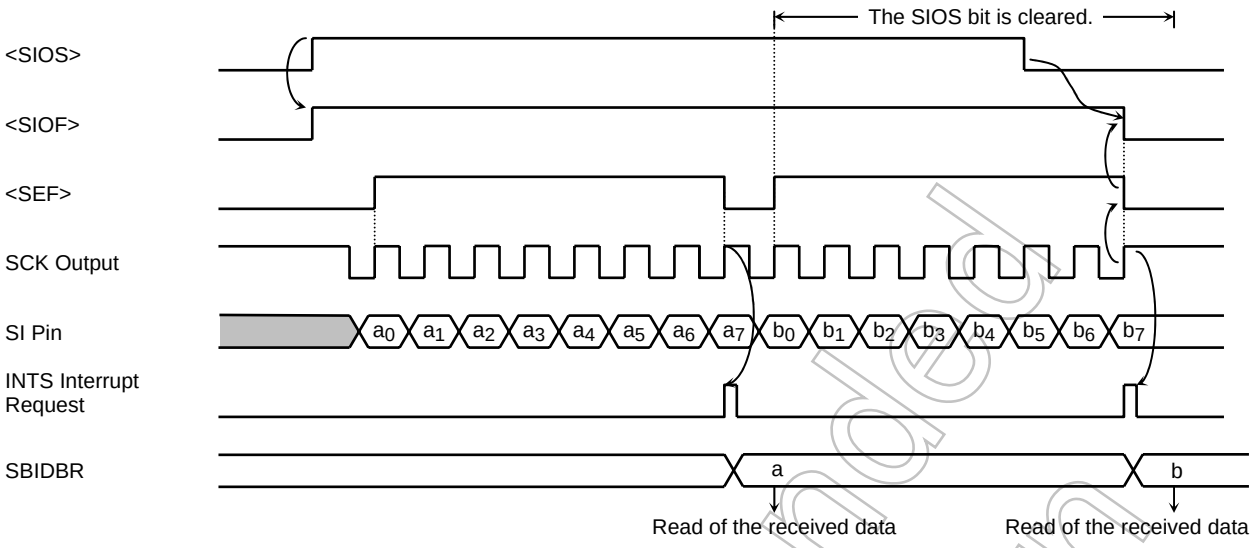


Figure 15.27 Receive Mode (Internal Clock Mode)

(3) 8-Bit Transmit/Receive mode

Configure the SIO interface in Transmit/Receive mode and write the transmit data into the SBIDBR. Then setting the SIOS bit in the SBICR1 initiates transmission and reception. The transmit data is shifted out through the SO pin, with the least-significant bit (LSB) first, with the falling edge of the serial clock, while at the same time the receive data is shifted in through the SI pin with the rising edge of the serial clock. Once the shift register is fully loaded with eight bits of the received data, it is transferred to the SBIDBR, and the INTS interrupt is generated. The INTS interrupt service routine must then pick up the received data from the SBIDBR and writes the next transmit data into the SBIDBR. Because the SBIDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In Internal Clock mode, the SIO interface will be in wait state (SCK will stop) after a read of the received data until a write of the transmit data.

In External Clock mode, shift operations continue, synchronous to the external clock. Therefore, software must read the received data and write the transmit data before the next shift operation begins. In this mode, the maximum data rate is a function of the maximum latency between when the INTS interrupt is generated and when the interrupt service routine reads the received data and writes the transmit data.

At the beginning of a transmission, the value of the last bit of the previously transmitted byte appears on the SO pin between when the SBISR.SIOF bit is set and when SCK subsequently goes low.

Transmission/reception can be terminated by the INTS interrupt service routine clearing the SIOS bit to 0 or setting the SIOINH bit to 1. If the SIOS bit is cleared, reception continues until the shift register is fully loaded and transferred to the SBIDBR. In this case, software can check the SBISR.SIOF bit to determine whether transmission/reception has come to an end (0 = end-of-reception/transmission). If the SIOINH bit is set, the ongoing transmission/reception is aborted immediately, and the SIOF bit is cleared at that point.

Note: The contents of the SBIDBR are not preserved after changing the transfer mode. Before changing the transfer mode, clear the SIOS bit to complete the ongoing transmission/reception and have the INTS interrupt service routine pick up the last received data.

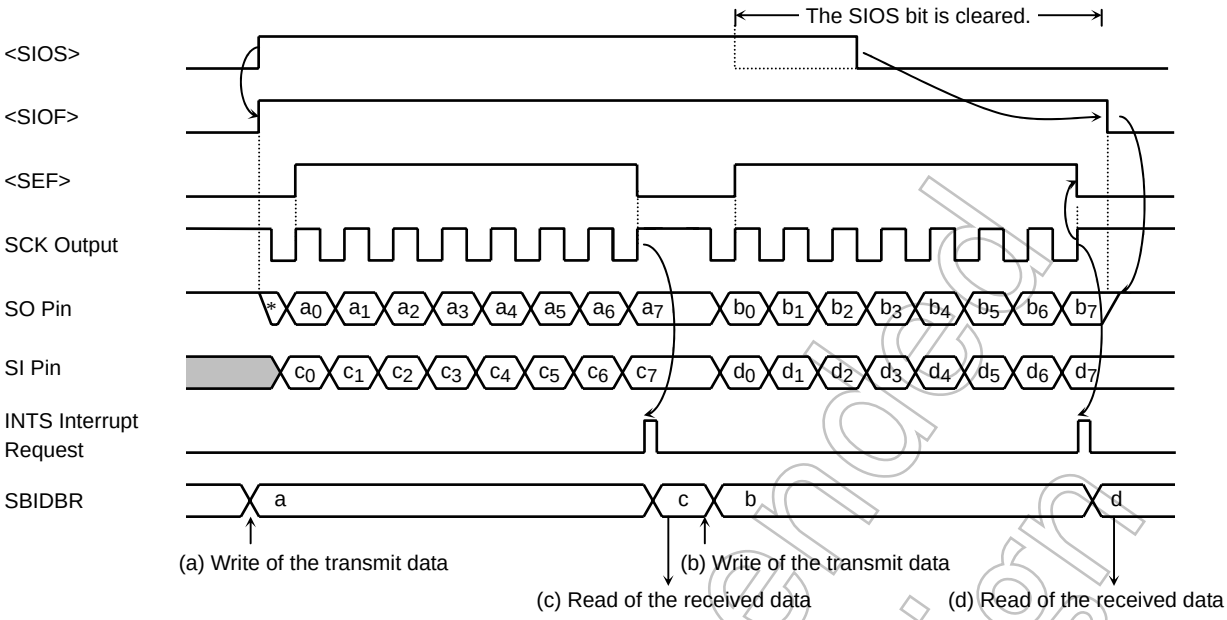


Figure 15.28 Receive/Transmit Mode (Internal Clock Mode)

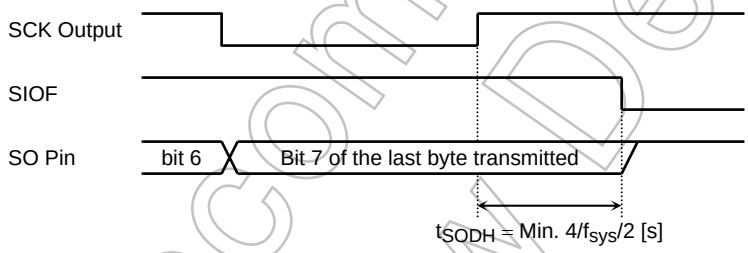


Figure 15.29 Retention Time of the Transmit Data in Receive/Transmit Mode

```
                    7 6 5 4 3 2 1 0
SBICR1   ← 0 1 1 0 0 X X X      Selects receive/transmit mode.

SBIDBR   ← X X X X X X X X      Writes the transmit data.
SBICR1   ← 1 0 1 0 0 X X X      Starts reception/transmission.
```

INTS interrupt

```
Reg.     ← SBIODBR              Reads the received data.
SBIDBR   ← X X X X X X X X      Writes the transmit data.
```

16. Analog-to-Digital Converter (ADC)

The TMP1962 has a 24-channel, multiplexed-input, 10-bit successive-approximation analog-to-digital converter (ADC).

Figure 16.1 shows a block diagram of the ADC. The 24 analog input channels (AN0-AN23) can be used as general-purpose digital inputs if not needed as analog channels.

Note: Please confirm the thing that the movement of the A/D converter has stopped when changing to IDLE and the STOP mode.

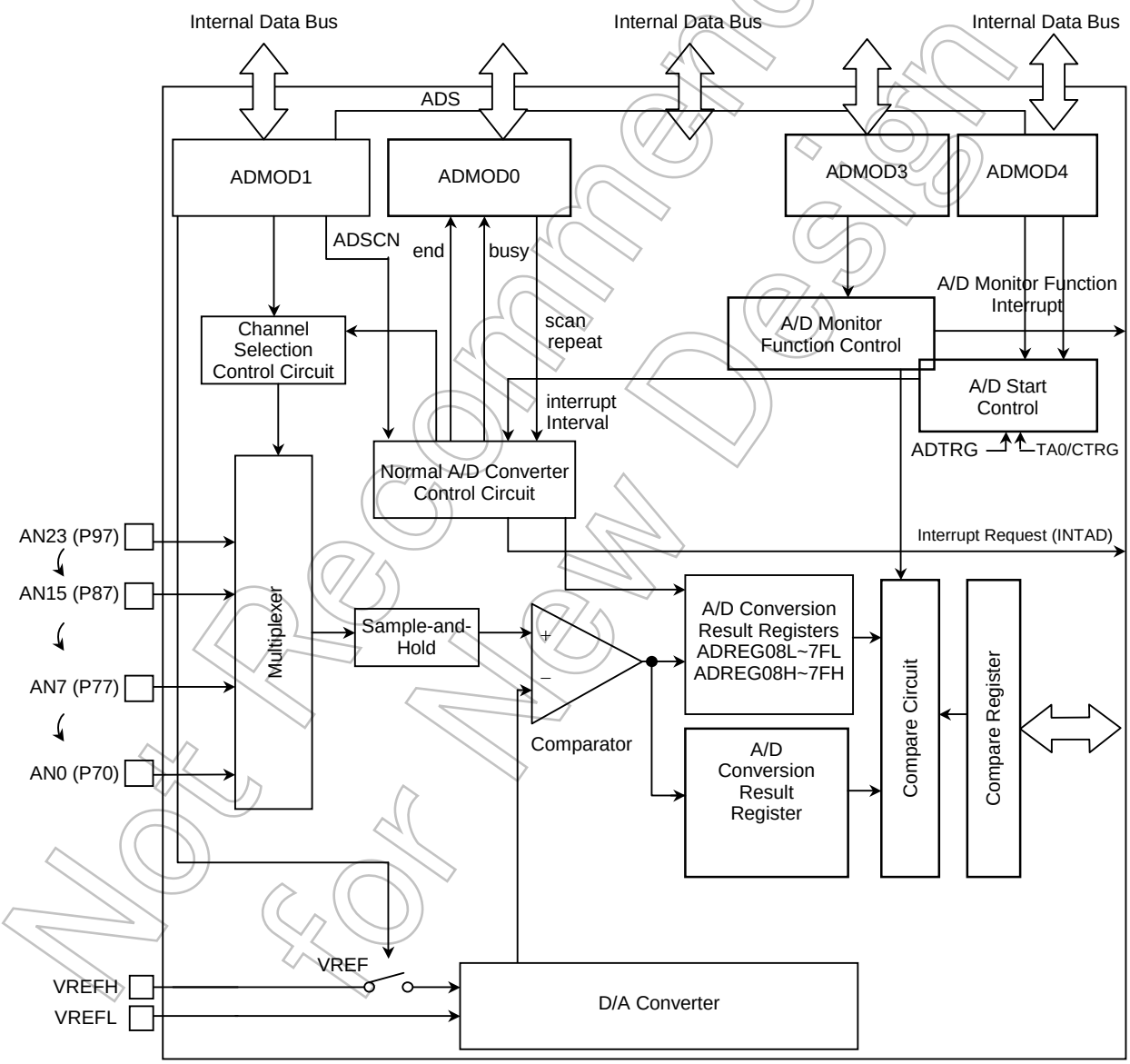


Figure 16.1 ADC Block Diagram

16.1 Register Description

The ADC has five mode control registers (ADMOD0,ADMMOD3,ADMOD4).

Figure 16.2 to Figure 16.6 show the registers available in the ADC.

A/D Mode Control Register 0

ADMOD0 (0xFFFF_F31B)

	7	6	5	4	3	2	1	0
Bit Symbol	EOCFN	ADBFN		ITM1	ITM0	REPEAT	SCAN	ADS
Read/Write	R			R/W				
Reset Value	0	0		0	0	0	0	0
Function	End-of-conversion flag 0: During conver-sion 1:Comp-leted	A/D conversion busy flag 0: Idle 1: During conver-sion		Interrupt in Fixed-Channel Continuous Conversion mode	Interrupt in Fixed-Channel Continuous Conversion mode	Continuous conversion mode 0: Single 1:Conti-nuous	Channel scan mode 0:Fixed-Channel 1:Channel Scan	A/D conversion start 0: Don't care 1: Start This bit is always read as 0.

Interrupt in Fixed-Channel Continuous Conversion mode

	Fixed-Channel Continuous Conversion Mode SCAN = 0, REPEAT = 1
00	Generates INTAD interrupt when a single conversion has been completed.
01	Generates INTAD interrupt when a sequence of four conversions has been completed.
10	Generates INTAD interrupt when a sequence of eight conversions has been completed.
11	Setting prohibited

Figure 16.2 A/D Conversion Registers

A/D Mode Control Register 1

ADMOD1 (0xFFFF_F31A)

	7	6	5	4	3	2	1	0
Bit Symbol	VREFON	I2AD	ADSCN	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	VREF control 0: Off 1: On	IDLE 0: Off 1: On	Channel scan mode 0: 4-channel 1: 8-channel	Analog input channel select				

Analog input channel select

<SCAN> <ADCH4,3.2, 1, 0>	0 Fixed-Channel Mode	1 Channel Scan Mode (ADSCN = 0)	1 Channel Scan Mode (ADSCN = 1)
0000	AN0	AN0	AN0
0001	AN1	AN0 to AN1	AN0 to AN1
0010	AN2	AN0 to AN2	AN0 to AN2
0011	AN3	AN0 to AN3	AN0 to AN3
0100	AN4	AN4	AN0 to AN4
0101	AN5	AN4 to AN5	AN0 to AN6
0110	AN6	AN4 to AN6	AN0 to AN6
0111	AN7	AN4 to AN7	AN0 to AN7
1000	AN8	AN8	AN8
1001	AN9	AN8 to AN9	AN8 to AN9
1010	AN10	AN8 to AN10	AN8 to AN10
1011	AN11	AN8 to AN11	AN8 to AN11
1100	AN12	AN12	AN8 to AN12
1101	AN13	AN12 to AN13	AN8 to AN13
1110	AN14	AN12 to AN14	AN8 to AN14
1111	AN15	AN12 to AN15	AN8 to AN15
10000	AN16	AN16	AN16
10001	AN17	AN16 to AN17	AN16 to AN17
10010	AN18	AN16 to AN18	AN16 to AN18
10011	AN19	AN16 to AN19	AN16 to AN19
10100	AN20	AN20	AN16 to AN20
10101	AN21	AN20 to AN21	AN16 to AN21
10110	AN22	AN20 to AN22	AN16 to AN22
10111	AN23	AN20 to AN23	AN16 to AN23

Note 1: Set the VREFON bit to 1 before a conversion is started, i.e., before setting the ADS bit in the ADMOD0 or before an external trigger is activated.

Note 2: If the TMP1962 will enter a standby mode upon the completion of A/D conversion, clear the VREFON bit to 0.

Figure 16.3 A/D Conversion Registers

A/D Mode Control Register 3

ADMOD3 (0xFFFF_F318)

	7	6	5	4	3	2	1	0
Bit Symbol			ADOBIC		REGS2	REGS1	REGS0	ADOBSV
Read/Write	R/W		R/W					
Reset Value	0		0	0	0	0	0	0
Function	Must be written as 0.		A/D monitor interrupt setting 0: Less than compare register 1: Greater than compare register	Must be written as 0.	A/D conversion result register to be compared with compare register when A/D monitor function is enabled			A/D monitor function 0: Disable 1: Enable

<REGS.2, 1, 0>	Target A/D Conversion Result Register
0000	ADREG08
0001	ADREG19
0010	ADREG2A
0011	ADREG3B
0100	ADREG4C
0101	ADREG5D
0110	ADREG6E
0111	ADREG7F

A/D Mode Control Register 4

ADMOD4 (0xFFFF_F31F)

	7	6	5	4	3	2	1	0
Bit Symbol			ADHS	ADHTG			ADRST1	ADRST0
Read/Write	R/W						W	W
Reset Value	0		0	0			—	—
Function	Must be written as 0.		Hardware trigger source conversion 0: External trigger 1: TA0TRG	Hardware trigger conversion 0: Disable 1: Enable			Software reset A write of 10 followed by a write of 01	

Note 1: When enabling an external resource to trigger A/D conversion, set the PI0F bit in the PIFC to 1, thus configuring the PI0 pin as $\overline{\text{ADTRG}}$, before setting the ADHTG bit. When using an 8-bit timer as a trigger, first set the ADHS bit to 1 when the timer is not operating. Then, set the ADHTG bit to enable trigger operation. Finally, operate the timer so that A/D conversion will be initiated at constant intervals.

Note 2: When disabling an external trigger ($\overline{\text{ADTRG}}$) for A/D conversion, first clear the ADHTG bit to 0.

A/D Conversion Result Low Register 08

ADREG08L (0xFFFF_F303)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR01	ADR00					OVR0	ADR0RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 08

ADREG08H (0xFFFF_F302)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							

A/D Conversion Result Low Register 19

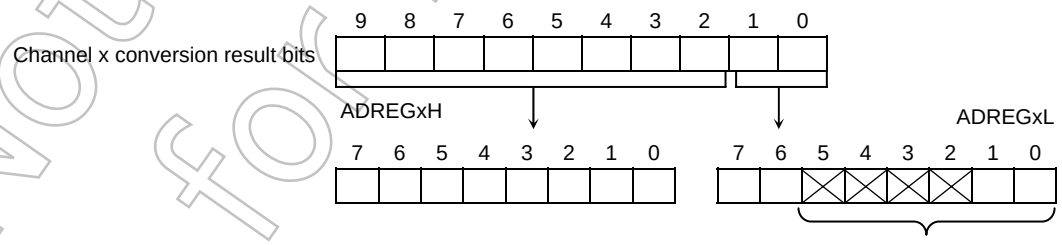
ADREG19L (0xFFFF_F301)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR11	ADR10					OVR1	ADR1RF
Read/Write	R						R	R
Reset Value	不定						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 19

ADREG19H (0xFFFF_F300)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							



- Bits 5-2 are always read as 1.
- Bit 0 (ADRxRF), when set, indicates that the conversion result has been stored in the ADREGxH/L register pair. This bit is cleared when the ADREGxL is read.
- Bit 1 (OVRx) indicates an overrun error. This bit is set if a next conversion result is written to the ADREGxH/L before both the ADREGxH and ADREGxL are read. Reading the flag causes it to be cleared.

Figure 16.4 A/D Conversion Registers

A/D Conversion Result Low Register 2A

ADREG2AL (0xFFFF_F307)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR21	ADR20					OVR2	ADR2RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 2A

ADREG2AH (0xFFFF_F306)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							

A/D Conversion Result Low Register 3B

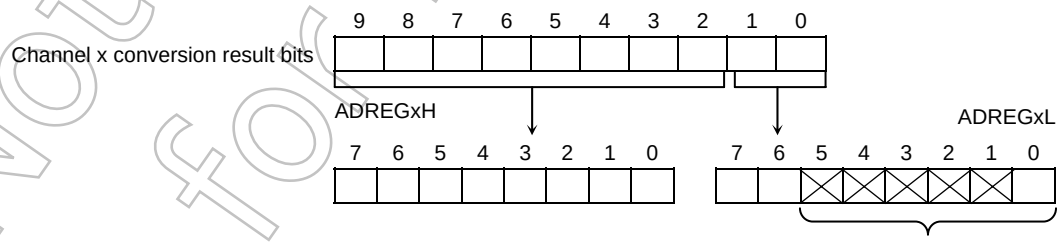
ADREG3BL (0xFFFF_F305)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR31	ADR30					OVR3	ADR3RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 3B

ADREG3BH (0xFFFF_F304)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							



- Bits 5-2 are always read as 1.
- Bit 0 (ADRxRF), when set, indicates that the conversion result has been stored in the ADREGxH/L register pair. This bit is cleared when the ADREGxL is read.
- Bit 1 (OVRx) indicates an overrun error. This bit is set if a next conversion result is written to the ADREGxH/L before both the ADREGxH and ADREGxL are read. Reading the flag causes it to be cleared.

Figure 16.5 A/D Conversion Registers

A/D Conversion Result Low Register 4C

ADREG4CL (0xFFFF_F30B)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR41	ADR40					OVR4	ADR4RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 4C

ADREG4CH (0xFFFF_F30A)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							

A/D Conversion Result Low Register 5D

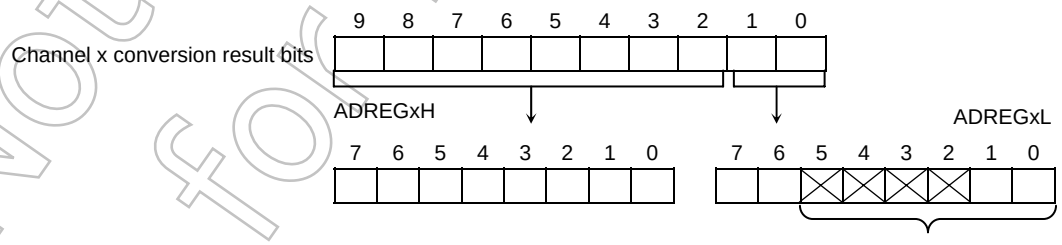
ADREG5DL (0xFFFF_F309)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR51	ADR50					OVR5	ADR5RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 5D

ADREG5DH (0xFFFF_F308)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							



- Bits 5-2 are always read as 1.
- Bit 0 (ADRxRF), when set, indicates that the conversion result has been stored in the ADREGxH/L register pair. This bit is cleared when the ADREGxL is read.
- Bit 1 (OVRx) indicates an overrun error. This bit is set if a next conversion result is written to the ADREGxH/L before both the ADREGxH and ADREGxL are read. Reading the flag causes it to be cleared.

A/D Conversion Result Low Register 6E

ADREG6EL (0xFFFF_F30F)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR61	ADR60					OVR6	ADR6RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 6E

ADREG6EH (0xFFFF_F30E)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							

A/D Conversion Result Low Register 7F

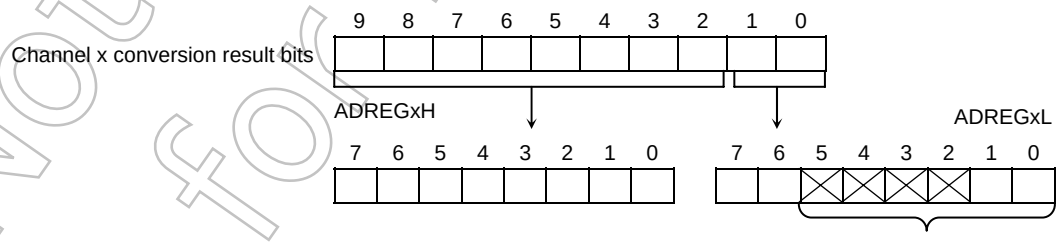
ADREG7FL (0xFFFF_F30D)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR71	ADR70					OVR7	ADR7RF
Read/Write	R						R	R
Reset Value	Undefined						0	0
Function	Lower 2 bits of an A/D conversion result						Overrun flag 0: No overrun 1: Overrun	Conversion result store flag 1: Stored

A/D Conversion Result High Register 7F

ADREG7FH (0xFFFF_F30C)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							



- Bits 5-2 are always read as 1.
- Bit 0 (ADRxRF), when set, indicates that the conversion result has been stored in the ADREGxH/L register pair. This bit is cleared when the ADREGxL is read.
- Bit 1 (OVRx) indicates an overrun error. This bit is set if a next conversion result is written to the ADREGxH/L before both the ADREGxH and ADREGxL are read. Reading the flag causes it to be cleared.

A/D Conversion Result Compare Low Register

ADCOMREGL (0xFFFF_F317)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR21	ADR20						
Read/Write	R/W							
Reset Value	0							
Function	Lower 2 bits of a value to be compared with an A/D conversion result							

A/D Conversion Result Compare High Register

ADCOMREGH (0xFFFF_F316)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R/W							
Reset Value	0							
Function	Upper 8 bits of a value to be compared with an A/D conversion result							

Note: Disable the A/D monitor function (set ADMOD3.ADOBSV to 0) before attempting to set or modify the contents of these registers.

A/D Conversion Clock Setting Register

ADCLK (0xFFFF_F31C)

	7	6	5	4	3	2	1	0
Bit Symbol						ADCLK2	ADCLK1	ADCLK0
Read/Write						R/W	R/W	R/W
Reset Value						0	0	0
Function						A/D prescaler output 000: fc 1XX: fc/16 001: fc/2 010: fc/4 011: fc/8		

Note 1: A/D conversion is performed at the clock frequency selected in the above register. To assure conversion accuracy, however, the conversion clock frequency must not exceed 20.25 MHz.

Fc = 40.5MHz <ADCLK2:0> = "001" → 7.95 μsec

Note 2: Do not change the clock frequency while A/D conversion is in progress.

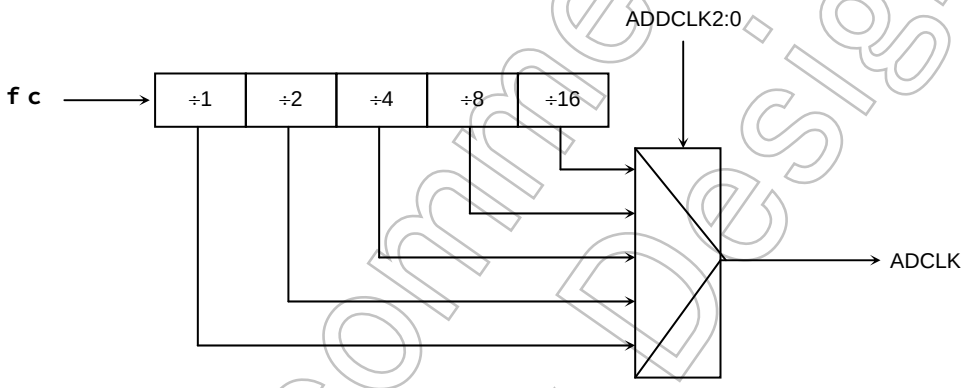


Figure 16.6 A/D Conversion Registers

16.2 Operation

16.2.1 Analog Reference Voltages

The VREFH and VREFL pins provide the reference voltages for the ADC. Clearing the VREFON bit in the ADMOD1 turns off the switch between VREFH and VREFL. Once the VREFON bit is cleared, the internal reference voltage requires a recovery time of 3 μs (T.B.D.) to stabilize after the VREFON bit is again set to 1. The ADS bit in the ADMOD0 must then be set to initiate an conversion.

- Fixed-Channel mode (ADMOD0.SCAN = 0)

  When the SCAN bit in the ADMOD0 is cleared, the ADC runs conversions on a single input channel selected from AN0-AN23 via the ADCH[4:0] field in the ADMOD1.

- Channel Scan mode (ADMOD0.SCAN = 1)

  When the SCAN bit in the ADMOD0 is set, the ADC runs conversions on sequential channels in a specific group selected via the ADCH[4:0] field in the ADMOD1.

After a reset, the ADMOD0.SCAN bit defaults to 0, and the ADMOD1.ADCH[3:0] field defaults to 0000. Thus, the AN0 pin is selected as the conversion channel. The AN0-AN23 pins can be used as general-purpose input ports if not used as analog input channels.

16.2.2 Starting an A/D Conversion

The ADC initiates conversion when the ADS bit in the ADMOD0 is set. The ADHTG bits in the ADMOD4 enable a hardware trigger source for conversion, respectively. When the ADHS bit in the ADMOD4 is cleared to 0, conversion is triggered by a falling edge applied to $\overline{\text{ADTRG}}$ pin. When the ADHS bit is set to 1, conversion is triggered by a TA0TRG output from 8-Bit Timer 0.

When conversion starts, the busy flag (ADMOD0.ADBF) is set.

.

16.2.3 Conversion Modes and Conversion-Done Interrupts

The ADC supports the following four conversion modes. For a normal A/D conversion, the REPEAT and SCAN bits in the ADMOD0 select one of the four conversion modes. For a high-priority A/D conversion, the ADC only supports Fixed-Channel Single Conversion mode, regardless of the settings of the REPEAT and SCAN bits.

- Fixed-Channel Single Conversion mode
- Channel Scan Single Conversion mode
- Fixed-Channel Continuous Conversion mode
- Channel Scan Continuous Conversion mode

The REPEAT and SCAN bits in the ADMOD0 select the conversion mode. Once a conversion is started, the ADBFN bit in the ADMOD0 is set to 1. The ADC generates the INTAD interrupt and sets the EOCF bit in the ADMOD0 at the end of the specified conversion process. If REPEAT = 0, the ADBFN bit is cleared when the ADC sets the EOCF bit. If REPEAT = 1, the ADC continues conversion without clearing ADBFN.

1) Fixed-Channel Single Conversion mode

This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 00. In this mode, the ADC performs a single conversion on a single selected channel. When a conversion is completed, the ADC sets the ADMOD0.EOCF bit, clears the ADMOD0.ADBF bit and generates the INTAD interrupt. The EOCF bit is cleared when it is read.

2) Channel Scan Single Conversion mode

This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 01. In this mode, the ADC performs a single conversion on each of a selected group of channels. When a single conversion sequence is completed, the ADC sets the ADMOD0.EOCF bit, clears the ADMOD0.ADBF bit and generates the INTAD interrupt. The EOCF bit is cleared when it is read.

3) Fixed-Channel Continuous Conversion mode

This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 10. In this mode, the ADC repeatedly converts a single selected channel. When a conversion process is completed, the ADC sets the ADMOD.EOCF bit. The ADMOD0.ADBF bit remains set. The ITM[1:0] bits in the ADMOD0 control interrupt generation in this mode. The timing when the EOCF bit is set also depends on the ITM[1:0] bits. The EOCF bit is cleared when it is read.

If the ITM[1:0] field is set to 00, the ADC generates an interrupt after each conversion. The results of conversion are always stored in the ADREG08 register pair. The EOCF bit is set when the ADC stores the results in the ADREG08.

If the ITM[1:0] field is set to 01, the ADC generates an interrupt after every four conversions. The results of conversions are sequentially stored in the ADREG08 to ADREG3B register pairs, in that order. The EOCF bit is set when the ADC stores the results in the ADREG3B. The next conversion results are again stored in the ADREG08, and so on. The EOCF bit is cleared when it is read.

If the ITM[1:0] field is set to 10, the ADC generates an interrupt after every eight conversions. The results of conversions are sequentially stored in the ADREG08 to ADREG7F register pairs, in that order. The EOCF bit is set when the ADC stores the results in the ADREG7F. The next conversion results are again stored in the ADREG08, and so on. The EOCF bit is cleared when it is read.

4) Channel Scan Continuous Conversion mode

This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 11. In this mode, the ADC repeatedly converts the selected group of channels. When a single conversion sequence is completed, the ADC sets the ADMOD0.EOCF bit and generates the INTAD interrupt. The ADMOD0.ADBF bit remains set. The EOCF bit is cleared when it is read.

In continuous conversion modes (3) and 4)), clearing the ADMOD0.REPEAT bit stops the conversion sequence after the ongoing conversion process is completed. The ADMOD0.ADBF bit is cleared.

Before putting the TMP1962 in any standby mode (IDLE or STOP), check the ADC is being disabled (or disable the ADC).

Interrupt Request Generation and Flag Setting in Each A/D Conversion Mode

Mode	Interrupt Request Generation	EOCF Set Timing (Note)	ADBF (Upon Generation of Interrupt)	ADMOD0		
				ITM1:0	REPEAT	SCAN
Fixed-Channel Single Conversion Mode	After a conversion	After a conversion	0		0	0
Fixed-Channel Continuous Conversion Mode	After every conversion	After every conversion	1	00	1	0
	After every four conversions	After every four conversions	1	01		
	After every eight conversions	After every eight conversions	1	10		
Channel Scan Single Conversion Mode	After a scan conversion sequence	After a scan conversion sequence	0		0	1
Channel Scan Continuous Conversion Mode	After each scan conversion sequence	After each scan conversion sequence	1		1	1

Note: EOCF is cleared when it is read.

16.2.4 High-Priority Conversion Mode

The ADC can perform a high-priority A/D conversion while it is performing a normal A/D conversion sequence. A high-priority A/D conversion can be initiated by setting the HPADCE bit in the ADMOD2 to 1. It is also triggered by a hardware resource if so enabled using the HADHTG and HADHS bits in the ADMOD4. If a high-priority conversion is triggered during a normal conversion, the ADC stores the results of conversion for the current channel and then begins a single high-priority conversion for the channel specified with the HPADCH[4:0] bits in the ADMOD2. Upon the completion of the high-priority conversion, the ADC stores the results of the conversion in the ADREGSP, generates the end-of-high-priority-conversion interrupt, and then resumes the suspended normal conversion with the next channel. While a high-priority conversion is being performed, a trigger for another high-priority conversion is ignored.

For example, suppose the ADC is performing conversions for AN0-AN8 in Channel Scan Continuous Conversion mode. If the HPADCE bit is set to 1 while the ADC is converting data for AN3, it completes conversion for AN3 and then converts data for the channel specified with HPADCH[4:0]. After storing the results of conversion in the ADREGSP, the ADC resumes the suspended normal conversion sequence, beginning with conversion for AN4.

16.2.5 A/D Monitor Function

When the ADOBSV bit in the ADMOD3 is set to 1, the A/D monitor function is enabled. This function generates an interrupt if the value stored in the specified A/D conversion result register pair (specified with the REGS[3:0] bits in the ADMOD3) is greater or less (depending on ADMOD3.ADOBIC) than the contents of the compare register pair. The ADC performs this comparison each time it stores results to the specified register pair. The conversion result register pair used for the A/D monitor function is usually not read in the program, so that its overrun flag (OVRn) and conversion result storage flag (ADRnRF) are always set. When using the A/D monitor function, therefore, do not use flags for the register pair assigned for comparison.

16.2.6 Conversion Time

The A/D conversion clock can be selected from A/D prescaler output ϕT0/2, ϕT0/4, ϕT0/8, ϕT0/16 and ϕT0/32 through the programming of the ADCLK[2:0] field in the ADCLK register. To assure conversion accuracy, the conversion clock frequency must not exceed 20.25 MHz, i.e., conversion time must be no shorter than 7.95 μs.

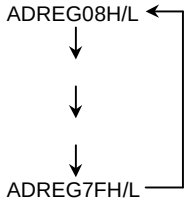
16.2.7 Storing and Reading the A/D Conversion Result

Conversion results are loaded into conversion result high/low register pairs (ADREG08H/L to ADREG7FH/L).

In Fixed-Channel Continuous Conversion mode, conversion data goes into the ADREG08H/L to ADREG7FH/L sequentially. If the ITM[1:0] field is set to 00, so that the ADC generates an interrupt after each conversion, conversion data is stored in the ADREG08H/L only. If the ITM[1:0] field is set to 01, so that the ADC generates an interrupt after every four conversions, conversion data goes into the ADREG08H/L to ADREG3BH/L sequentially.

Table 16.1 shows the relationships between the analog input channels and the A/D conversion result registers.

Table 16.1 Relationships Between Analog Input Channels and A/D Conversion Result Registers

Analog Input Channel (Port A)	A/D Conversion Result Registers	
	Modes Other Than Fixed-Channel Continuous Conversion Mode	Fixed-Channel Continuous Conversion Mode (for each sequence of eight conversions)
AN0	ADREG08H/L	
AN1	ADREG19H/L	
AN2	ADREG2AH/L	
AN3	ADREG3BH/L	
AN4	ADREG4CH/L	
AN5	ADREG5DH/L	
AN6	ADREG6EH/L	
AN7	ADREG7FH/L	
AN8	ADREG08H/L	
AN9	ADREG19H/L	
AN10	ADREG2AH/L	
AN11	ADREG3BH/L	
AN12	ADREG4CH/L	
AN13	ADREG5DH/L	
AN14	ADREG6EH/L	
AN15	ADREG7FH/L	
AN16	ADREG08H/L	
AN17	ADREG19H/L	
AN18	ADREG2AH/L	
AN19	ADREG3BH/L	
AN20	ADREG4CH/L	
AN21	ADREG5DH/L	
AN22	ADREG6EH/L	
AN23	ADREG7FH/L	

16.2.8 Data Polling

When the results of A/D conversion are processed by means of data polling without using interrupts, the EOCF bit in the ADMOD0 should be polled. If this flag is set, the specified A/D conversion result register pairs contain results. Then, read those registers. To detect an overrun, first read the ADREGxH and then read the ADREGxL. If the OVRn is cleared to 0 and ADRnRF is set to 1 in the ADREGxL, the register pair contains valid conversion results.

17. Watchdog Timer (WDT)

The TMP1962 contains a watchdog timer (WDT). The WDT is used to regain control of the system in the event of software or system lockups due to spurious noises, etc. When a watchdog timer time-out occurs, the WDT generates a nonmaskable interrupt to the CPU.

Also, the time-out event can be programmed for system reset generation, which is accomplished by routing the time-out signal to the internal reset pin.

17.1 Implementation

Figure 17.1 shows a block diagram of the WDT.

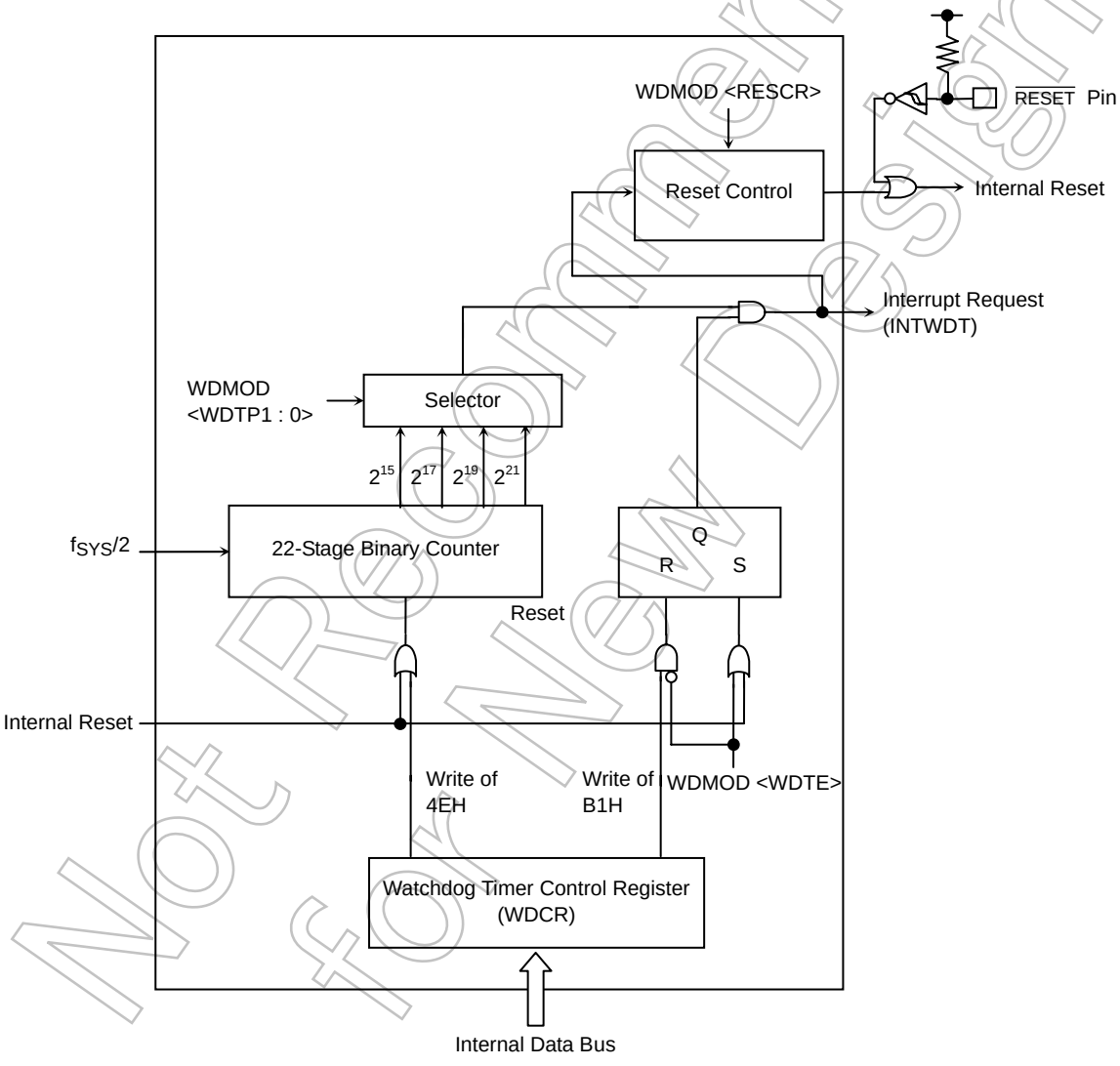


Figure 17.1 WDT Block Diagram

The WDT contains a 22-stage binary counter clocked by the $f_{SYS}/2$ clock. This binary counter provides 2^{15}, 2^{17}, 2^{19} or 2^{21} as a counter overflow signal, as programmed into the WDTP[1:0] field in the WDMOD. When a counter overflow occurs, the WDT generates a WDT interrupt, as shown below.

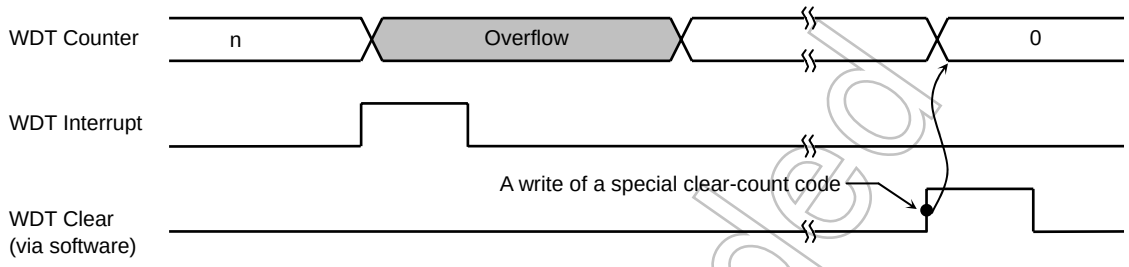


Figure 17.2 Default Operation

Also, the counter overflow can be programmed to cause a system reset as the time-out action. If so programmed, a counter overflow causes the WDT to assert the internal reset signal for a 22- to 29-state time. After a reset, the f_{SYS} clock is generated by dividing the high-speed oscillator clock (fc) by eight through the clock gear function (when the PLL is used); the WDT clock source ($f_{SYS}/2$) is derived from this f_{SYS} clock.

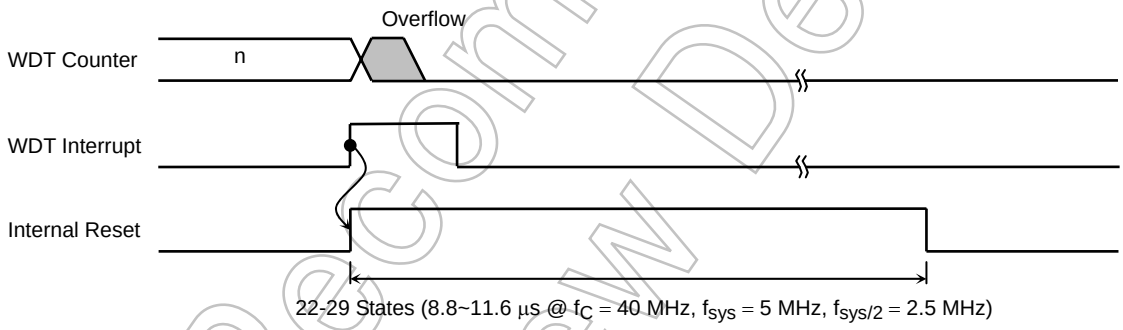


Figure 17.3 Reset Operation

Note: The TMP1962 continues sampling the $\overline{\text{PLLOFF}}$ pin during a reset operation caused by the WDT. Therefore, the $\overline{\text{PLLOFF}}$ pin must be tied to either logic high or logic low.

17.2 Register Description

The WDT is controlled by two registers called WDMOD and WDCR.

17.2.1 Watchdog Timer Mode Register (WDMOD)

(1) Time-out period (WDMOD.WDTP[1:0])

This 2-bit field determines the duration of the WDT time-out interval. Upon reset, the WDTP[1:0] field defaults to 00. Figure 17.4 shows possible time-out periods.

(2) WDT enable (WDMOD.WDTE)

Upon reset, the WDTE bit is set to 1, enabling the WDT. To disable the WDT, the clearing of the WDTE bit must be followed by a write of a special key code (B1H) to the WDCR register. This prevents a "lost" program from disabling the WDT operation. The WDT can be re-enabled simply by setting the WDTE bit.

(3) System reset (WDMOD.RESCR)

This bit is used to program the WDT to generate a system reset on a time-out. Upon reset, this bit is cleared; thus the time-out does not cause a system reset.

17.2.2 Watchdog Timer Control Register (WDCR)

This register is used to disable the WDT and to clear the WDT binary counter.

- Disabling the WDT

  The WDT can be disabled by clearing the WDMOD.WDTE to 0 and then writing the special disable code (B1H) to the WDCR register.

```
WDMOD  ← 0 – – – – – – –    Clears the WDTE bit to 0.
WDCR   ← 1 0 1 1 0 0 0 1    Writes the disable code (B1H) to the WDCR.
```

- Enabling the WDT

  The WDT can be enabled simply by setting the WDTE bit in the WDMOD to 1.

- Clearing the WDT counter

  Writing the special clear-count code (4EH) to the WDCR resets the binary counter to zero. The counting process begins again.

```
WDCR   ← 0 1 0 0 1 1 1 0    Writes the clear-count code (4EH) to the WDCR.
```

Note: Writing the disable code (BIH) clears the binary counter.

WDMOD (0xFFFF_F093)

	7	6	5	4	3	2	1	0
Bit Symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	—
Read/Write	R/W	R/W				R/W		R/W
Reset Value	1	0	0			0	0	0
Function	WDT enable 1: Enable	Time-out period 00: $2^{16}/f_{SYS}$ 01: $2^{18}/f_{SYS}$ 10: $2^{20}/f_{SYS}$ 11: $2^{22}/f_{SYS}$				IDLE 0: Off 1: On	1: System reset by WDT	Must be written as 0.

System reset

0	—
1	Internally routes the WDT time-out signal to the system reset

Time-out period @ fc = 40.5 MHz

Clock Gear Value SYSCR1. GEAR[1:0]	Watchdog Timer Time-out Period WDMOD<WDTP1, 0>			
	00	01	10	11
00 (fc)	1.6 ms	6.5 ms	25.9 ms	104 ms
01 ($fc/_2$)	3.2 ms	12.9 ms	51.8 ms	207 ms
10 ($fc/_4$)	6.5 ms	25.9 ms	104 ms	414 ms
11 ($fc/_8$)	12.9 ms	51.8 ms	207 ms	829 ms

WDT enable

0	Disable
1	Enable

Figure 17.4 Watchdog Timer Mode Register

WDCR (0xFFFF_F092)

	7	6	5	4	3	2	1	0
Bit Symbol	—							
Read/Write	W							
Reset Value	—							
Function	B1H: WDT disable code 4EH: WDT clear-count code							

Special code

B1H	WDT disable code
4EH	WDT clear-count code
Other values	—

Figure 17.5 Watchdog Timer Control Register

17.3 Operation

The watchdog timer is a kind of timer that generates an interrupt request if it times out. The WDT allows the user to program the time-out period in the WDTP[1:0] field in the WDMOD register. While enabled, the software can reset the counter to zero at any time by writing a special clear-count code. If the software is unable to reset the counter before it reaches the time-out count, the WDT generates the INTWD interrupt. In response to the interrupt, the CPU jumps to a system recovery routine to regain control of the system. The WDT can also output a time-out signal to a peripheral device so that the device can respond to the problem.

The WDT begins counting immediately after reset.

When the TMP1962 goes into STOP mode, the WDT counter is reset to zero automatically and stops counting. The WDT continues counting while an off-chip peripheral has mastership of the bus (i.e., $\overline{\text{BUSAK}}$ = 0). In IDLE mode, the I2WDT bit in the WDMOD determines whether or not to disable the WDT. The I2WDT bit can be programmed before putting the TMP1962 in IDLE mode.

Examples:

(1) Clearing the WDT binary counter

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Writes the clear-count code (4EH) to the WDCR.

(2) Programming the time-out interval to $2^{18}/f_{SYS}$

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	1	–	–	–	–	–

(3) Disabling the watchdog timer

		7	6	5	4	3	2	1	0	
WDMOD	←	0	–	–	–	–	–	–	–	Clears the WDTE bit to 0.
WDCR	←	1	0	1	1	0	0	0	1	Writes the disable code (B1H) to the WDCR.

18. Key-Pressed Wake-up

18.1 Outline

- The TMP1962 has 14 key input channels (KEY0-KEYD) that enable the pressing of a key to terminate STOP mode or trigger an external interrupt. These 14 interrupts are, however, assumed as the same interrupt source (as specified in the CG block) when sent to the Interrupt Controller (INTC). Each key input can be enabled or disabled individually using the KWUPSTn register.
- The interrupt sensitivity and polarity (rising-edge triggered, falling-edge triggered, high-level sensitive or low-level sensitive) can be specified individually for each key input using the KWUPSTn register.
- The interrupt service routine clears the key interrupt request through the KWUPCLR register.
- Each key input pin has an internal pull-up resistor, which can be enabled or disabled by programming bit 0 (PE) of the KWUPCNT. The settings of the PE and DPE bits apply to all 14 key inputs.

18.2 Operation

The TMP1962 has 14 key input pins (KEY0-KEYD). The KWUPEN bit in the CG's IMCGB1 register controls whether the key inputs are used to exit STOP mode or used as general-purpose interrupt sources. When KWUPEN is set to 1, all of KEY0-KEYD are used for STOP wake-up signaling. For each key input, the KEYnEN bit in the KWUPSTn must be programmed to either enable or disable interrupts and the KEYn[1:0] field in the same register must be programmed to specify signal sensitivity. The KWUP circuit block detects key inputs and transmits the results to the IMCGB1 register in the CG, identifying the high level as an active state. The EMCG[51:50] bits in the IMCGB1 must be set to 01 (high-level sensitive). The CG in turn transmits the results to the INTC, also identifying the high level as an active state. The INTC must also be programmed so that the corresponding interrupt is high-level sensitive (01). When the KWUPEN bit in the IMCGB1 register is cleared to 0 (default), all of KEY0 to KEYD are used as general-purpose interrupts. In that case, the CG does not need to be programmed; only the INTC must be programmed so that the interrupt is high-level sensitive. The KWUPSTn must also be programmed to enable interrupts and specify signal sensitivity. In the key interrupt service routine, writing 1010 to the KWUPCLR causes all key interrupt requests to be cleared.

Note: If another key input is detected before the interrupt service routine clears the interrupt request corresponding to the first key input, the routine clears all interrupt requests simultaneously. If another key input is detected after the interrupt service routine clears the interrupt request corresponding to the first key input, the new key input triggers another key interrupt.

18.3 Pull-up Resistors

Each key input pin has an internal pull-up resistor. When the KYPE bit in the KWUPCNT is set to 1, pull-up resistors for all key input pins (KEY0-KEYD) are enabled, except for the pins for which key input is disabled by clearing the KEYnEN bit in the KWUPSTn.

(1) When using key input pins in static pull-up mode, the following procedures must be observed:

- Initial setup after power-on

1) Program the KWUPCNT (KYPE = 1).
2) Set the KWUPSTn.KEYnEN bit to 1 for each key input to be used.
3) Wait until the pull-up resistors are disabled.
4) Specify an interrupt trigger in the KWUPSTn for each key input to be used.
5) Clear the interrupt request with KWUPCLR.
6) Program the CG and INTC, as described in Chapter 6, "Interrupts."

- Modifying an interrupt trigger for key inputs

1) Disable key interrupts in the INTC (IMC1.IL6[2:0] = 000).
2) Modify an interrupt trigger in the KWUPSTn for each relevant key input.
3) Clear the interrupt request with KWUPCLR.
4) Enable key interrupts in the INTC (program IMC1.IL6[2:0] as required).

- Enabling additional key inputs

1) Disable key interrupts in the INTC (IMC1.IL6[2:0] = 000).
2) Set the KWUPSTn.KEYnEN bit to 1 for each key input to be used.
3) Wait until the pull-up resistors are disabled.
4) Specify an interrupt trigger in the KWUPSTn for each key input to be used.
5) Clear the interrupt request with KWUPCLR.
6) Enable key interrupts in the INTC (program IMC1.IL6[2:0] as required).

(2) When using key input pins in dynamic pull-up mode, the following procedures must be observed:

- Initial setup after power-on

1) Program the KWUPCNT (KYPE = 1, TnSn = desired interval).
2) Specify an interrupt trigger in the KWUPSTn for each key input to be used.
3) Clear the interrupt request with KWUPCLR.
4) Set the KWUPSTn.KEYnEN bit to 1 for each key input to be used.
5) Program the CG and INTC, as described in Chapter 6, "Interrupts."

- Modifying an interrupt trigger for key inputs

1) Disable key interrupts in the INTC (IMC1.IL6[2:0] = 000).
2) Modify an interrupt trigger in the KWUPSTn for each relevant key input.
3) Clear the interrupt request with KWUPCLR.
4) Enable key interrupts in the INTC (program IMC1.IL6[2:0] as required).

- Enabling additional key inputs

1) Disable key interrupts in the INTC (IMC1.IL6[2:0] = 000).
2) Specify an interrupt trigger in the KWUPSTn for each key input to be used.
3) Clear the interrupt request with KWUPCLR.
4) Set the KWUPSTn.KEYnEN bit to 1 for each key input to be used.
5) Enable key interrupts in the INTC (program IMC1.IL6[2:0] as required).

(3) When using key input pins without enabling pull-up resistors, the following procedures must be observed:

- Initial setup after power-on

1) Program the KWUPCNT (KYPE = 1).
2) Specify an interrupt trigger in the KWUPSTn for each key input to be used.
3) Clear the interrupt request with KWUPCLR.
4) Set the KWUPSTn.KEYnEN bit to 1 for each key input to be used.
5) Program the CG and INTC, as described in Chapter 6, "Interrupts."

- Modifying an interrupt trigger for key inputs

1) Disable key interrupts in the INTC (IMC1.IL6[2:0] = 000).
2) Modify an interrupt trigger in the KWUPSTn for each relevant key input.
3) Clear the interrupt request with KWUPCLR.
4) Enable key interrupts in the INTC (program IMC1.IL6[2:0] as required).

- Enabling additional key inputs

1) Disable key interrupts in the INTC (IMC1.IL6[2:0] = 000).
2) Specify an interrupt trigger in the KWUPSTn for each key input to be used.
3) Clear the interrupt request with KWUPCLR.
4) Set the KWUPSTn.KEYnEN bit to 1 for each key input to be used.
5) Enable key interrupts in the INTC (program IMC1.IL6[2:0] as required).

Key-Pressed Wake-up Control Register: KWUPCNT

	7	6	5	4	3	2	1	0
(0xFFFF_F372) Bit Symbol								KYPE
Read/Write	R/W		R/W					
Reset Value	0		0	0	0	0	0	0
Function	Must be written as 0.		Must be set to 00.		Must be set to 00.		Must be set to 0.	0: Disable pull-up 1: Enable pull-up

18.4 Key Input Detection Timing

(1) When pull-up registers are disabled (KYPE = 0)

For each key input, the KEYn[1:0] bits in the KWUPSTn register can specify one of four interrupt trigger types: high level, low level, rising edge and falling edge. The states of key inputs are always monitored.

(2) When pull-up registers are enabled (KYPE = 1)

For each key input, the KEYn[1:0] bits in the KWUPSTn register can specify one of four interrupt trigger types: high level, low level, rising edge and falling edge. The states of key inputs are always monitored.

		7	6	5	4	3	2	1	0
KWUPST0 (0xFFFF_F363)	Bit Symbol			KEY01	KEY00				KEY0EN
	Read/Write			R/W					R/W

	Reset Value			1	0				0
	Function			KEY0 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY0 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST1	Bit Symbol			KEY11	KEY10				KEY1EN
(0xFFFF_F362)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY1 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY1 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST2	Bit Symbol			KEY21	KEY20				KEY2EN
(0xFFFF_F361)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY2 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY2 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST3	Bit Symbol			KEY31	KEY30				KEY3EN
(0xFFFF_F360)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY3 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY3 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST4	Bit Symbol			KEY41	KEY40				KEY4EN
(0xFFFF_F367)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY4 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY4 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST5	Bit Symbol			KEY51	KEY50				KEY5EN
(0xFFFF_F366)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY5 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY5 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST6	Bit Symbol			KEY61	KEY60				KEY6EN
(0xFFFF_F365)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY6 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY6 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST7	Bit Symbol			KEY71	KEY70				KEY7EN
(0xFFFF_F364)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY7 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY7 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST8	Bit Symbol			KEY81	KEY80				KEY8EN
(0xFFFF_F36B)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY8 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY8 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST9	Bit Symbol			KEY91	KEY90				KEY9EN
(0xFFFF_F36A)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEY9 interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEY9 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPSTA	Bit Symbol			KEYA1	KEYA0				KEYAEN
(0xFFFF_F369)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEYA interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEYA interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPSTB	Bit Symbol			KEYB1	KEYB0				KEYBEN
(0xFFFF_F368)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEYB interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEYB interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPSTC	Bit Symbol			KEYC1	KEYC0				KEYCEN
(0xFFFF_F36F)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEYC interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEYC interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPSTD	Bit Symbol			KEYD1	KEYD0				KEYDEN
(0xFFFF_F36E)	Read/Write			R/W					R/W
	Reset Value			1	0				0
	Function			KEYD interrupt trigger 00: Low level 01: High level 10: Falling level 11: Rising level					KEYD interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPCLR	Bit Symbol					KEYCLR3	KEYCLR2	KEYCLR1	KEYCLR0
(0xFFFF_F373)	Read/Write					W			
	Reset Value								
	Function					"1010" write of 00 clears all key interrupt requests			

19. ROM Correction

This chapter describes the ROM correction function supported by the TMP1962.

19.1 Features

- Up to eight 8-word sequences of data can be replaced.
- When the address stored in an address register (ADDREGn) matches the program counter (PC) value or the address generated by the DMAC (the lower five bits of the address are "don't care"), the data at the specified address in the on-chip ROM is replaced with the data from the RAM area corresponding to the address register.
- Writing an address to an address register causes ROM correction for the address to be enabled automatically.
- A correction requiring the replacement of more than eight words can also be performed by replacing the ROM data with an instruction code which makes a branch to a specified location in the RAM area which contains substitution data.

19.2 Operation

To correct data in a ROM area (or a projected ROM area), store the physical start address of the area in an address register (ADDREG0-ADDREG7). Store the substitution data in the RAM area corresponding to the address register. Writing an address to an address register causes ROM correction for the address to be enabled automatically. Upon reset, the ROM correction function is disabled. If the initial routine executed upon reset is used to correct ROM data, write an address to the relevant address register after a reset is released. The address registers to which addresses are written are enabled for ROM correction. When the stored address matches the PC value (if the CPU has the bus right) or the source or destination address issued by the DMAC (if the DMAC has the bus right), the data at the specified address in the ROM is replaced with the data stored in the corresponding RAM area. For example, storing addresses in the ADDREG0 and ADDREG3 enables correction for the respective ROM areas, so that the ROM correction circuit block constantly monitors the PC and DMAC-issued addresses for a match with a specified address and, if a match is detected, replaces data, while ignoring the ADDREG2 and ADDREG4-ADDREG7. Each address register has bits 31:5 although only bits 19:5 are used for address comparison, in order to simplify the circuit. A match detected in the ROM correction circuit is internally ANDed with the ROMCS signal, which indicates a specified ROM address block, to determine an exact match. ROM addresses specified for correction must be located on eight-word boundaries, i.e., the lower five bits are 0. In other words, ROM data is always replaced in 32-byte units. If only part of 32 bytes need to be replaced, substitution RAM data corresponding to the other bytes must be the same as the current data in the corresponding ROM addresses.

The following table shows the relationship between the address registers and RAM areas.

Address Register	RAM Area
ADDREG0	0xFFFD_FF00 ~ 0xFFFD_FF1F
ADDREG1	0xFFFD_FF20 ~ 0xFFFD_FF3F
ADDREG2	0xFFFD_FF40 ~ 0xFFFD_FF5F
ADDREG3	0xFFFD_FF60 ~ 0xFFFD_FF7F
ADDREG4	0xFFFD_FF80 ~ 0xFFFD_FF9F
ADDREG5	0xFFFD_FFA0 ~ 0xFFFD_FFBF
ADDREG6	0xFFFD_FFC0 ~ 0xFFFD_FFDF
ADDREG7	0xFFFD_FFE0 ~ 0xFFFD_FFFF

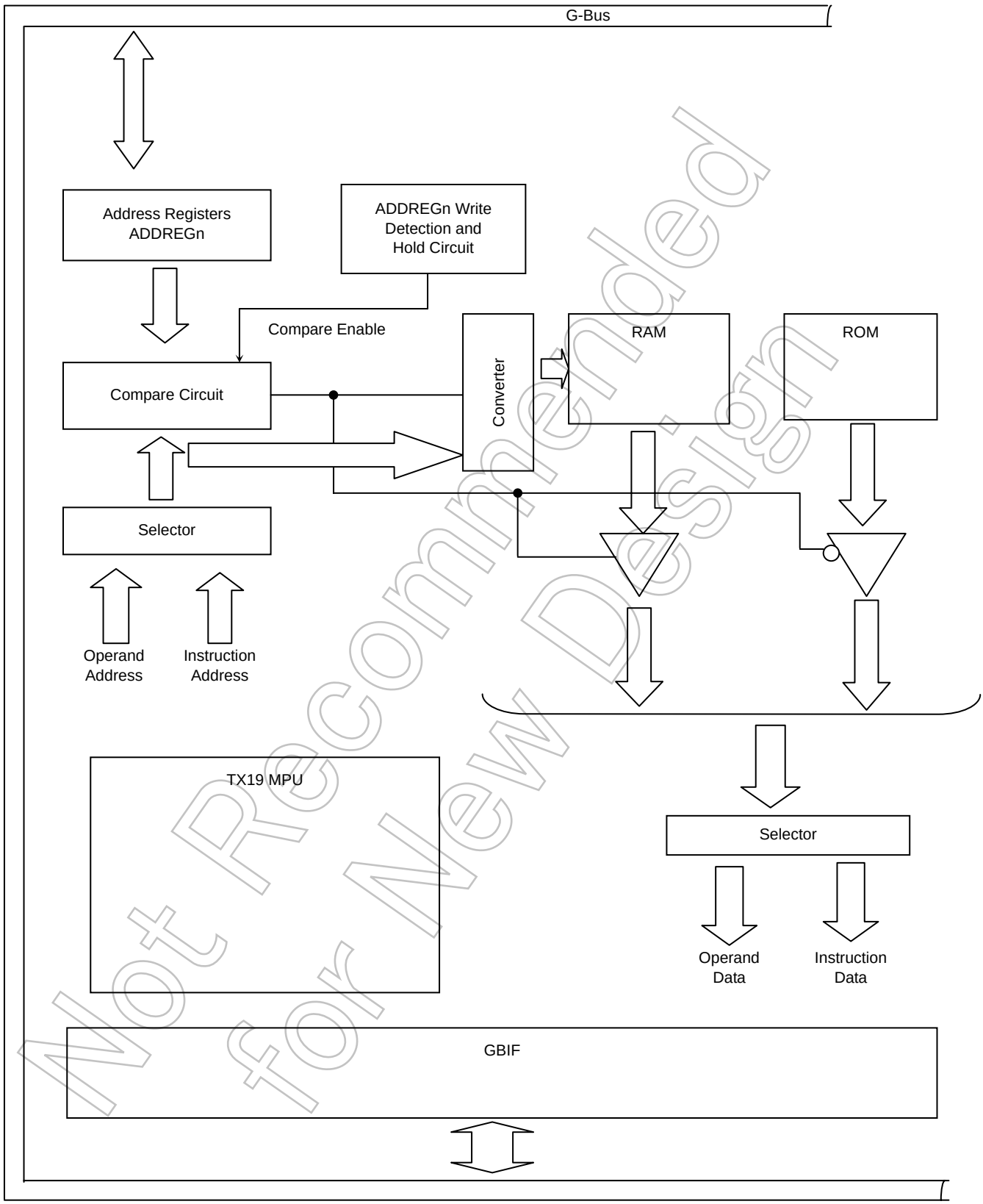


Figure 19.1 ROM Correction Block Diagram

19.3 Registers

(1) Address registers

ADDREG0 (0xFFFF_E540)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD07	ADD06	ADD05					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD015	ADD014	ADD013	ADD012	ADD011	ADD010	ADD09	ADD08
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD023	ADD022	ADD021	ADD020	ADD019	ADD018	ADD017	ADD016
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD031	ADD030	ADD029	ADD028	ADD027	ADD026	ADD025	ADD024
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG1 (0xFFFF_E544)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD17	ADD16	ADD15					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD115	ADD114	ADD113	ADD112	ADD111	ADD110	ADD19	ADD18
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD123	ADD122	ADD121	ADD120	ADD119	ADD118	ADD117	ADD116
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD131	ADD130	ADD129	ADD128	ADD127	ADD126	ADD125	ADD124
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG2 (0xFFFF_E548)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD27	ADD26	ADD25					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD215	ADD214	ADD213	ADD212	ADD211	ADD210	ADD29	ADD28
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD223	ADD222	ADD221	ADD220	ADD219	ADD218	ADD217	ADD216
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD231	ADD230	ADD229	ADD228	ADD227	ADD226	ADD225	ADD224
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG3 (0xFFFF_E54C)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD37	ADD36	ADD35					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD315	ADD314	ADD313	ADD312	ADD311	ADD310	ADD39	ADD38
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD323	ADD322	ADD321	ADD320	ADD319	ADD318	ADD317	ADD316
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD331	ADD330	ADD329	ADD328	ADD327	ADD326	ADD325	ADD324
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG4 (0xFFFF_E550)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD47	ADD46	ADD45					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD415	ADD414	ADD413	ADD412	ADD411	ADD410	ADD49	ADD48
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD423	ADD422	ADD421	ADD420	ADD419	ADD418	ADD417	ADD416
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD431	ADD430	ADD429	ADD428	ADD427	ADD426	ADD425	ADD424
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG5 (0xFFFF_E554)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD57	ADD56	ADD55					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD515	ADD514	ADD513	ADD512	ADD511	ADD510	ADD59	ADD58
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD523	ADD522	ADD521	ADD520	ADD519	ADD518	ADD517	ADD516
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD531	ADD530	ADD529	ADD528	ADD527	ADD526	ADD525	ADD524
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG6 (0xFFFF_E558)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD67	ADD66	ADD65					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD615	ADD614	ADD613	ADD612	ADD611	ADD610	ADD69	ADD68
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD623	ADD622	ADD621	ADD620	ADD619	ADD618	ADD617	ADD616
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD631	ADD630	ADD629	ADD628	ADD627	ADD626	ADD625	ADD624
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

ADDREG7 (0xFFFF_E55C)

	7	6	5	4	3	2	1	0
Bit Symbol	ADD77	ADD76	ADD75					
Read/Write	R/W							
Reset Value	0	0	0					
Function								
	15	14	13	12	11	10	9	8
Bit Symbol	ADD715	ADD714	ADD713	ADD712	ADD711	ADD710	ADD79	ADD78
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol	ADD723	ADD722	ADD721	ADD720	ADD719	ADD718	ADD717	ADD716
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol	ADD731	ADD730	ADD729	ADD728	ADD727	ADD726	ADD725	ADD724
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

Note 1: DMA transfer to an address register cannot be supported. DMA transfer to a substitution data area in the RAM is supported. The ROM correction function is supported when either the CPU or DMAC has an access right.

Note 2: Writing the initial value 0x00 replaces the reset address.

20. DSU Interface

The DSU interface is used for software debugging using an external DSU-probe unit. This serves as an interface to the DSU-probe, and cannot be used as a general-purpose port. Consult the DSU-probe operation manual for a description of debugging using a DSU-probe.

(1) Security feature

The TMP1962C10BXBG supports on-board debugging while it is installed on a printed circuit board. The TMP1962C10BXBG provides a security feature to prevent debugging. Unsecuring the device enables debugging with a DSU-probe.

(2) Securing the device (Disabling debugging with a DSU-probe)

The device is secured in the initial state. Debugging with a DSU-probe is disabled until the device is unsecured.

(3) Unsecuring the device (Enabling debugging with a DSU-probe)

The device may only be unsecured by clearing the SEQON bit in the SEQMOD register and then writing a special code (0x0000_00C5) to the Security Control (SEQCNT) register. This prevents runaway software from inadvertently turning off the security feature. However, a DSU-probe cannot read the contents of on-chip ROM or write to registers other than the processor core, on-chip memory and external device. When the device is reset, the device is secured until it is unsecured.

SEQMOD (0xFFFF_E510)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write								
Reset Value								
Function								
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write								
Reset Value								
Function								
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write								
Reset Value								
Function								
	7	6	5	4	3	2	1	0
Bit Symbol								SEQON
Read/Write								R/W
Reset Value								1
Function								1: Security on 0: Security off

Note: This register must be read as a 32-bit quantity. Bits 1 to 31 are read as 0s.

SEQCNT (0xFFFF_E514)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							

Note: This register must be read as a 32-bit quantity.

(4) Application example

The following flowchart exemplifies how to use the security feature with a DSU-probe.

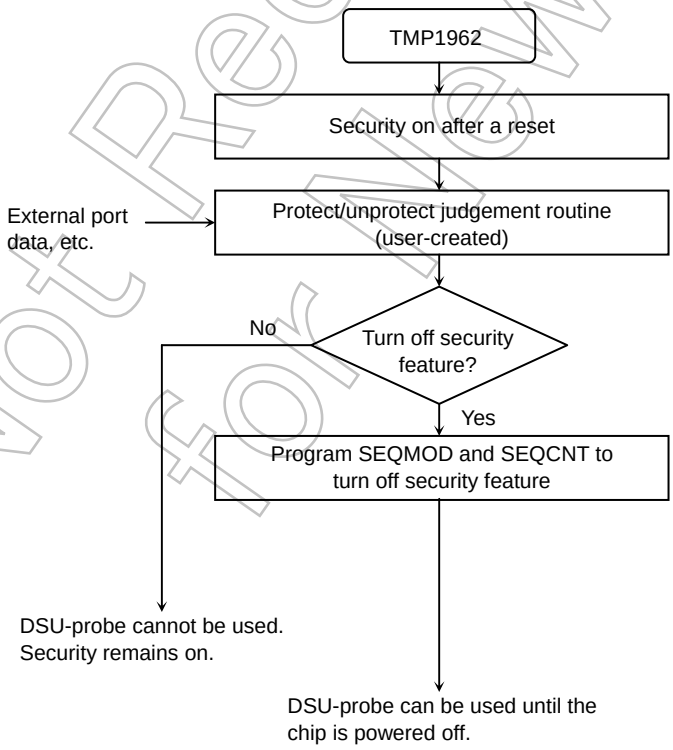


Figure 20.1 Using the Security Feature

21. JTAG Interface

The TMP1962 processor provides a boundary-scan interface that is compatible with Joint Test Action Group (JTAG) specifications, using the industry-standard JTAG protocol (IEEE Standard 1149.1/D6).

This chapter describes that interface, including descriptions of boundary scanning, the pins and signals used by the interface, and the Test Access Port (TAP).

21.1 What Boundary Scanning Is

With the evolution of ever-denser integrated circuits (ICs), surface-mounted devices, double-sided component mounting on printed-circuit boards (PCBs), and buried vias, in-circuit tests that depend upon making physical contact with internal board and chip connections have become more and more difficult to use. The greater complexity of ICs has also meant that tests to fully exercise these chips have become much larger and more difficult to write.

One solution to this difficulty has been the development of *boundary-scan* circuits. A boundary-scan circuit is a series of shift register cells placed between each pin and the internal circuitry of the IC to which the pin is connected, as shown in Figure 21.1. Normally, these boundary-scan cells are bypassed; when the IC enters test mode, however, the scan cells can be directed by the test program to pass data along the shift register path and perform various diagnostic tests. To accomplish this, the tests use the four signals described in the next section: **TDI**, **TDO**, **TMS**, **TCK, and** $\overline{\textbf{TRST}}$.

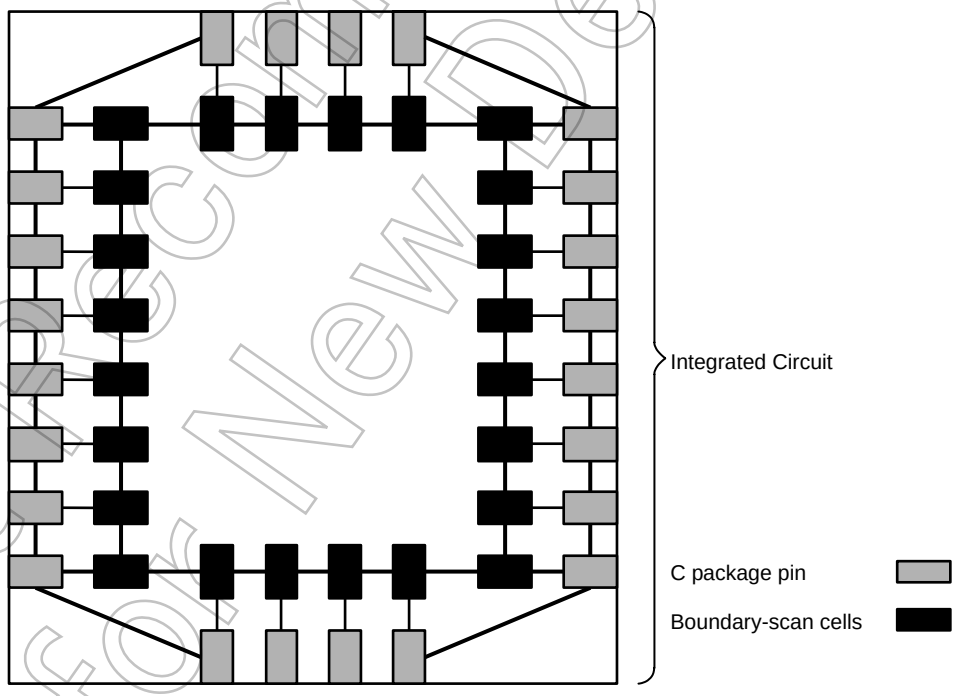


Figure 21.1 JTAG Boundary-scan Cells

21.2 Signal Summary

The JTAG interface signals are listed below and shown in Figure 21.2.

- TDI JTAG serial data in
- TDO JTAG serial data out
- TMS JTAG test mode select
- TCK JTAG serial clock input
- $\overline{\text{TRST}}$ JTAG test reset input

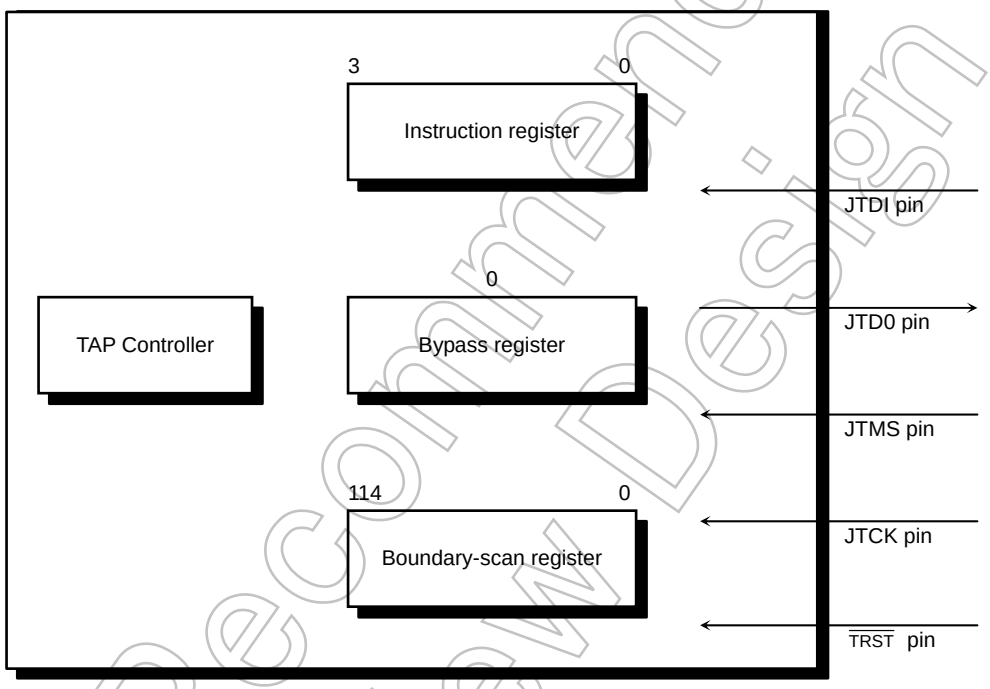


Figure 21.2 JTAG Interface Signals and Registers

The JTAG boundary-scan mechanism (referred to in this chapter as *JTAG mechanism*) allows testing of the connections between the processor, the printed circuit board to which it is attached, and the other components on the circuit board.

The JTAG mechanism does not provide any capability for testing the processor itself.

21.3 JTAG Controller and Registers

The processor contains the following JTAG controller and registers:

- *Instruction* register
- *Boundary-scan* register
- *Bypass* register
- *ID Code* register
- Test Access Port (TAP) controller

The processor executes the standard JTAG EXTEST operation associated with External Test functionality testing.

The basic operation of JTAG is for the TAP controller state machine to monitor the JTMS input signal. When it occurs, the TAP controller determines the test functionality to be implemented. This includes either loading the JTAG instruction register (IR), or beginning a serial data scan through a data register (DR), listed in Table 21.1. As the data is scanned in, the state of the JTMS pin signals each new data word, and indicates the end of the data stream. The data register to be selected is determined by the contents of the *Instruction* register.

21.3.1 Instruction Register

The JTAG *Instruction* register includes eight shift register-based cells; this register is used to select the test to be performed and/or the test data register to be accessed. As listed in Table 21.1, this encoding selects either the *Boundary-scan* register or the *Bypass* register or Device Identification register.

Table 21.1 JTAG Instruction Register Bit Encoding

Instruction Code (MSB → LSB)	Instruction	Selected Data Register
0000	EXTEST	Boundary Scan Register
0001	SAMPLE/PRELOAD	Boundary Scan Register
0010 to 1110	Reserved	Reserved
1111	BYPASS	Bypass register

Figure 21.3 shows the format of the *Instruction* register

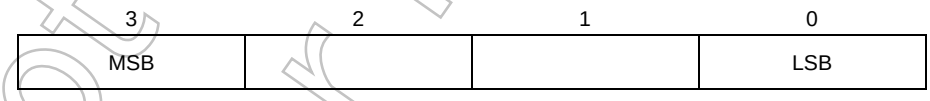


Figure 21.3 Instruction Register

The instruction code is shifted out to the Instruction register from the LSB.

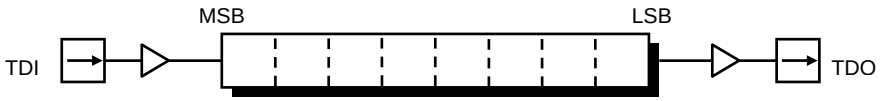


Figure 21.4 Instruction Register Shift Direction

21.3.2 Bypass Register

The *Bypass* register is 1 bit wide. When the TAP controller is in the Shift-DR (Bypass) state, the data on the TDI pin is shifted into the *Bypass* register, and the *Bypass* register output shifts to the TDO output pin.

In essence, the *Bypass* register is a short-circuit which allows bypassing of board-level devices, in the serial boundary-scan chain, which are not required for a specific test. The logical location of the *Bypass* register in the boundary-scan chain is shown in Figure 21.5. Use of the *Bypass* register speeds up access to boundary-scan registers in those ICs that remain active in the board-level test datapath.

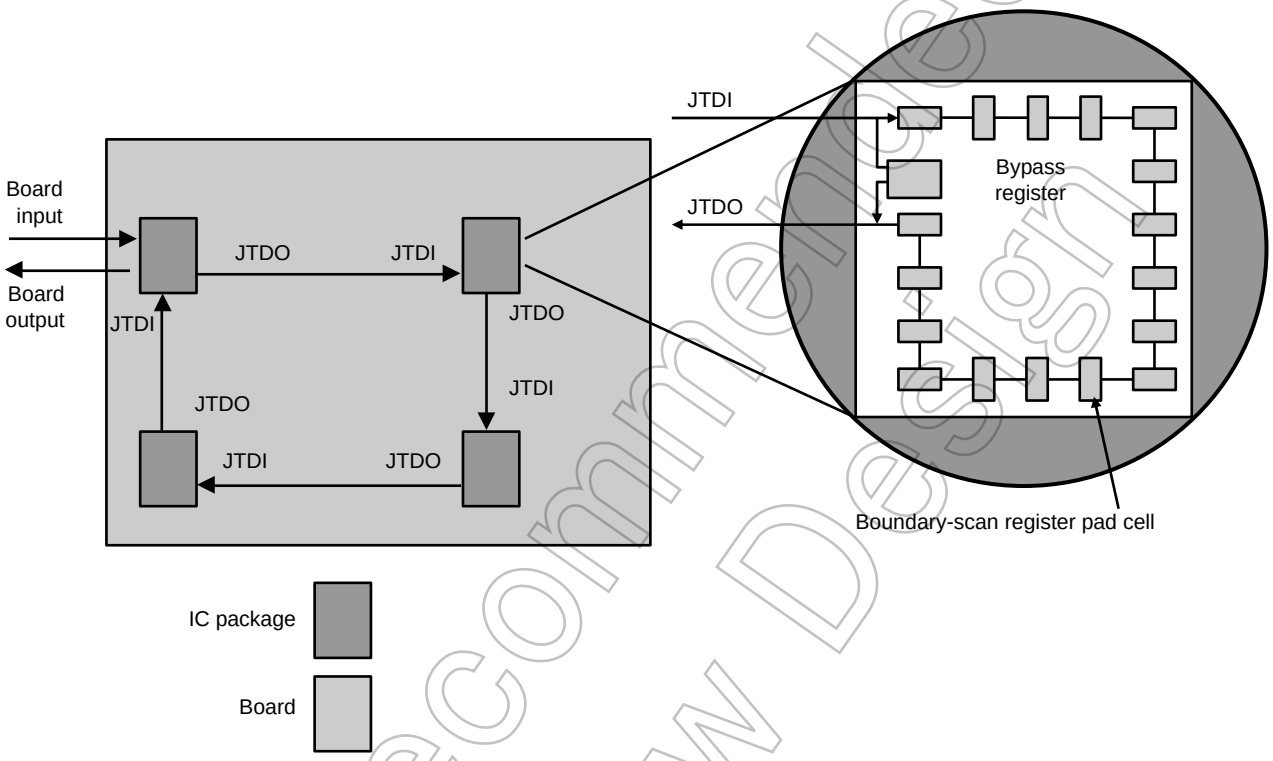


Figure 21.5 Bypass Register Operation

21.3.3 Boundary-Scan Register

The *Boundary Scan* register includes all of the inputs and outputs of the TMP1962 processor, except some analog output and control signals. The pins of the TMP1962 chip can be configured to drive any arbitrary pattern by scanning into the *Boundary Scan* register from the Shift-DR state. Incoming data to the processor is examined by shifting while in the Capture-DR state with the *Boundary Scan* register enabled.

The *Boundary-scan* register is a single, 115-bit-wide, shift register-based path containing cells connected to all input and output pads on the TMP1962 processor.

The TDI input is loaded to the LSB of the Boundary Scan register. The MSB of the Boundary Scan register is retrieved from the JTDO output.

21.3.4 Test Access Port (TAP)

The Test Access Port (TAP) consists of the five signal pins: $\overline{\textbf{TRST}}$, **TDI**, **TDO**, **TMS**, and **TCK**. Serial test data and instructions are communicated over these five signal pins, along with control of the test to be executed.

As Figure 21.6 shows, data is serially scanned into one of the three registers (*Instruction* register, *Bypass* register, or the *Boundary-scan* register) from the **TDI** pin, or it is scanned from one of these three registers onto the **TDO** pin.

The **TMS** input controls the state transitions of the main TAP controller state machine.

The **TCK** input is a dedicated test clock that allows serial JTAG data to be shifted synchronously, independent of any chip-specific or system clocks.

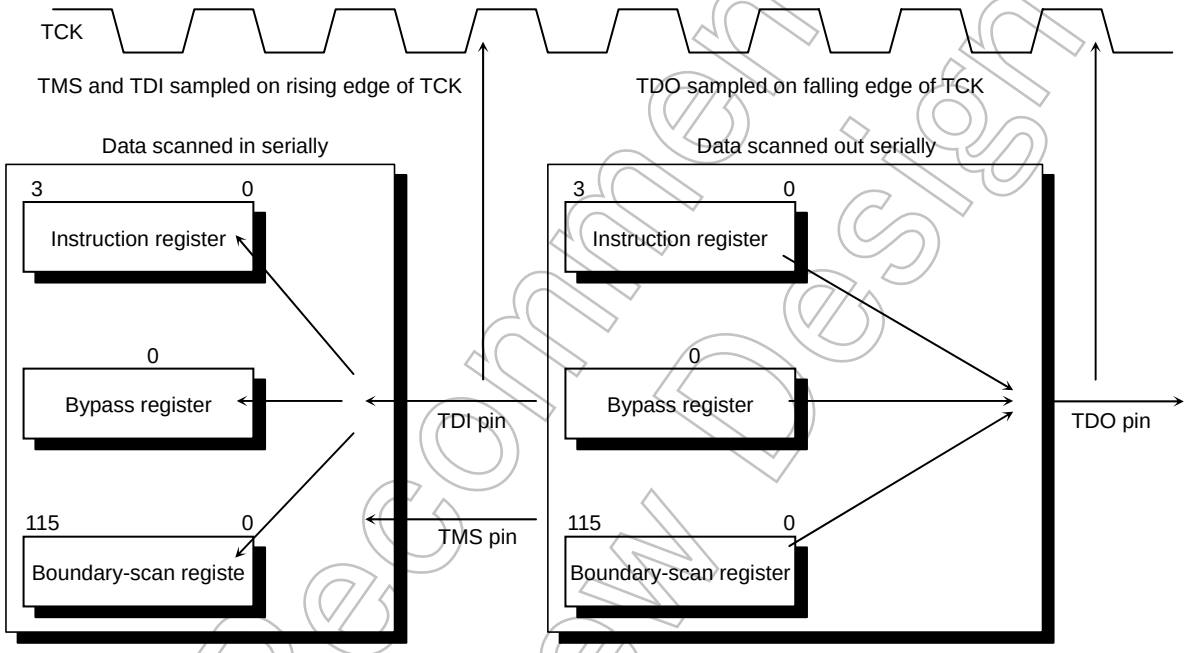


Figure 21.6 JTAG Test Access Port

Data on the **TDI** and **TMS** pins is sampled on the rising edge of the **TCK** input clock signal. Data on the **TDO** pin changes on the falling edge of the **TCK** clock signal.

21.3.5 TAP Controller

The processor implements the 16-state TAP controller as defined in the IEEE JTAC specification.

21.3.6 Controller Reset

The TAP controller state machine can be put into Reset state the following:

- assertion of the $\overline{\textbf{TRST}}$ signal (Low) resets the TAP controller.
- keeping the **TMS** input signal asserted through five consecutive rising edges of **TCK** input.

In either case, keeping **TMS** asserted maintains the Reset state.

21.3.7 TAP Controller

The state transition diagram of the TAP controller is shown in Figure 21.7. Each arrow between states is labeled with a 1 or 0, indicating the logic value of TMS that must be set up before the rising edge of TCK to cause the transition.

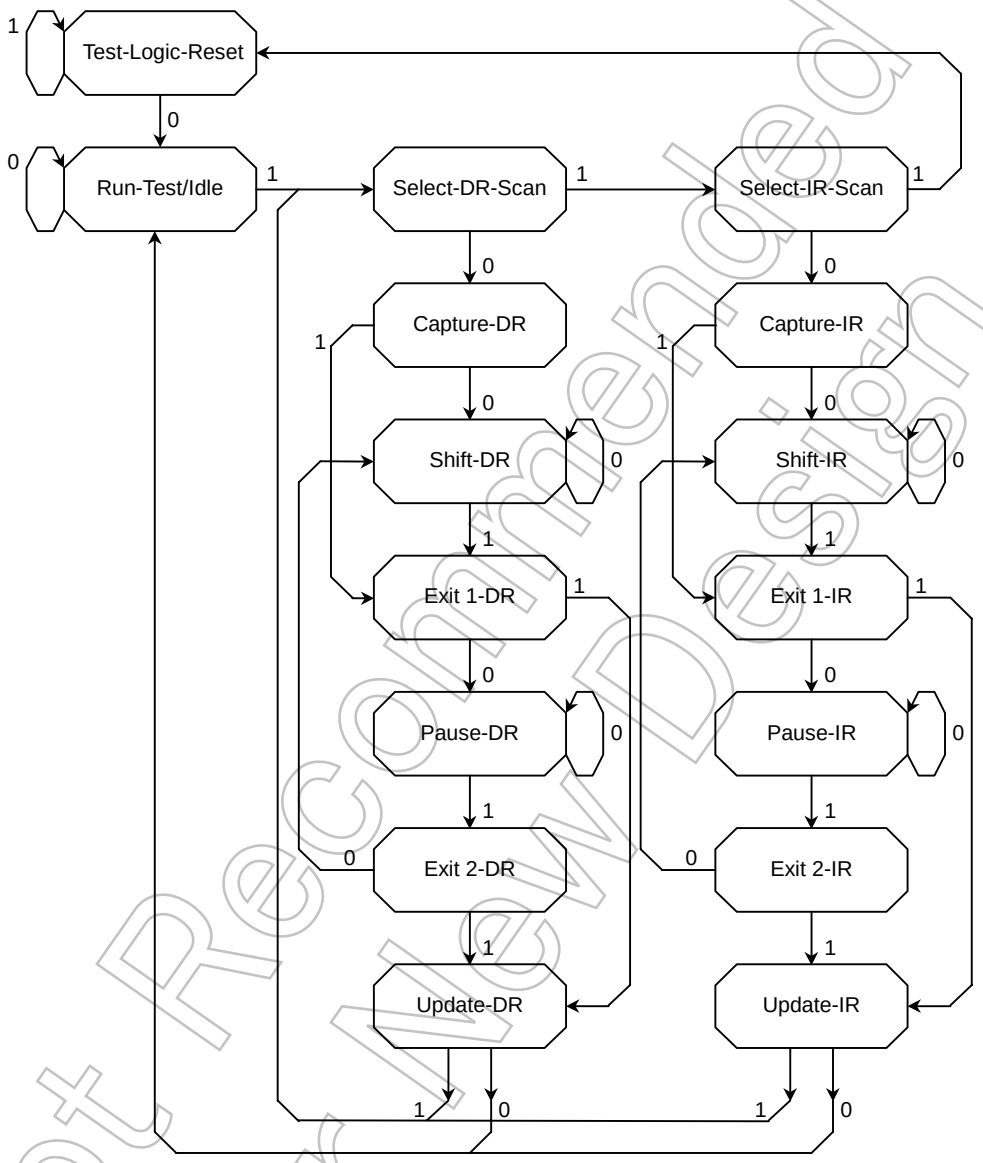


Figure 21.7 TAP Controller State Diagram

The following paragraphs describe each of the controller states. The left vertical column in Figure 21.7 is the data column, and the right vertical column is the instruction column. The data column and instruction column reference data register (DR) and instruction register (IR), respectively.

- Test-Logic-Reset
  When the TAP controller is in the Reset state, the Device Identification register is selected as default. The three most significant bits of the Boundary-scan register are cleared to 0, disabling the outputs.

  The controller remains in this state while TMS is high. If TMS is held low while the controller is in this state, then the controller moves to the Run-Test/Idle state.

- Run-Test/Idle
  In the Run-Test/Idle state, the IC is put in a test mode only when certain instructions such as a built-in self test (BIST) instruction are present. For instructions that do not cause any activities in this state, all test data registers selected by the current instruction retain their previous states.

  The controller remains in this state while TMS is held low. When TMS is high, the controller moves to the Select-DR-Scan state.

- Select-DR-Scan
  This is a temporary controller state. Here, the IC does not execute any specific functions.

  If TMS is held low when the controller is in this state, then the controller moves to the Capture-DR state. If TMS is held high, the controller moves to the Select-IR-Scan state in the instruction column.

- Select-IR-Scan
  This is a temporary controller state. Here, the IC does not execute any specific functions.

  If TMS is held low when the controller is in this state, then the controller moves to the Capture-IR state. If TMS is held high, the controller returns to the Test-Logic-Reset state.

- Capture-DR
  In this controller state, if the test data register selected by the current instruction on the rising edge of TCK has parallel inputs, then data can be parallel-loaded into the shift portion of the data register. If the test data register does not have parallel inputs, or if data need not be loaded into the selected data register, then the data register retains its previous state.

  If TMS is held low while the controller is in this state, the controller moves to the Shift-DR state. If TMS is held high, the controller moves to the Exit1-DR state.

- Shift-DR
  In this controller state, the test data register connected between TDI and TDO shifts data one stage forward towards its serial output.

  When the controller is in this state, then it remains in the Shift-DR state if TMS is held low, or moves to the Exit1-DR state if TMS is held high.

- Exit 1-DR
  This is a temporary controller state.

  If TMS is held low when the controller is in this state, the controller moves to the Pause-DR state. If TMS is held high, the controller moves to the Update-DR state.

- Pause-DR
  This state allows the shifting of the data register selected by the instruction register to be temporarily suspended. Both the instruction register and the data register retain their current states.

  When the controller is in this state, then it remains in the Pause-DR state if TMS is held low, or moves to the Exit2-DR state if TMS is held high.

- Exit 2-DR
  This is a temporary controller state.

  When the controller is in this state, then it returns to the Shift-DR state if TMS is held low, or moves on to the Update-DR state if TMS is held high.

- Update-DR
  In this state, data is latched, on the falling edge of TCK, onto the parallel outputs of the data registers from the shift register path. The data held at the parallel output does not change while data is shifted in the associated shift register path.

  When the controller is in this state, it moves to either the Run-Test/Idle state if TMS is held low, or the Select-DR-Scan state if TMS is held high.

- Capture-IR
  In this state, data is parallel-loaded into the instruction register. The two least significant bits are assigned the values “01”. The higher-order bits of the instruction register can receive any design specific values. The Capture-IR state is used for testing the instruction register. Faults in the instruction register, if any exist, may be detected by shifting out the data loaded in it.

  When the controller is in this state, it moves to either the Shift-IR state if TMS is low, or the Exit1-IR state if TMS is high.

- Shift-IR
  In this state, the instruction register is connected between TDI and TDO and shifts the captured data toward its serial output on the rising edge of TCK.

  When the controller is in this state, it remains in the Shift-IR state if TMS is low, or moves to the Exit1-IR state if TMS is high.

- Exit 1-IR
  This is a temporary controller state.

  When the controller is in this state, it moves to either the Pause-IR state if TMS is held low, or the Update-IR state if TMS is held high.

- Pause-IR
  This state allows the shifting of the instruction register to be temporarily suspended. Both the instruction register and the data register retain their current states.

  When the controller is in this state, it remains in the Pause-IR state if TMS is held low, or moves to the Exit2-IR state if TMS is held high.

- Exit 2-IR
  This is a temporary controller state.

  When the controller is in this state, it moves to either the Shift-IR state if TMS is held low, or the Update-IR state if TMS is held high.

- Update-IR
  This state allows the instruction previously shifted into the instruction register to be output in parallel on the rising edge of TCK. Then it becomes the current instruction, setting a new operational mode.

  When the controller is in this state, it moves to either the Run-Test/Idle state if TMS is low, or the Select-DR-Scan state if TMS is high.

Table 21.2 shows the boundary scan order of the processor signals.

Table 21.2 TMP1962 JTAG Boundary-Scan Ordering

[TDI]	1: PJ4	2: PJ2	3: PCST3	4: PSCT0	5: PJ3	6: SDI/$\overline{\text{DINT}}$
7: PCST2	8: SDAO/TPC	9: BW0	10: $\overline{\text{DBGE}}$	11: DCLK	12: PJ0	13: PCST1
14: SYSRDY	15: PJ1	16: $\overline{\text{DRESET}}$	17: TEST5	18: $\overline{\text{RESET}}$	19: $\overline{\text{PLLOFF}}$	20: $\overline{\text{NMI}}$
21: BW1	22: PN7	23: PN2	24: PN6	25: PN5	26: PN3	27: PN4
28: PN1	29: PO5	30: PN0	31: PO6	32: PO3	33: PO2	34: PO1
35: PP6	36: PO0	37: PP5	38: PP7	39: PO4	40: PP3	41: PP1
42: PP2	43: PO7	44: PP4	45: PP0	46: PB2	47: PB3	48: PB7
49: PB0	50: PB1	51: PB6	52: PB4	53: PA1	54: PA4	55: PF3
56: PA7	57: PA3	58: PA0	59: PF4	60: PA2	61: PB5	62: PG5
63: PA6	64: PA5	65: PF7	66: PF2	67: PF1	68: PF0	69: PG7
70: PF6	71: PG0	72: PF5	73: PD7	74: PG3	75: PG1	76: PG4
77: PG2	78: PD6	79: PD3	80: PD1	81: PD2	82: PG6	83: PD5
84: PD0	85: PD4	86: PC2	87: PC6	88: PC1	89: PC0	90: PC7
91: PH1	92: PC3	93: PH0	94: PC4	95: PH5	96: PH4	97: PH3
98: PH2	99: PC5	100: PE2	101: PE0	102: PE1	103: PH7	104: PH6
105: PE5	106: PE3	107: PE4	108: RSTPUP	109: PE6	110: PE7	111: P53
112: P60	113: P57	114: P50	115: P52	116: P64	117: P63	118: P56
119: P67	120: P62	121: P66	122: P55	123: P51	124: P26	125: P65
126: P20	127: P54	128: P61	129: P21	130: P22	131: P02	132: P23
133: P25	134: P24	135: P00	136: P27	137: P01	138: P06	139: P03
140: P05	141: P04	142: P11	143: P10	144: P07	145: P12	146: P15
147: P13	148: P14	149: P36	150: P17	151: P34	152: P40	153: P32
154: P33	155: P30	156: P16	157: P41	158: P35	159: P44	160: P37
161: P31	162: P43	163: P42	164: PK5	165: PI7	166: PI6	167: PK7
168: PI5	169: PI4	170: PI3	171: PM7	172: PK6	173: PI1	174: PI2
175: PI0	176: PK3	177: PK1	178: PK4	179: PK2	180: PL7	181: PM5
182: PK0	183: PM4	184: PM6	185: PM3	186: PM0	187: PM1	188: PM2
189: PL6	190: PL2	191: PL5	192: PL4	193: PL0	194: PL3	195: PL1
196: P75	197: P74	198: P77	199: P73	200: P70	201: P71	202: P72
203: P86	204: P84	205: P83	206: P85	207: P76	208: P82	209: P80
210: P81	211: P87	212: P97	213: P93	214: P94	215: P96	216: P90
217: P95	218: P91	219: P92	[TDO]:			

21.4 Instructions for JTAG

This section defines the instructions supplied and the operations that occur in response to those instructions.

21.4.1 The EXTEST Instruction

This instruction is used for external interconnect test, and targets the boundary scan register between TDI and TDO. The EXTEST instruction permits BSR cells at output pins to shift out test patterns in the Update-DR state and those at input pins to capture test results in the Capture-DR state.

Typically, before EXTEST is executed, the initialization pattern is first shifted into the boundary scan register using the SAMPLE/PRELOAD instruction. In the Update-DR state, the boundary scan register loaded with the initialization pattern causes known data to be driven immediately from the IC onto its external interconnects. This eliminates the possibility of bus conflicts damaging the IC outputs. The flow of data through the boundary scan register while the EXTEST instruction is selected is shown in Figure 21.8, which follows:

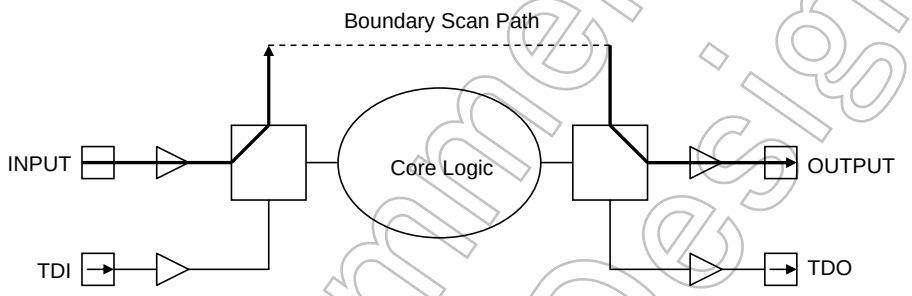


Figure 21.8 Test Data Flow While the EXTEST Instruction is Selected

The following steps describe the basic test algorithm of an external interconnect test.

1. Initialize the TAP controller to the Test-Logic-Reset state.
2. Load the instruction register with SAMPLE/PRELOAD. This causes the boundary scan register to be connected between TDI and TDO.
3. Initialize the boundary scan register by shifting in determinate data.
4. Then, load the initial test data into the boundary scan register.
5. Load the instruction register with EXTEST.
6. Capture the data applied to the input pin into the boundary scan register.
7. Shift out the captured data while simultaneously shifting in the next test pattern.
8. Read out the data in the boundary scan register onto the output pin.

Steps 6 to 8 are repeated for each test pattern.

21.4.2 The SAMPLE/PRELOAD Instruction

This instruction targets the boundary scan register between TDI and TDO. As the instruction's name implies, two functions are performed through use of the SAMPLE/ PRELOAD instruction.

- SAMPLE allows the input and output pads of an IC to be monitored. While it does so, it does not disconnect the system logic from the IC pins. The SAMPLE function occurs in the Capture-DR controller state. An example application of SAMPLE is to take a snapshot of the activity of the IC's I/O pins so as to verify the interaction between ICs during normal functional operation. The flow of data for the SAMPLE phase of the SAMPLE/PRELOAD instruction is shown in Figure 21.9.

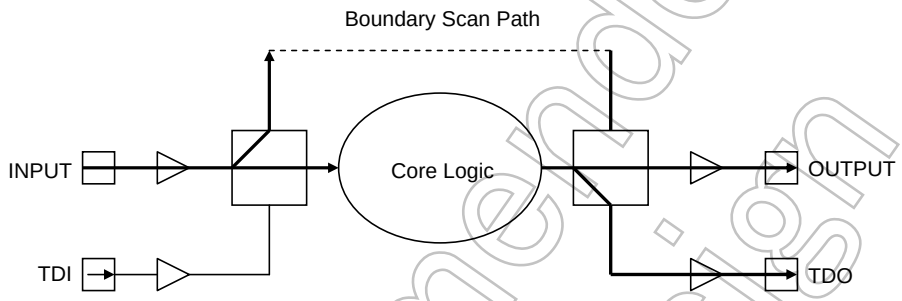


Figure 21.9 Test Data Flow While SAMPLE is Selected

- PRELOAD allows the boundary scan register to be initialized before another instruction is selected. For example, prior to selection of the EXTEST instruction, initialization data is shifted into the boundary scan register using PRELOAD as described in the previous subsection. PRELOAD permits shifting of the boundary scan register without interfering with the normal operation of the system logic. The flow of data for the PRELOAD phase of the SAMPLE/PRELOAD instruction is shown in Figure 21.10.

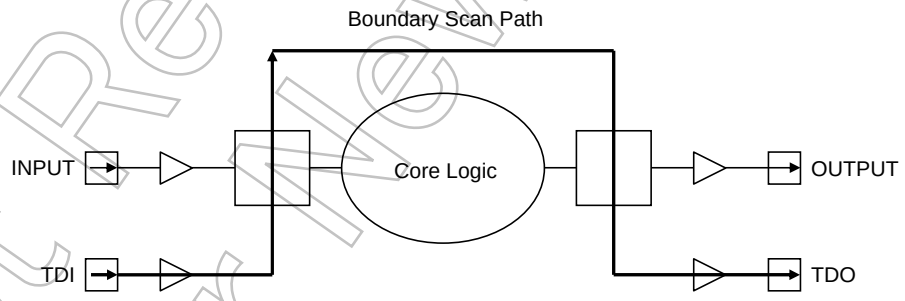


Figure 21.10 Test Data Flow While PRELOAD is Selected

21.4.3 The BYPASS Instruction

This instruction targets the bypass register between JTDI and JTDO. The bypass register provides a minimum length serial path through the IC (or between JTDI and JTDO) when the IC is not required for the current test. The BYPASS instruction does not cause interference to the normal operation of the on-chip system logic. The flow of data through the bypass register while the BYPASS instruction is selected is shown in Figure 21.11.

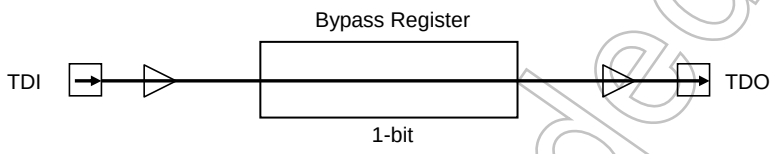


Figure 21.11 Test Data Flow While the Bypass Instruction is Selected

21.5 Note

This section describes details of JTAG boundary-scan operation that are specific to the processor.

- The **X2**, and **X1** signal pads do not support JTAG.
- When performing a JTAG operation, be sure to run the **MasterClock** before and after a reset operation to properly release the processor reset.
- Reset for JTAG

(1) JTAG circuit is initialized by $\overline{\textbf{TRST}}$ assertion. And then deassert $\overline{\textbf{TRST}}$.

(2) At input to **TMS** = 1 and asserted for more 5 TCK cycles.

22. Electrical Characteristics

The letter x in equations presented in this chapter represents the cycle period of the fsys clock selected through the programming of the SYSCR1.SYSCK bit. The fsys clock may be derived from either the high-speed or low-speed crystal oscillator. The programming of the clock gear function also affects the fsys frequency. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.SYSCK = 0) and a clock gear factor of 1/fc (SYSCR1.GEAR[1:0] = 00).

22.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC15} (Core)	−0.3 to 3.0	V
		V_{CC2} (I/O)	−0.3 to 4.0	
		V_{CC3} (I/O)	−0.3 to 4.0	
		AVCC (A/D)	−0.3 to 3.6	
Input voltage		V_{IN}	−0.3 to V_{CC} + 0.3	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	50	
Power dissipation (Ta = 85°C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−65 to 150	°C
Operating temperature		T_{OPR}	−20 to 85	°C

V_{CC15} = DVCC15 = CVCC15, $V_{CC}2$ = DVCC2, $V_{CC}3$ = DVCC3n (n = 1 to 4),
AVCC = AVCC3m (m = 1 to 2), V_{SS} = DVSS* = AVSS* = CVSS

Note: 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 Ratings value is exceeded with respect to current, voltage, power dissipation, 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.

22.2 DC Electrical Characteristics (1/4)

Ta = –20 to 85°C

Parameter		Symbol	Conditions	Min	Typ (Note 1)	Max	Unit
Supply voltage CVCC15 = DVCC15 CVSS = DVSS = 0 V		DVCC15	fosc = 10 to 13.5 MHz fsys = 3.75 to 40.5 MHz PLLON, INTLV = “H”	1.35		1.65	V
		DVCC2	fsys = 3 to 40.5 MHz	2.3		3.3	
		DVCC3n (n = 1 to 4)	fsys = 3 to 40.5 MHz	1.65		3.3	
Low-level input voltage	P7-P9 (Used as a port)	V_{IL1}	2.7 V ≤ AVCC32 ≤ AVCC31 ≤ 3.3 V	–0.3		0.3 AVCC31 0.3 AVCC32	V
			1.65 ≤ AVCC32 ≤ AVCC31 < 2.7 V				
	P0-P6, PA-PC, PD0-PD6, PE0-PE2, PF2-PF7, PG-PH, PI7, PJ1-PJ4, PL-PP	V_{IL2}	1.65 V ≤ DVCC3n ≤ 3.3 V (n = 1 to 4)			0.3 DVCC3n	
			2.3 V ≤ DVCC2 ≤ 3.3 V			0.3 DVCC2	
	PD7, PE3-PE7, PF0-PF1, PI0-PI6, PJ0, PK, $\overline{\text{PLLOFF}}$, RSTPUP, $\overline{\text{RESET}}$ $\overline{\text{DRESET}}$, $\overline{\text{DBGE}}$ SDI/ $\overline{\text{DINT}}$, TCK, TMS, TDI, $\overline{\text{TRST}}$ $\overline{\text{NMI}}$, BW0, BW1	V_{IL3}	2.7 V ≤ DVCC3n ≤ 3.3 V (n = 1 to 4) 2.7 V ≤ DVCC2 ≤ 3.3 V			0.15 DVCC3n	
			1.65 V ≤ DVCC3n < 2.7 V (n = 1 to 4) 2.3 V ≤ DVCC2 < 2.7 V			0.1 DVCC3n 0.1 DVCC2	
	X1	V_{IL4}	1.35 V ≤ CVCC15 ≤ 1.65 V			0.1 CVCC15	

Note 1: Ta = 25°C, DVCC3 = 3.0 V, DVCC2 = 2.5 V, AVCC3 = 3.3 V, unless otherwise noted.

22.3 DC Electrical Characteristics (2/4)

Ta = –20 to 85°C

Parameter		Symbol	Conditions		Min	Typ. (Note 1)	Max	Unit
High-level input voltage	P7-P9 (Used as a port)	V_{IH1}	2.7 V ≤ AVCC32 ≤ AVCC31 ≤ 3.3 V		0.7 AVCC31		DVCC3n + 0.3 DVCC2 + 0.3 CVCC15 + 0.2	V
			1.65 ≤ AVCC32 ≤ AVCC31 < 2.7 V		0.7 AVCC32			
	P0-P6, PA-PC, PD0-PD6, PE0-PE2, PF2-PF7, PG-PH, PI7, PJ1-PJ4, PL-PP	V_{IH2}	1.65 V ≤ DVCC3n ≤ 3.3 V (n = 1 to 4)		0.7 DVCC3n			
			2.3 V ≤ DVCC2 ≤ 3.3 V		0.7 DVCC2			
	PD7, PE3-PE7, PF0-PF1, PI0-PI6, PJ0, PK, $\overline{\text{PLLOFF}}$, RSTPUP, $\overline{\text{RESET}}$ $\overline{\text{DRESET}}$, $\overline{\text{DBGE}}$ SDI/$\overline{\text{DINT}}$, TCK, TMS, TDI, $\overline{\text{TRST}}$ $\overline{\text{NMI}}$, BW0, BW1	V_{IH3}	2.7 V ≤ DVCC3n ≤ 3.3 V (n = 1 to 4) 2.7 V ≤ DVCC2 ≤ 3.3 V		0.85 DVCC3n			
			1.65 V ≤ DVCC3n < 2.7 V (n = 1 to 4) 2.3 V ≤ DVCC2 < 2.7 V		0.9 DVCC3n 0.9 DVCC2			
	X1	V_{IH4}	1.35 V ≤ CVCC15 ≤ 1.65 V		0.9 CVCC15			
Low-level output voltage		V_{OL}	I_{OL} = 2 mA	DVCC3n ≥ 2.7 V DVCC2 ≥ 2.7 V			0.4	V
			I_{OL} = 500 μA	DVCC3n < 2.7 V DVCC2 < 2.7 V			0.2 DVCC3n ≤ 0.4 0.2 DVCC2 ≤ 0.4	
High-level output voltage		V_{OH}	I_{OH} = –2 mA	DVCC3n ≥ 2.7 V DVCC2 ≥ 2.7 V	2.4			
			I_{OH} = –500 μA	DVCC3n < 2.7 V DVCC2 < 2.7 V	0.8 DVCC3n 0.8 DVCC2			

Note 1: Ta = 25°C, DVCC3 = 3.0 V, DVCC2 = 2.5 V, AVCC3 = 3.3 V, unless otherwise noted.

22.4 DC Electrical Characteristics (3/4)

Ta = –20 to 85°C

Parameter	Symbol	Conditions	Min	Typ. (Note 1)	Max	Unit
Input leakage current	I_{LI}	$0.0 \le V_{IN} \le$ DVCC2 $0.0 \le V_{IN} \le$ DVCC3n (n = 1 to 4) $0.0 \le V_{IN} \le$ AVCC31 $0.0 \le V_{IN} \le$ AVCC32		0.02	±5	μA
Output leakage current	I_{LO}	$0.2 \le V_{IN} \le$ DVCC2 – 0.2 $0.2 \le V_{IN} \le$ DVCC3n – 0.2 (n = 1 to 4) $0.2 \le V_{IN} \le$ AVCC31 – 0.2 $0.2 \le V_{IN} \le$ AVCC32 – 0.2		0.05	±10	
Power-down voltage (STOP mode RAM backup)	V_{STOP} (DVCC15)		1.35		1.65	V
	V_{STOP1} (DVCC2)	V_{IL2} = 0.2DVCC2, V_{IL3} = 0.1DVCC2 V_{IH2} = 0.8DVCC2, V_{IH3} = 0.9DVCC2	2.3		3.3	
	V_{STOP2} (DVCC3) (AVCC3)	V_{IL} = 0.3DVCC33, V_{IL1} = 0.3AVCC31,32 V_{IL2} = 0.3DVCC3n, V_{IL3} = 0.1DVCC3n V_{IH} = 0.7DVCC33, V_{IH1} = 0.7AVCC31,32 V_{IH2} = 0.7DVCC3n, V_{IH3} = 0.9DVCC3n (n = 1 to 4)	1.65		3.3	
Pull-up resistor at Reset	RRST	DVCC2 = 2.5 V ±0.2 V	20	50	240	kΩ
Schmitt width PD7, PE3-PE7, PF0-PF1, PI0-PI6, PJ0, PK, $\overline{\text{PLLOFF}}$, RSTPUP, $\overline{\text{RESET}}$, $\overline{\text{DRESET}}$, $\overline{\text{DBGE}}$, SDI/$\overline{\text{DINT}}$, TCK, TMS, TDI, $\overline{\text{TRST}}$, $\overline{\text{NMI}}$, BW0, BW1	VTH	2.7 V ≤ DVCC3n ≤ 3.3 V (n = 2, 4) 2.7 V ≤ DVCC2 ≤ 3.3 V	0.4	0.9		V
		1.65 V ≤ DVCC3n < 2.7 V (n = 2, 4) 2.3 V ≤ DVCC2 < 2.7 V	0.3	0.6		
Programmable pull-up/ pull-down resistor P32-P37,P40-P43 KEY0-KEYD, DRESET, $\overline{\text{DBGE}}$, SDI/$\overline{\text{DINT}}$, TCK, TMS, TDI, $\overline{\text{TRST}}$	PKH	DVCC3n = 3.0 V ±0.3 V (n = 2 to 4)	15	50	100	kΩ
		DVCC3n = 2.5 V ±0.2 V (n = 2 to 4) DVCC2 = 2.5 V ±0.2 V	20	50	240	
		DVCC3n = 2.0 V ±0.2 V (n = 2 to 4) DVCC2 = 2.0 V ±0.2 V	25	160	600	
Pin capacitance (Except power supply pins)	C_{IO}	fc = 1 MHz			10	pF

Note 1: Ta = 25°C, DVCC3n = 3.0 V, DVCC2 = 2.5 V, AVCC3 = 3.0 V, unless otherwise noted.

22.5 DC Electrical Characteristics (4/4)

DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, DVCC3n = 3.0 V ±0.3 V, AVCC3m = 3.3 V ±0.2 V

Ta = 20 to 85°C (n = 1 to 4, m = 1, 2)

Parameter	Symbol	Conditions	Min	Typ. (Note 1)	Max	Unit
NORMAL (Note 2): Gear = 1/1	I_{CC}	f_{sys} = 40.5 MHz (f_{OSC} = 13.5 MHz, PLLON) INTLV = "H"		38	46	mA
IDLE (Doze)				16	28	
IDLE (Halt)				13	23	
STOP		DVCC15 = CVCC15 = 1.35 to 1.65 V DVCC2 = 2.3 to 2.7 V DVCC3n = 1.65 to 3.3 V AVCC3m = 3.7 to 3.3 V		50	900	μA

Note 1: Ta = 25°C, DVCC15 = 1.5 V, DVCC2 = 2.5 V, DVCC3n = 3.0 V, AVCC3m = 3.0 V, unless otherwise noted.

Note 2: Measured with the CPU drystone operating, all I/O peripherals channel on, and 16-bit external bus operated with 4 system clocks.

Note 3: The supply current flowing through the DVCC15, DVCC2, DVCC3n, CVCC15 and AVCC3m pins is included in the digital supply current parameter (ICC).

22.6 10-bit ADC Electrical Characteristics

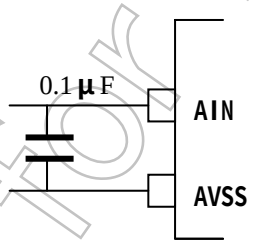
DVCC15= CVCC15= 1.5 ±0.15 V, DVCC2 = 2.5 ±0.2 V, DVCC3n = 3.0 ±0.3 V,
AVCC3m = 3.0 ±0.3 V, AVSS = DVSS, Ta = −20 to 85°C

Parameter		Symbol	Conditions	Min	Typ. (Note 1)	Max	Unit
Analog reference voltage (+)		VREFH		2.7 AVCCm − 0.3	 AVCC	3.3 AVCCm + 0.3	V
Analog reference voltage (−)		VREFL		AVSS	AVSS	AVSS + 0.2	V
Analog input voltage		VAIN		VREFL		VREFH	V
Analog supply current	A/D conversion	IREF	AVCCm = VREFH = 3.0 V ±0.3 V DVSS = AVSS = VREFL		0.35	1.0	mA
	Non-A/D conversion		AVCCm = VREFH = 3.0 V ±0.3 V DVSS = AVSS = VREFL		0.02	10	μA
Analog input capacitance		—			5.0		pF
Analog input impedance		—			5.0		kΩ
INL error		—	AVCCm = VREFH = 3.0 V ±0.3 V DVSS = AVSS = VREFL AIN resistance < 13.3 kΩ AIN load capacitance < 20 pF AVCCm load capacitance ≥ 10 μF VREFH load capacitance ≥ 10 μF Conversion time ≥ 7.9 μs Note*		±2	±3	LSB
DNL error		—			±1.5	±3	LSB
Offset error		—			±2	±3	LSB
Gain error		—			±2	±6	LSB

Note 1: 1LSB = (VREFH − VREFL)/1024[V]

Note 2: The supply current flowing through the AVCCm pin is included in the digital supply current parameter (ICC).

Note*:Connection of an external capacitor is recommended*



AC Electrical Characteristics

22.6.1 Multiplex Bus Mode

(1) DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, AVCC3m = 3.0 ±0.3 V, DVCC33 = 3.0 V ±0.3 V, Ta = –20 to 85°C

1. ALE width = 0.5 clock cycle, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A15 valid to ALE low	t_{AL}	0.5x – 4.3		8		ns
3	A0-A15 hold after ALE low	t_{LA}	0.5x – 1.8		10.5		ns
4	ALE pulse width high	t_{LL}	0.5x – 0.3		12		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	0.5x – 2.3		10		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 0.6		24		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	x – 5.1		19.5		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	x – 5.1		19.5		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (2 + W) – 35.8		38	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (2 + W) – 35.8		38	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 30.7		18.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.7		46.5		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x – 0.1		24.5		ns
16	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 3.2		46		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 4.2		45		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (3 + 0.5) – 21.6		64.5	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (3 + 0.5) – 21.6		64.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 4.1	x (1.5 + 3 + N –2) – 18.7	57.4	67.4	ns

Note: Nos. 1 to 18 indicate the values obtained with 1 programmed wait state. Nos. 19 and 20 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

2. ALE width = 1.5 clock cycles, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A15 valid to ALE low	t_{AL}	1.5x – 3.9		33		ns
3	A0-A15 hold after ALE low	t_{LA}	0.5x – 1.8		10.5		ns
4	ALE pulse width high	t_{LL}	1.5x – 0.4		36.5		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	0.5x – 2.4		10		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 0.6		24		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x – 5.2		44		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x – 5.2		44		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (3 + W) – 35.9		62.5	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (3 + W) – 35.9		62.5	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 30.7		18.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.7		46.5		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x		24.6		ns
16	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 3.2		46		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 4.2		45		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (4 + 0.5) – 21.7		89	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (4 + 0.5) – 21.7		89	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 4.1	x (1.5 + 3 + N – 2) – 18.7	57.4	67.4	ns

Note: Nos. 1 to 18 indicate the values obtained with 1 programmed wait state. Nos. 19 and 20 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

(2) DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 2.5 V ±0.2 V, Ta = −20 to 85°C

1. ALE width = 0.5 clock cycle, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A15 valid to ALE low	t_{AL}	0.5x – 2.3		10		ns
3	A0-A15 hold after ALE low	t_{LA}	0.5x – 1.8		10.5		ns
4	ALE pulse width high	t_{LL}	0.5x – 0.3		12		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	0.5x – 2.3		10		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 0.6		24		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	x – 5.1		19.5		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	x – 5.1		19.5		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (2 + W) – 36.8		37	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (2 + W) – 36.8		37	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 31.7		17.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.2		47		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x – 0.1		24.5		ns
16	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.7		46.5		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		45.5		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x –0.1		24.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (3 + 0.5) – 22.6		63.5	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (3 + 0.5) – 22.6		63.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 5.1	x (1.5 + 3 + N – 2) – 19.7	56.4	66.4	ns

Note: Nos. 1 to 18 indicate the values obtained with 1 programmed wait state. Nos. 19 and 20 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

2. ALE width = 1.5 clock cycles, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A15 valid to ALE low	t_{AL}	1.5x – 2.4		34.5		ns
3	A0-A15 hold after ALE low	t_{LA}	0.5x – 1.8		10.5		ns
4	ALE pulse width high	t_{LL}	1.5x – 0.4		36.5		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	0.5x – 2.4		10		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 0.6		24		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x – 5.2		44		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x – 5.2		44		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (3 + W) – 36.9		61.5	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (3 + W) – 36.9		61.5	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 31.7		17.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.2		47		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x – 0.1		24.5		ns
16	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.7		46.5		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		45.5		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x –0.1		24.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (4 + 0.5) – 22.6		88.1	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (4 + 0.5) – 22.6		88.1	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 5.1	x (1.5 + 3 + N – 2) – 19.7	56.4	66.4	ns

Note: Nos. 1 to 18 indicate the values obtained with 1 programmed wait state. Nos. 19 and 20 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

(3) DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 1.8 V ±0.15 V, Ta = −20 to 85°C

1. ALE width = 0.5 clock cycle, 2 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6	33333			ns
2	A0-A15 valid to ALE low	t_{AL}	0.5x – 2.3		10		ns
3	A0-A15 hold after ALE low	t_{LA}	0.5x – 1.8		10.5		ns
4	ALE pulse width high	t_{LL}	0.5x – 0.3		12		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	0.5x – 2.3		10		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 0.6		24		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	x – 5.1		19.5		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	x – 5.1		19.5		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (2 + W) – 42.4		56	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (2 + W) – 42.4		56	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 37.3		36.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.3		71.5		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x – 0.1		24.5		ns
16	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.8		71		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		70		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (3 + 0.5) – 28.1		58	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (3 + 0.5) – 28.1		58	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 6.1	x (1.5 + 3 + N – 2) – 24.7	55.4	61.4	ns

Note: Nos. 1 to 18 indicate the values obtained with 1 programmed wait state. Nos. 19 and 20 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

2. ALE width = 1.5 clock cycles, 2 programmed wait states

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A15 valid to ALE low	t_{AL}	1.5x – 2.4		34.5		ns
3	A0-A15 hold after ALE low	t_{LA}	0.5x – 1.8		10.5		ns
4	ALE pulse width high	t_{LL}	1.5x – 0.4		36.5		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	0.5x – 2.3		10		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 0.6		24		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x – 5.2		44		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x – 5.2		44		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (3 + W) – 42.5		80.5	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (3 + W) – 42.5		80.5	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 37.3		36.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.3		71.5		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x – 0.1		24.5		ns
16	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.8		71		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		70		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (4 + 0.5) – 28.1		82.6	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (4 + 0.5) – 28.1		82.6	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 1) – 6.1	x (1.5 + 3 + N – 1) – 24.7	80	86	ns

Note: Nos. 1 to 18 indicate the values obtained with 1 programmed wait state. Nos. 19 and 20 indicate the values obtained with 4 externally generated wait states with N = 1.

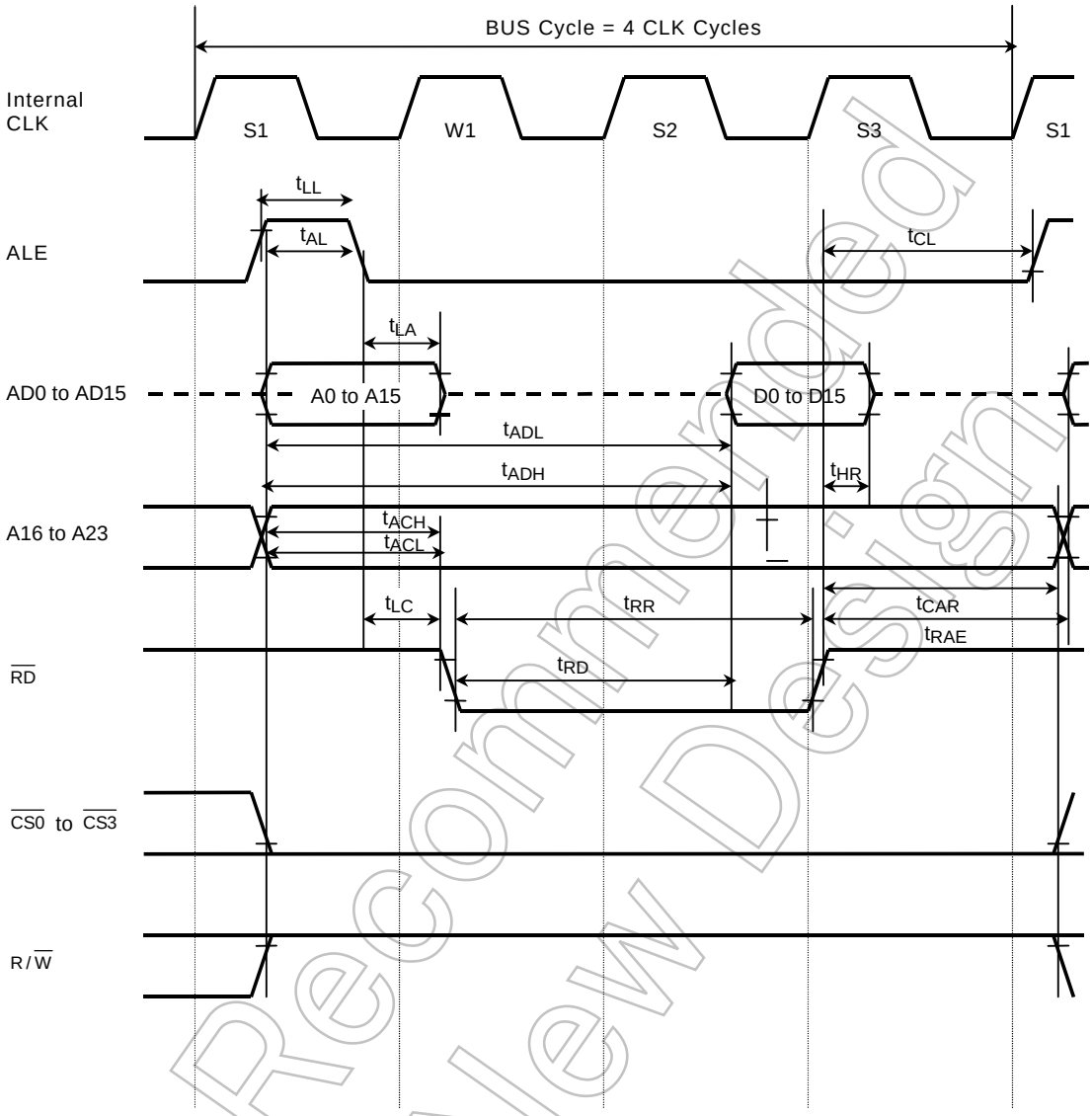
AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

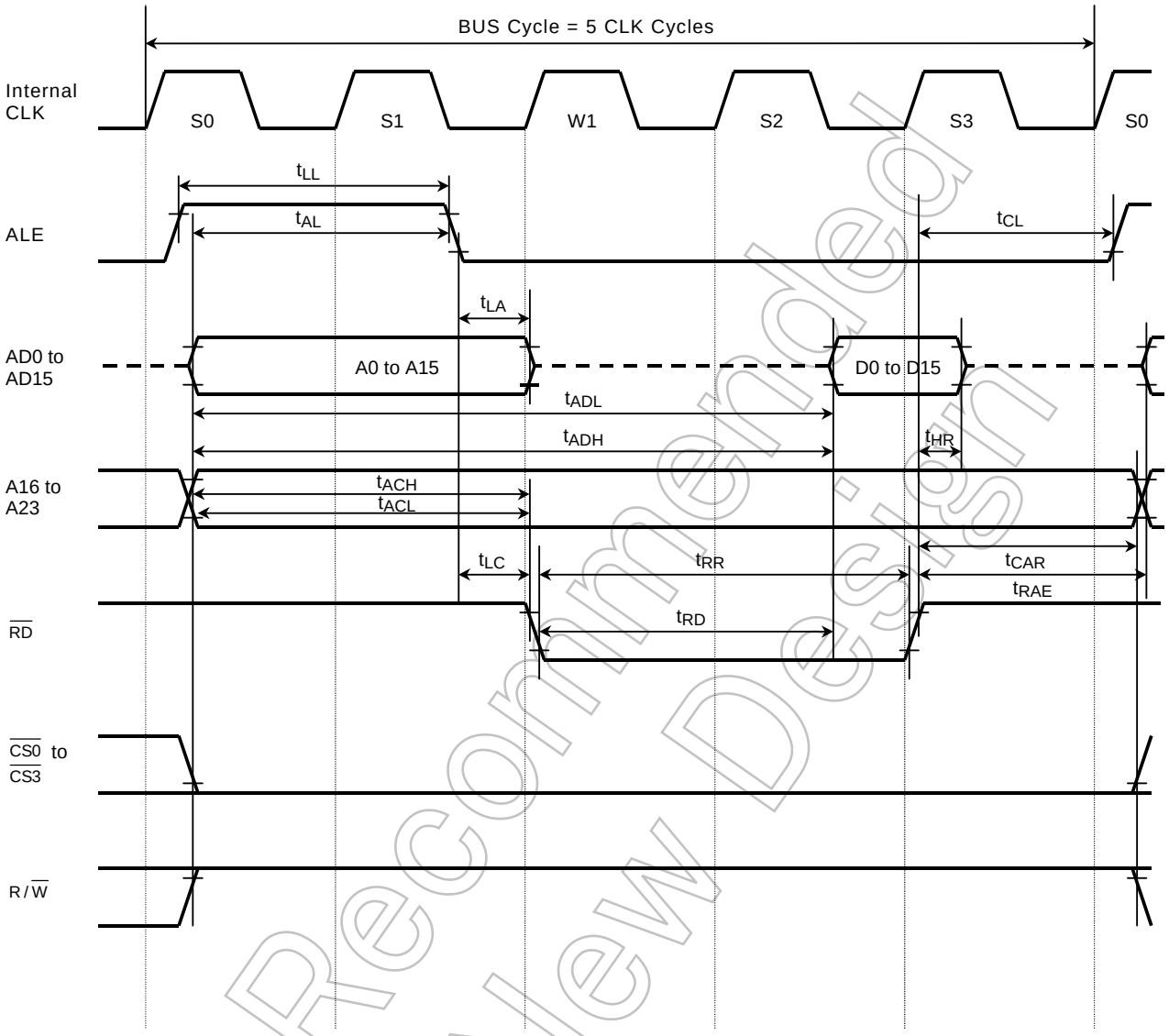
W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

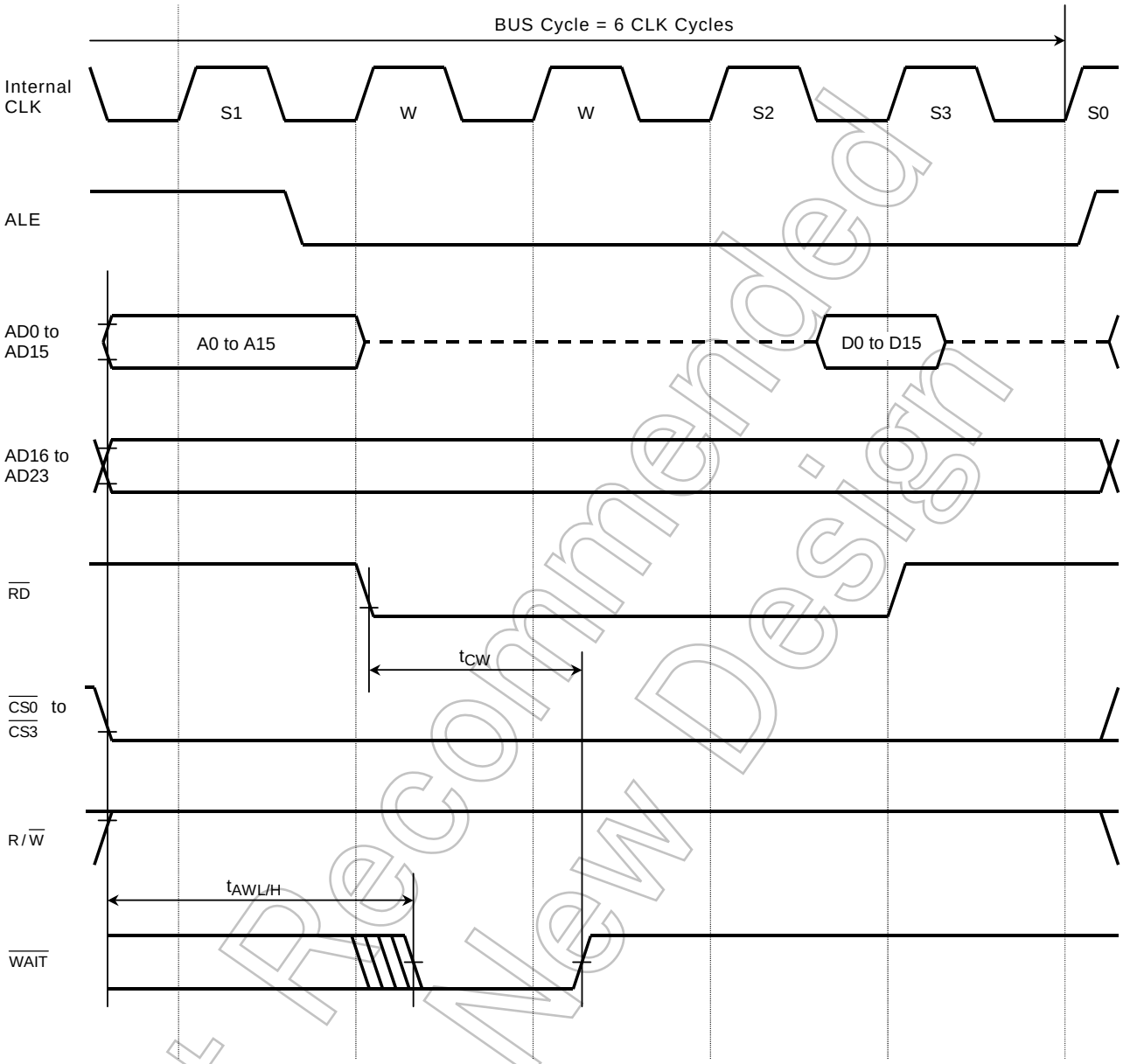
(1) Read cycle timing, ALE width = 0.5 clock cycle, 1 programmed wait state



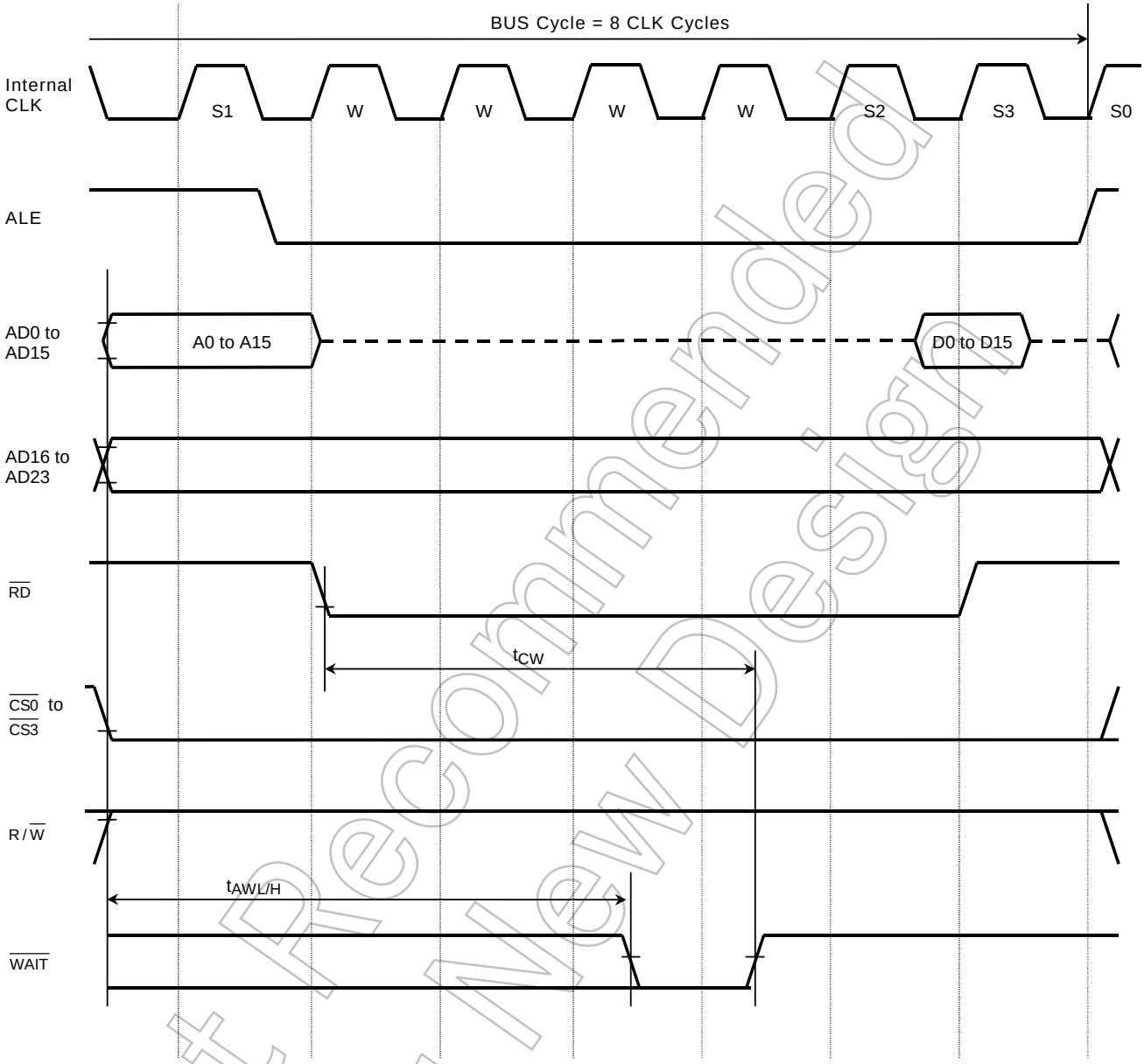
(2) Read cycle timing, ALE width = 1.5 clock cycles, 1 programmed wait state



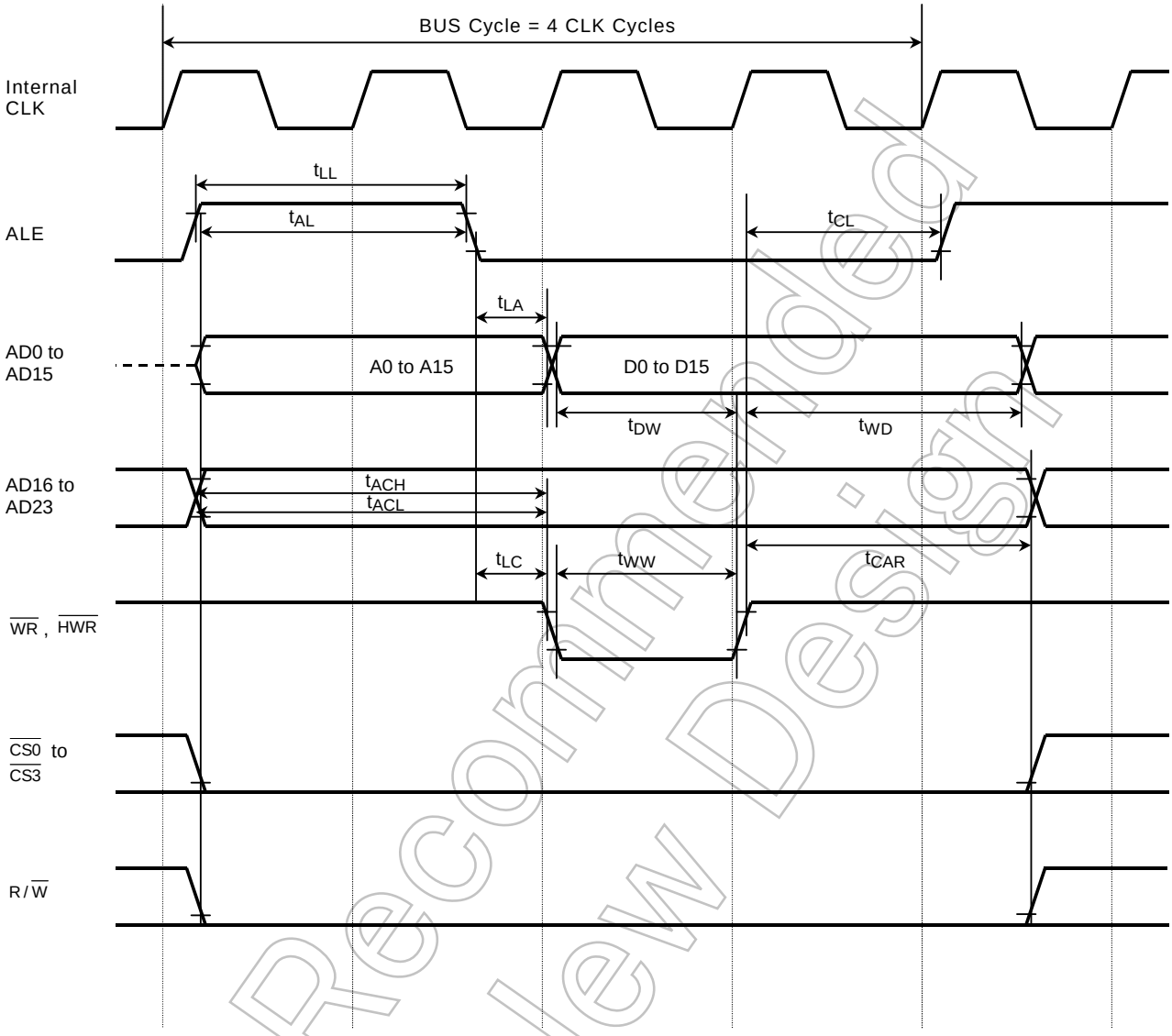
(3) Read cycle timing, ALE width = 1.5 clock cycles, 2 externally generated wait states with N = 1



(4) Read cycle timing, ALE width = 1.5 clock cycles, 4 externally generated wait states with N = 1



(5) Write cycle timing, ALE width = 1.5 clock cycles, zero wait state



22.6.2 Separate Bus Mode

(1) DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 3.0 V ±0.3 V, Ta = −20 to 85°C

1. SYSCR3<ALESEL> = 0, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	x – 5.1		19.5		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (2 + W) – 35.8		38	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 30.7		18.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.7		46.5		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x – 0.1		24.5		ns
9	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 3.2		46		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		—		1	ns
11	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 4.2		45		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x (3 + 0.5) – 21.6		64.5	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 4.1	x (1.5 + 3 + N – 2) – 18.7	57.4	67.4	ns

Note: Nos. 1 to 12 indicate the values obtained with 1 programmed wait state. Nos. 13 and 14 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

2. SYSCR3<ALESEL> = 1, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	2x – 5.2		44		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (3 + W) – 35.9		62.5	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 30.7		18.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.7		46.5		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x		24.6		ns
9	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 3.2		46		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		—		1	ns
11	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 4.2		45		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x (4 + 0.5) – 21.7		89	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 4.1	x (1.5 + 3 + N – 2) – 18.7	57.4	67.4	ns

Note: Nos. 1 to 12 indicate the values obtained with 1 programmed wait state. Nos. 13 and 14 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

(2) DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 2.5 V ±0.2 V, Ta = –20 to 85°C

1. SYSCR3<ALESEL> = 0, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	x – 5.1		19.5		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (2 + W) – 36.8		37	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 31.7		17.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.2		47		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x – 0.1		24.5		ns
9	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.7		46.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		—		1.5	ns
11	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		45.5		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x –0.1		24.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x (3 + 0.5) – 22.6		63.5	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 5.1	x (1.5 + 3 + N – 2) – 19.7	56.4	66.4	ns

Note: Nos. 1 to 12 indicate the values obtained with 1 programmed wait state. Nos. 13 and 14 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

2. SYSCR3<ALESEL> = 1, 1 programmed wait state

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	2x – 5.2		44		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (3 + W) – 36.9		61.5	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 31.7		17.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.2		47		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x – 0.1		24.5		ns
9	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.7		46.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		—		1.5	ns
11	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		45.5		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x –0.1		24.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x (4 + 0.5) – 22.6		88.1	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 5.1	x (1.5 + 3 + N – 2) – 19.7	56.4	66.4	ns

Note: Nos. 1 to 12 indicate the values obtained with 1 programmed wait state. Nos. 13 and 14 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

(3) DVCC15 = CVCC15 = 1.5 V ±0.15 V, DVCC2 = 2.5 V ±0.2 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 1.8 V ±0.15 V, Ta = −20 to 85°C

1. SYSCR3<ALESEL> = 0, 2 programmed wait states

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	x − 5.1		19.5		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x − 1.6		23		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (2 + W) − 42.4		56	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) − 37.3		36.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) − 2.3		71.5		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x − 0.1		24.5		ns
9	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) − 2.8		71		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		—		2	ns
11	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) − 3.8		70		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x − 0.1		24.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (3 + 0.5) − 28.1		58	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N − 2) − 6.1	x (1.5 + 3 + N −2) − 24.7	55.4	61.4	ns

Note: Nos. 1 to 12 indicate the values obtained with 2 programmed wait state. Nos. 13 and 14 indicate the values obtained with 4 externally generated wait states with N = 1.

AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

2. SYSCR3<ALESEL> = 1, 2 programmed states

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	24.6				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	2x – 5.2		44		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 1.6		23		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (3 + W) – 42.5		80.5	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) – 37.3		36.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 2.3		71.5		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x – 0.1		24.5		ns
9	$\overline{WR}$ or $\overline{HWR}$ width low	t_{WW}	x (1 + W) – 2.8		71		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		—		2	ns
11	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) – 3.8		70		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 0.1		24.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (4 + 0.5) – 28.1		82.6	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (0.5 + 3 + N – 2) – 6.1	x (1.5 + 3 + N – 2) – 24.7	55.4	61.4	ns

Note: Nos. 1 to 12 indicate the values obtained with 2 programmed wait state. Nos. 13 and 14 indicate the values obtained with 4 externally generated wait states with N = 1.

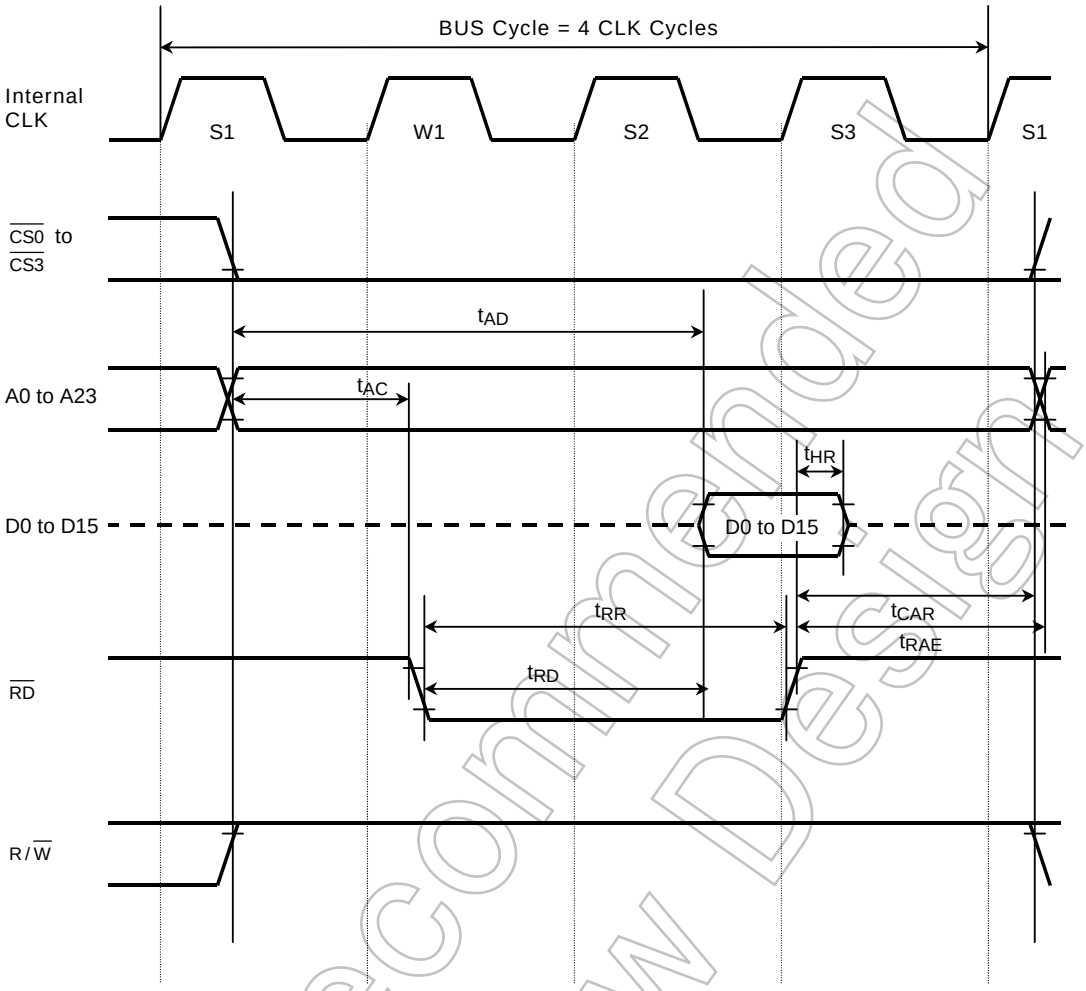
AC measurement conditions:

- Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
- Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

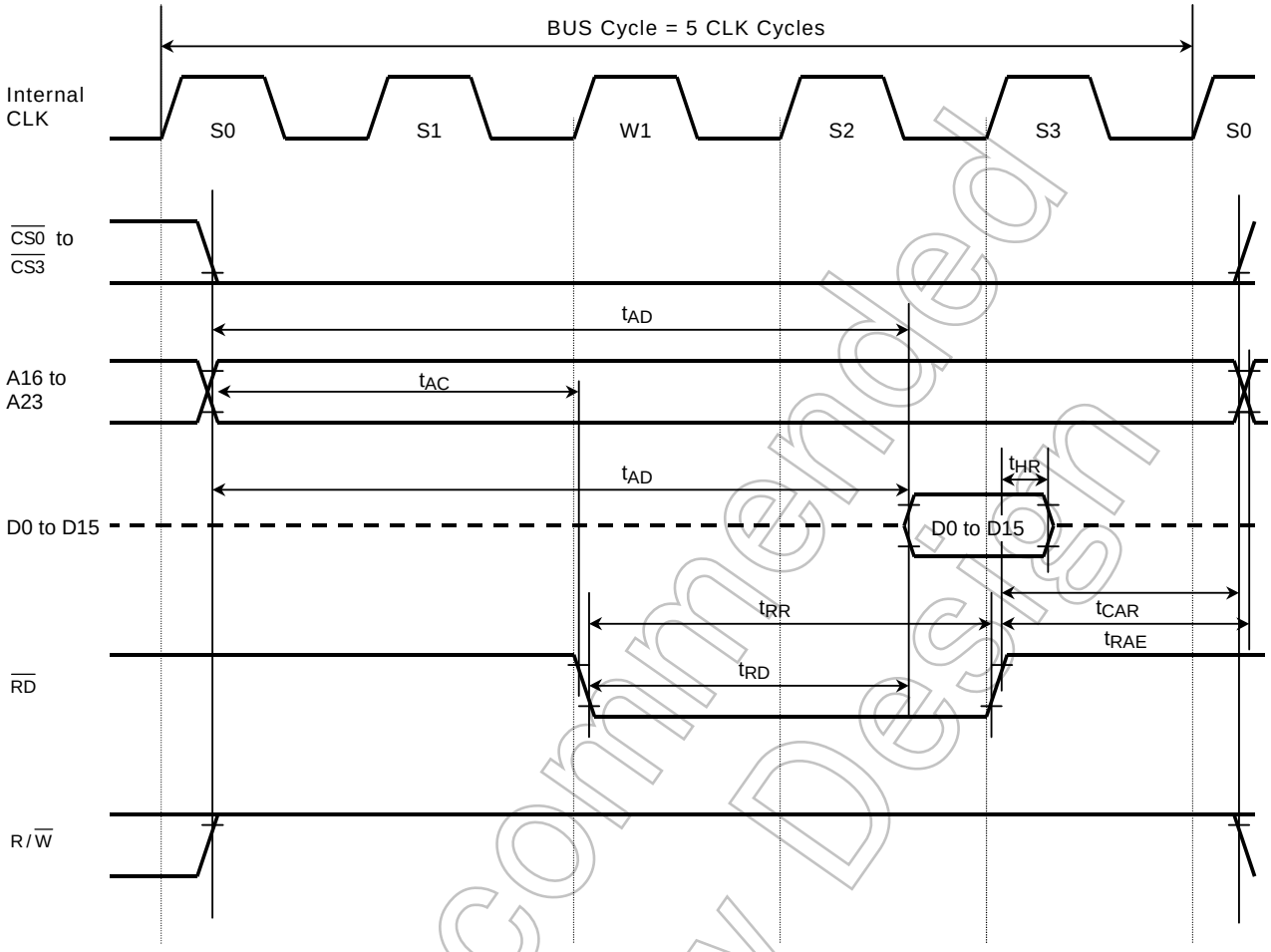
W: Number of wait state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (3 + N) wait insertion

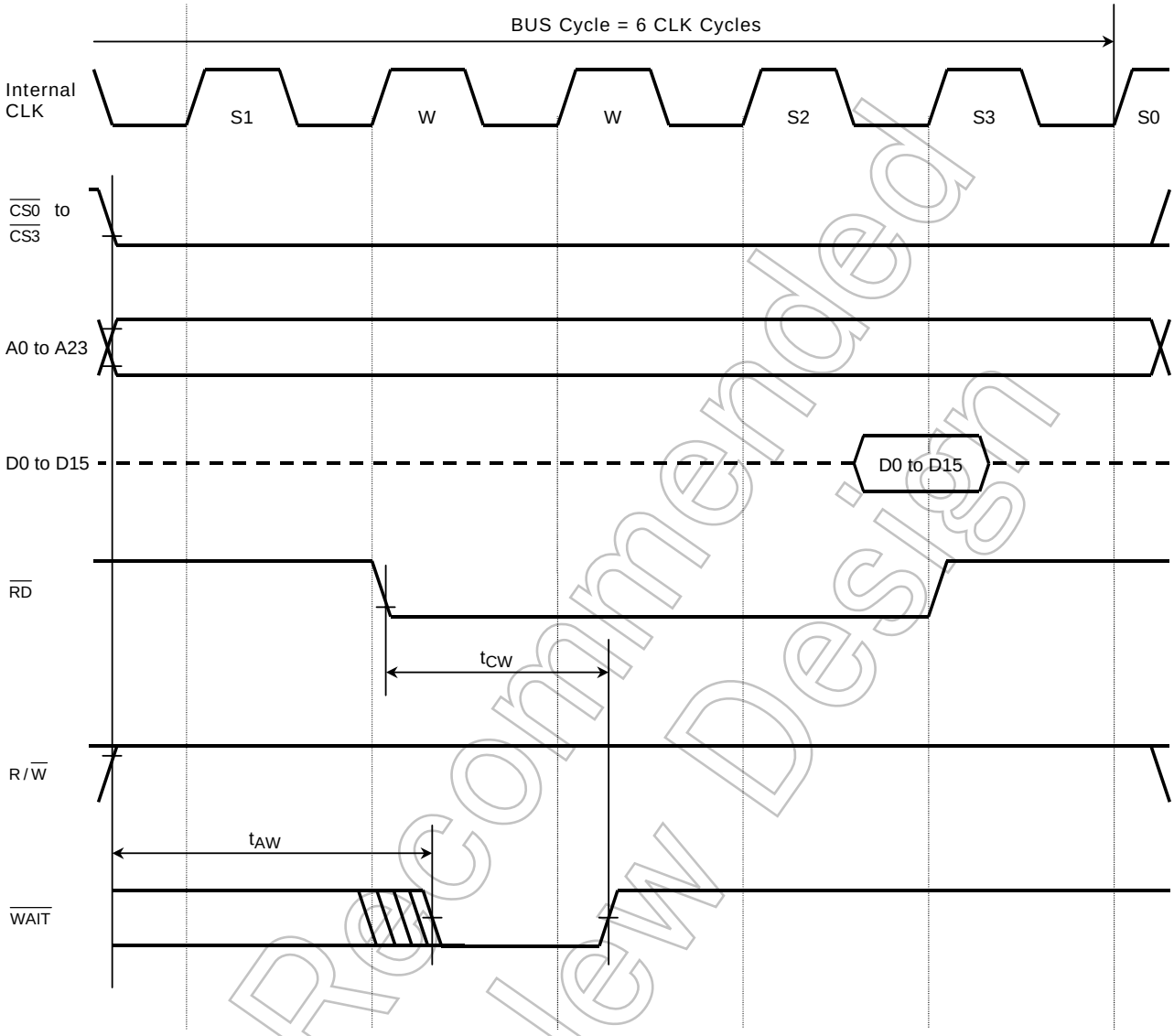
(1) Read cycle timing (SYSCR3<ALESEL> = 0, 1 programmed wait state)



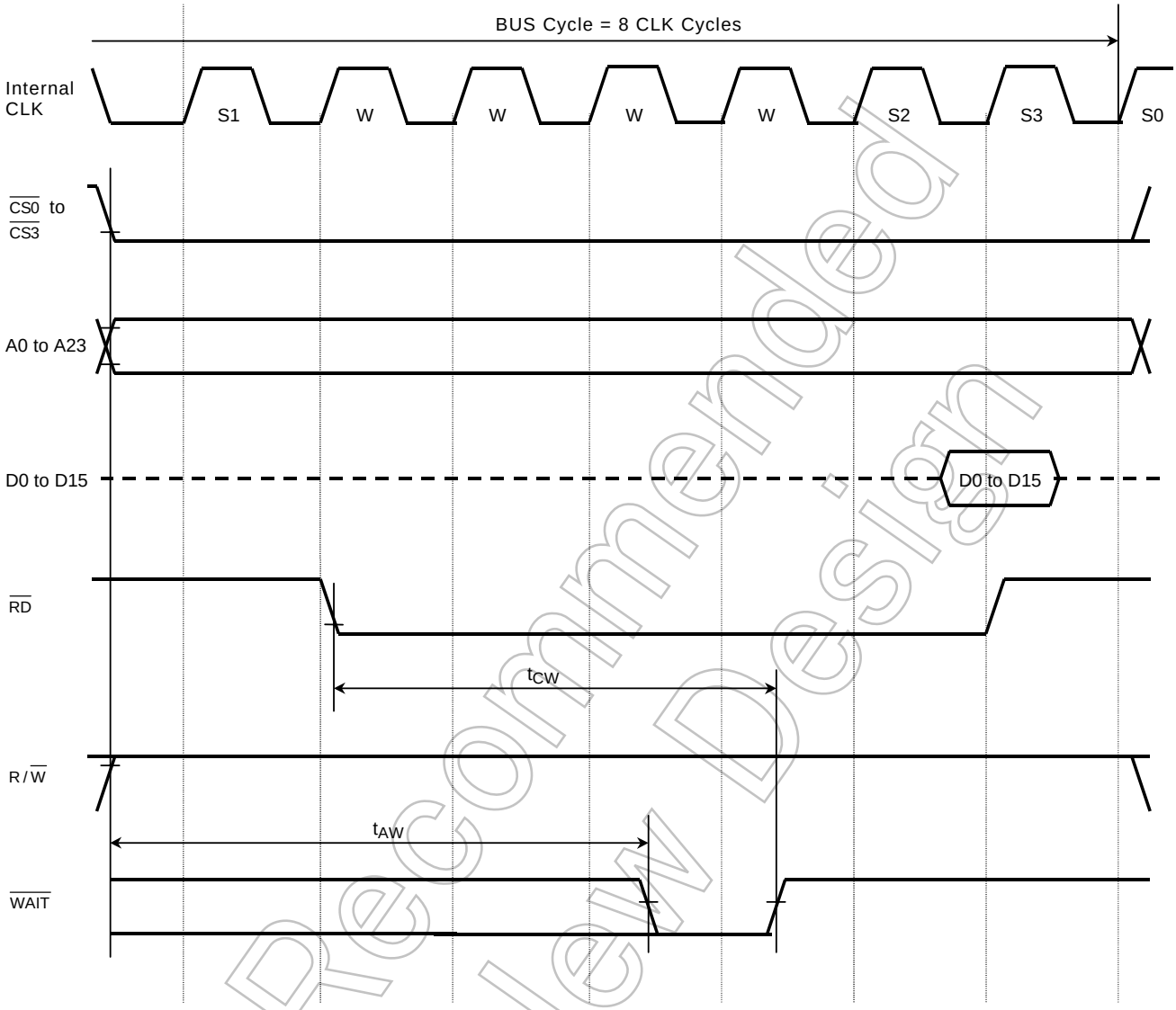
(2) Read cycle timing (SYSCR3<ALESEL> = 1, 1 programmed wait state)



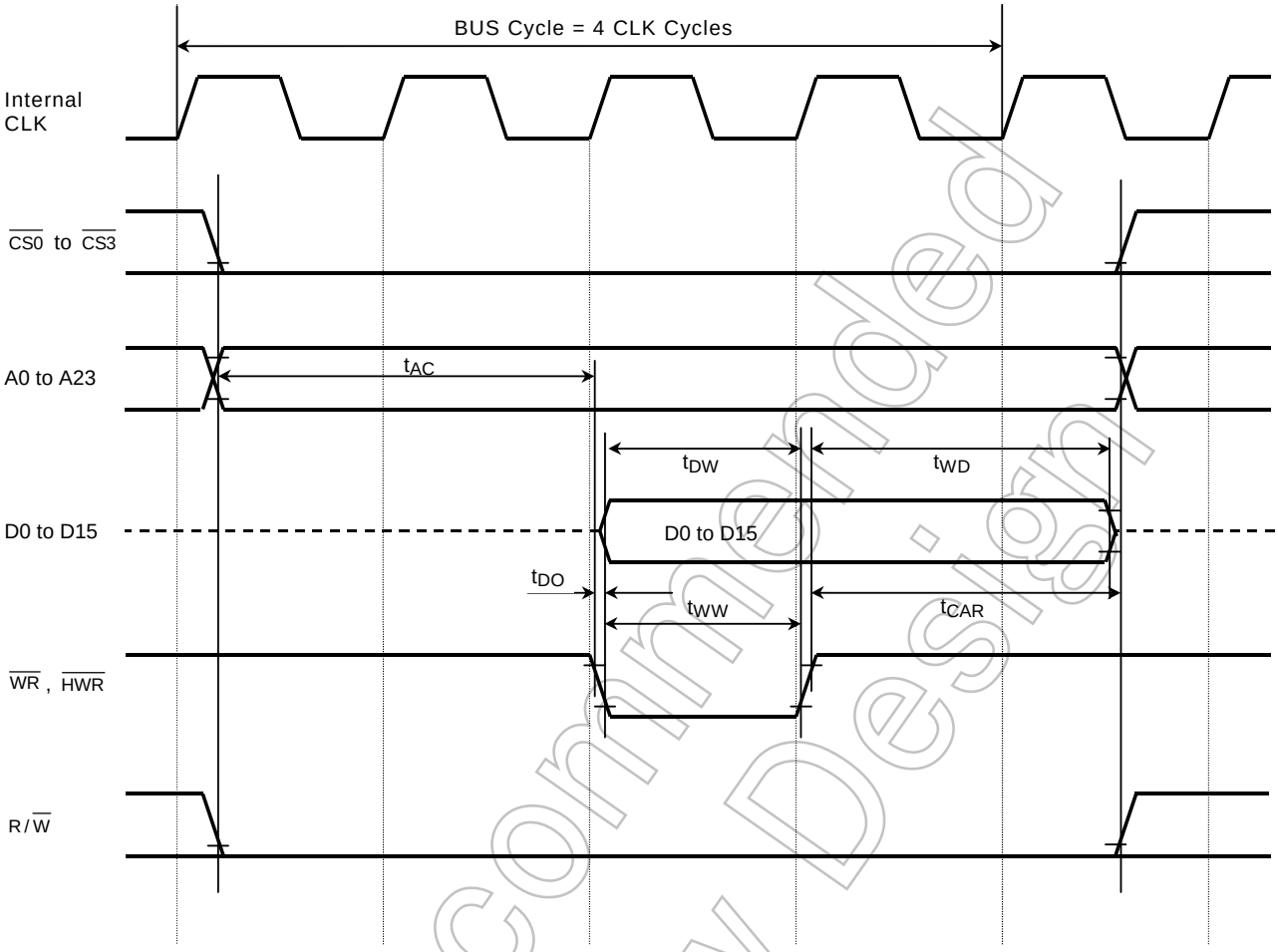
(3) Read cycle timing SYSCR3<ALESEL> = 1, 2 externally generated wait states with N = 1)



(4) Read cycle timing (SYSCR3<ALESEL> = 1, 4 externally generated wait states with N = 1)



(5) Write cycle timing (SYSCR3<ALESEL> = 1, zero wait sate)

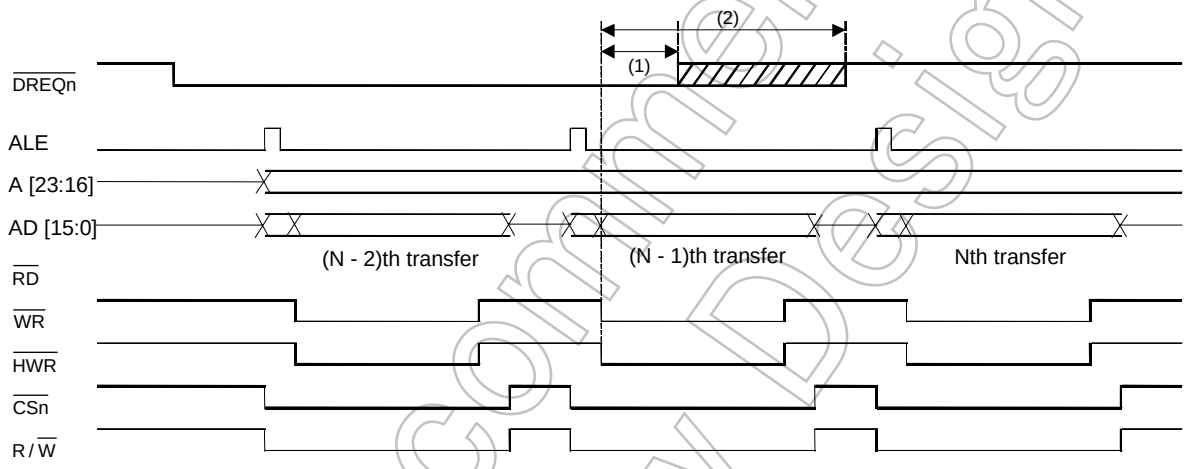


22.7 Transfer with DMA Request

The following shows an example of a transfer between the on-chip RAM and an external device in multiplex bus mode.

- 16-bit data bus width, non-recovery time
- Level data transfer mode
- Transfer size of 16 bits, device port size (DPS) of 16 bit
- Source/destination: on-chip RAM/external device

The following shows transfer operation timing of the on-chip RAM to an external bus during write operation (memory-to-memory transfer).



(1) Indicates the condition under which Nth transfer is performed successfully.

(2) Indicates the condition under which (N + 1)th transfer is not performed.

(1) DVCC2m = FVCC2 = CVCC2 = 2.5 V ±0.2 V, FVCC3 = 3.3 V ±0.3 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 2.3 V to 3.3 V, Ta = 20 to 85°C (m = 1 to 2)

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			(1) Min	(2) Max	Min	Max	
2	$\overline{RD}$ asserted to $\overline{DREQn}$ asserted (external device to on-chip RAM transfer)	tDREQ_r	Wx – 4.2	(2W + ALE + 6) x – 51	45	195	ns
3	$\overline{WR}$ / $\overline{HWR}$ asserted to $\overline{DREQn}$ asserted (on-chip RAM to external device transfer)	tDREQ_w	0	(2W + ALE + 4) x – 51.8	0	145	ns

(2) DVCC2m = FVCC2 = CVCC2 = 2.5 V ±0.2 V, FVCC3 = 3.3 V ±0.3 V, AVCC3m = 3.3 ±0.2 V, DVCC33 = 1.8 V ±0.15 V, Ta = 20 to 85°C (m = 1 to 2)

No.	Parameter	Symbol	Equation		40.5 MHz (fsys) (Note)		Unit
			(1) Min	(2) Max	Min	Max	
2	$\overline{RD}$ asserted to $\overline{DREQn}$ asserted (external device to on-chip RAM transfer)	tDREQ_r	Wx – 6.2	(2W + ALE + 6) x – 56	43	190	ns
3	$\overline{WR}$ / $\overline{HWR}$ asserted to $\overline{DREQn}$ asserted (on-chip RAM to external device transfer)	tDREQ_w	0	(2W + ALE + 4) x – 56.8	0	140	ns

W: Number of wait-state cycles inserted. In the case of (1 + N) externally generated wait states with N = 1, W becomes 2.

ALE: Apply ALE = 0 for ALE 0.5 clock, ALE = 1 for ALE 1.5 clock. The values in the above table are obtained with W = 2, ALE = 0.

22.8 Serial Channel Timing

(1) I/O Interface Mode (DVCC3n = 3.0 V ±0.3V)

In the table below, the letter x represents the fsys cycle period, which varies, depending on the programming of the clock gear function.

1. SCLK input mode (SIO0 to SIO6)

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	12x		296		ns
TxD data to SCLK rise of fall*	t_{OSS}	2x – 45		4		ns
TxD data hold after SCLK rise or fall*	t_{OHS}	8x – 15		182		ns
RxD data valid to SCLK rise or fall*	t_{SRD}	30		30		ns
RxD data hold after SCLK rise or fall*	t_{HSR}	2x – 30		19		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

2. SCLK output mode (SIO0 to SIO6)

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
SCLK period (programmable)	t_{SCY}	8x		197		ns
TxD data to SCLK rise	t_{OSS}	4x – 10		88		ns
TxD data hold after SCLK rise	t_{OHS}	4x – 10		88		ns
RxD data valid to SCLK rise	t_{SRD}	45		45		ns
RxD data hold after SCLK rise	t_{HSR}	0		0		ns

(2) I/O Interface Mode (DVCC3n = 2.5 V ±0.2 V)

In the table below, the letter x represents the fsys cycle period, which varies, depending on the programming of the clock gear function.

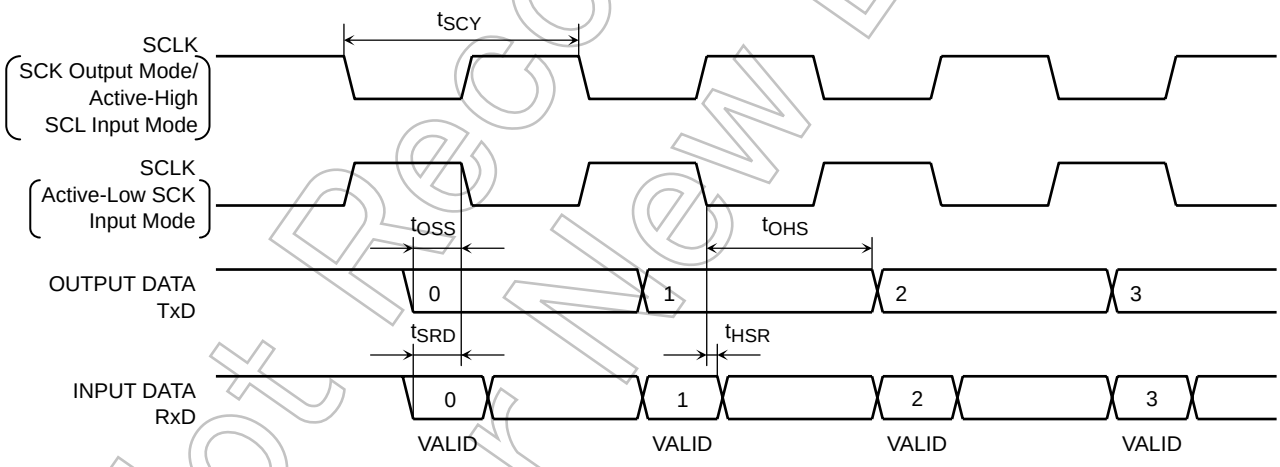
1. SCLK input mode (SIO0 to SIO6)

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	16x		395		ns
TxD data to SCLK rise of fall*	t_{OSS}	4x – 60		38		ns
TxD data hold after SCLK rise or fall*	t_{OHS}	10x – 15		232		ns
RxD data valid to SCLK rise or fall*	t_{SRD}	30		30		ns
RxD data hold after SCLK rise or fall*	t_{HSR}	2x + 10		59		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

2. SCLK output mode (SIO0 to SIO6)

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
SCLK period (programmable)	t_{SCY}	8x		197		ns
TxD data to SCLK rise	t_{OSS}	4x – 10		88		ns
TxD data hold after SCLK rise	t_{OHS}	4x – 10		88		ns
RxD data valid to SCLK rise	t_{SRD}	60		60		ns
RxD data hold after SCLK rise	t_{HSR}	0		0		ns



22.9 SBI Timing

(1) I2C Mode

In the table below, the letters x and T represent the fsys and ϕT0 cycle periods, respectively. The letter n denotes the value of n programmed into the SCK[2:0] (SCL output frequency select) field in the SBI0CR1.

Parameter	Symbol	Equation		Standard Mode f_{SYS} = 8 MHz n = 4		Fast Mode f_{SYS} = 32 MHz n = 4		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	t_{SCL}	0		0	100	0	400	kHz
Hold time for START condition	$t_{HD:STA}$			4.0		0.6		μs
SCL clock low width (Input) (Note 1)	t_{LOW}			4.7		1.3		μs
SCL clock high width (Output) (Note 2)	t_{HIGH}			4.0		0.6		μs
Setup time for a repeated START condition	$t_{SU;STA}$	(Note 5)		4.7		0.6		μs
Data hold time (Input) (Note 3, 4)	$t_{HD;DAT}$			0.0		0.0		μs
Data setup time	$t_{SU;DAT}$			250		100		ns
Setup time for STOP condition	$t_{SU;STO}$			4.0		0.6		μs
Bus free time between STOP and START conditions	t_{BUF}	(Note 5)		4.7		1.3		μs

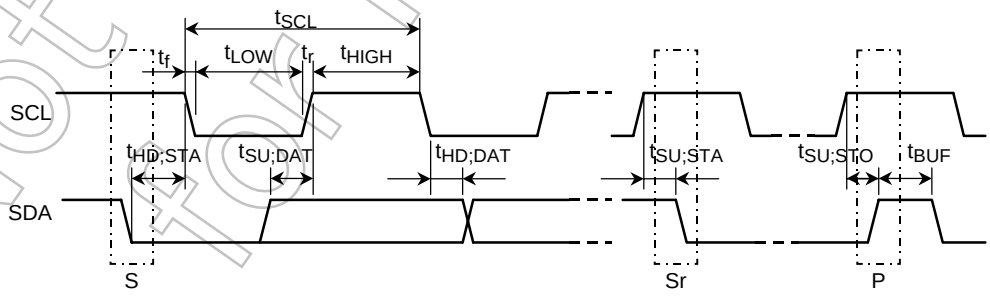
Note 1: SCL clock low width (output) is calculated with (2 (n − 1) + 4) T.
Standard mode: 6 μsec@Typ (fsys = 8 MHz, n = 4)
Fast mode: 1.5 μsec@Typ (fsys = 32 MHz, n = 4)

Note 2: SCL clock high width (output) is calculated with (2 (n − 1)) T.
Standard mode: 4 μsec@Typ (fsys = 8MHz, n = 4)
Fast mode: 1 μsec@Typ (fsys = 32 MHz, n = 4)

Note 3: The output data hold time is equal to 12x.

Note 4: The Philips I^2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the fall edge of SCL. However, TMP1962C10BXBG SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software-dependent



S: START condition
Sr: Repeated START condition
P: STOP condition

Note 6: To operate the SBI in I^2C Fast mode, the fysy frequency must be no less than 20 MHz. To operate the SBI in I^2C Standard mode, the fsys frequency must be no less than 4 MHz.

(2) Clock-Synchronous 8-Bit SIO Mode

In the table below, the letters x and T represent the fsys and ϕT0 cycle periods, respectively. The letter n denotes the value of n programmed into the SCK[2:0] (SCL output frequency select) field in the SBI0CR1.

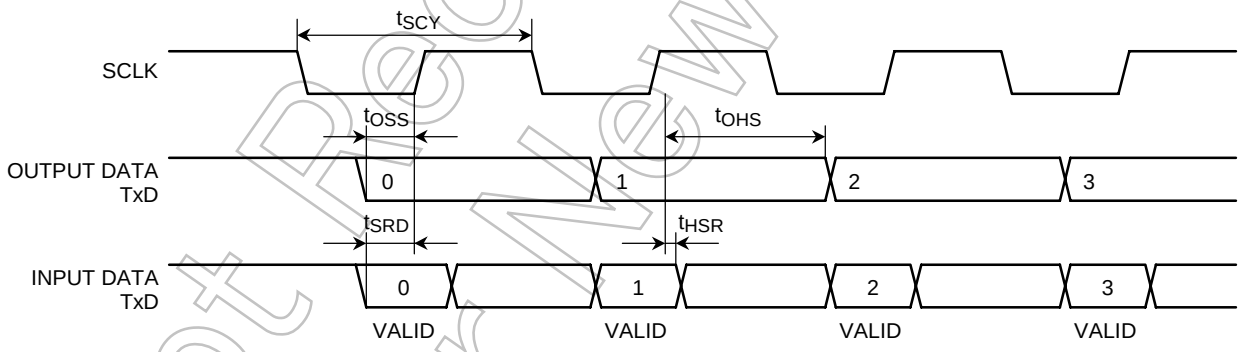
The electrical specifications below are for an SCK signal with a 50% duty cycle.

1. SCK Input Mode

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
SCK period	t_{SCY}	16x		395		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - (6x + 30)$		19		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) + 4x$		296		ns
SI data valid to SCK rise	t_{SRD}	0		0		ns
SI data hold after SCK rise	t_{HSR}	4x + 10		108		ns

2. SCK Output Mode

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
SCK period (programmable)	t_{SCY}	$2^n \cdot T$		1000		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - 20$		480		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) - 20$		480		ns
SI data valid to SCK rise	t_{SRD}	2x + 30		92		ns
SI data hold after SCK rise	t_{HSR}	0		0		ns



22.10 Event Counter

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2X + 100		149		ns
Clock high pulse width	t_{VCKH}	2X + 100		149		ns

22.11 Timer Capture

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2X + 100		149		ns
High pulse width	t_{CPH}	2X + 100		149		ns

22.12 General Interrupts

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTA	t_{INTAL}	X + 100		125		ns
High pulse width for INT0-INTA	t_{INTAH}	X + 100		125		ns

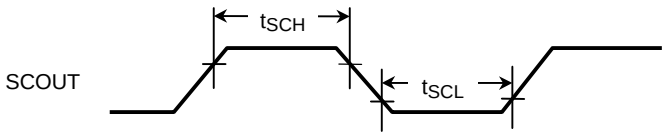
22.13 $\overline{\text{NMI}}$ and STOP Wake-up Interrupts

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for $\overline{\text{NMI}}$ and INT0-INT4	t_{INTBL}	100		100		ns
High pulse width for INT0-INT4	t_{INTBH}	100		100		ns

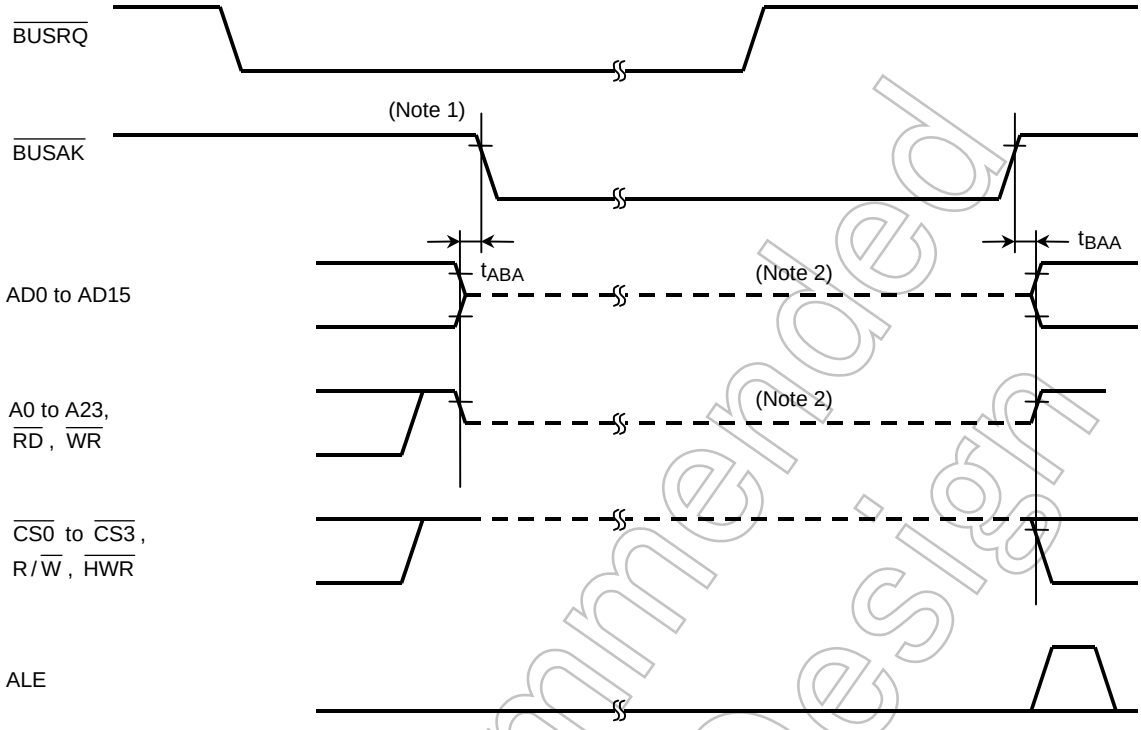
22.14 SCOUT Pin

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Clock high pulse width	t_{SCH}	0.5T – 5		7.4		ns
Clock low pulse width	t_{SCL}	0.5T – 5		7.4		ns

Note: In the above table, the letter T represents the cycle period of the SCOUT output clock.



22.15 Bus Request and Bus Acknowledge Signals



Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Bus float to $\overline{BUSAK}$ asserted	t_{ABA}	0	80	0	80	ns
Bus float after $\overline{BUSAK}$ negated	t_{BAA}	0	80	0	80	ns

Note 1: If the current bus cycle has not terminated due to wait-state insertion, the TMP1962C10BXBG does not respond to $\overline{BUSRQ}$ until the wait state ends.

Note 2: This broken line indicates that output buffers are disabled, not that the signals are at indeterminate states. The pin holds the last logic value present at that pin before the bus is relinquished. This is dynamically accomplished through external load capacitances. The equipment manufacturer may maintain the bus at a predefined state by means of off-chip restores, but he or she should designconsidering the time (determined by the CR constant) it takes for a signal to reach a desired state. The on-chip, integrated programmable pull-up/pull-down resistors remain active, depending on internal signal states.

22.16 KWUP Input

Pull-up Register Inactive

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY0-D	tky_{TBL}	100		100		ns
High pulse width for KEY0-D	tky_{TBH}	100		100		ns

Static Pull-up

Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY0-D	tky_{TBL}	100		100		ns

Dynamic Pull-up

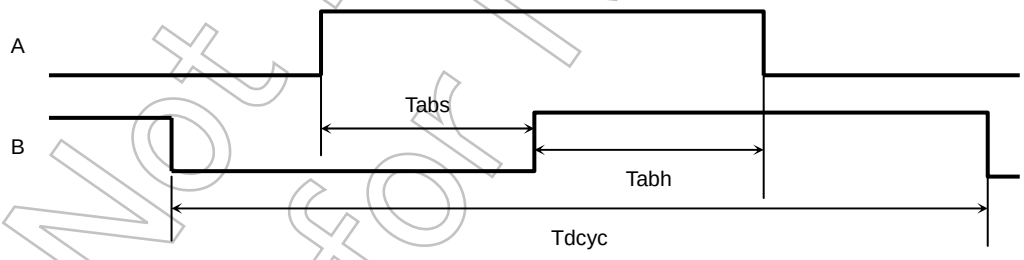
Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY0-D	tkyTBL	T2 + 100		T2 + 100		ns

T2: Dynamic pull-up frequency

22.17 Dual Pulse Input

Parameter	Symbol	Equation		40.5MHz		Unit
		Min	Max	Min	Max	
Dual input pulse period	Tdcyc	8Y		395		ns
Dual input pulse setup	Tabs	Y + 20		70		ns
Dual input pulse hold	Tabh	Y + 20		70		ns

Y: Sampling clock (fsys/2)



22.18 ADTRG Input

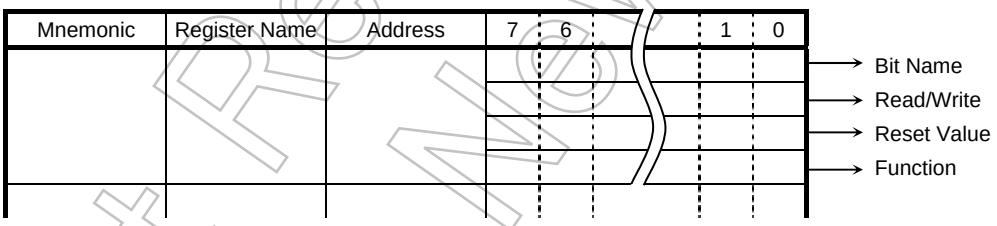
Parameter	Symbol	Equation		40.5 MHz		Unit
		Min	Max	Min	Max	
ADRG low level pulse width	tad_L	fsysy/2 + 20		32.4		ns
ADTRG high level pulse interval	Tadh	fsysy/2 + 20		32.4		ns

23. I/O Register Summary

The internal I/O registers occupy 8-kbyte addresses from FFFFE000H through FFFFFFFFH. (Registers specified as Big-endian)

1. Ports
2. Watchdog Timer (WDT)
3. Real-Time Clock (RTC)
4. 8-Bit Timer
5. 16-Bit Timer
6. I^2CBUS/Serial I/O (SIO)
7. UART/Serial I/O (SIO)
8. 10-Bit A/D Converter (ADC)
9. Key on Wake-up (KWUP)
10. 32-Bit Input Capture
11. 32-Bit Output Compare
12. Interrupt Controller (INTC)
13. DMA Controller (DMAC)
14. Chip Select (CS)/Wait Controller
15. Clock Generator (CG)
16. Flash Control
17. ROM Correction

Table Organization



Access

R/W: Read/write. The user can read and write the register bit.

R: Read only.

W: Write only.

W*: The user can read and write the register bit, but a read always returns a value of 1.

Big-Endian

[1] PORT

Address	Mnemonic
FFFFF000H	
1H	P0CR
2H	P1
3H	P0
4H	
5H	
6H	P1FC
7H	P1CR
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF010H	
1H	P2
2H	
3H	
4H	
5H	
6H	P2FC
7H	P2CR
8H	P3FC
9H	P3CR
AH	
BH	P3
CH	
DH	P4
EH	
FH	

Address	Mnemonic
FFFFF020H	
1H	
2H	P4FC
3H	P4CR
4H	
5H	
6H	
7H	
8H	
9H	
AH	P6
BH	P5
CH	P6FC
DH	P6CR
EH	P5FC
FH	P5CR

Address	Mnemonic
FFFFF030H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF040H	PA
1H	P9
2H	P8
3H	P7
4H	PACR
5H	
6H	
7H	
8H	PAFC
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF050H	PE
1H	PD
2H	PC
3H	PB
4H	PECR
5H	PDCR
6H	PCCR
7H	PBCR
8H	PEFC
9H	PDFC
AH	PCFC
BH	PBFC
CH	PEODE
DH	PDODE
EH	PCODE
FH	

Address	Mnemonic
FFFFF060H	PI
1H	PH
2H	PG
3H	PF
4H	PICR
5H	PHCR
6H	PGCR
7H	PFCR
8H	PIFC
9H	PHFC
AH	PGFC
BH	PFFC
CH	
DH	
EH	
FH	PFODE

Address	Mnemonic
FFFFF070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF0C0H	PM
1H	PL
2H	PK
3H	PJ
4H	PMCR
5H	PLCR
6H	PKCR
7H	PJCR
8H	
9H	PLFC
AH	PKFC
BH	PJFC
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF0D0H	
1H	PP
2H	PO
3H	PN
4H	
5H	PPCR
6H	POCR
7H	PNCR
8H	
9H	
AH	
BH	PNFC
CH	
DH	
EH	
FH	PNODE

Address	Mnemonic
FFFFF0E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF0F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	(reserved)
DH	(reserved)
EH	(reserved)
FH	(reserved)

Big-Endian

[2] WDT

Address	Mnemonic
FFFFF090H	
1H	
2H	WDCR
3H	WDMOD
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] RTC

Address	Mnemonic
FFFFF0A0H	
1H	
2H	
3H	RTCCR
4H	
5H	
6H	
7H	RTCREG
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF0B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[4] 8-Bit Timer

Address	Mnemonic
FFFFF100H	TA1REG
1H	TA0REG
2H	TA01CR
3H	TA01RUN
4H	TAG0ST
5H	TAG0IM
6H	TA1FFCR
7H	TA01MOD
8H	TA3REG
9H	TA2REG
AH	TA23CR
BH	TA23RUN
CH	(reserved)
DH	(reserved)
EH	TA3FFCR
FH	TA23MOD

Address	Mnemonic
FFFFF110H	TA5REG
1H	TA4REG
2H	TA45CR
3H	TA45RUN
4H	TAG1ST
5H	TAG1IM
6H	TA5FFCR
7H	TA45MOD
8H	TA7REG
9H	TA6REG
AH	TA67CR
BH	TA67RUN
CH	(reserved)
DH	(reserved)
EH	TA7FFCR
FH	TA67MOD

Address	Mnemonic
FFFFF120H	TA9REG
1H	TA8REG
2H	TA89CR
3H	TA89RUN
4H	TAG2ST
5H	TAG2IM
6H	TA9FFCR
7H	TA89MOD
8H	TABREG
9H	TAAREG
AH	TAABCR
BH	TAABRUN
CH	(reserved)
DH	(reserved)
EH	TABFFCR
FH	TAABMOD

Address	Mnemonic
FFFFF130H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] 16-Bit Timer

Address	Mnemonic
FFFFF140H	TB0FFCR
1H	TB0MOD
2H	TB0CR
3H	TB0RUN
4H	(reserved)
5H	(reserved)
6H	
7H	TB0ST
8H	TB0RG1H
9H	TB0RG1L
AH	TB0RG0H
BH	TB0RG0L
CH	TB0CP1H
DH	TB0CP1L
EH	TB0CP0H
FH	TB0CP0L

Address	Mnemonic
FFFFF150H	TB1FFCR
1H	TB1MOD
2H	TB1CR
3H	TB1RUN
4H	(reserved)
5H	(reserved)
6H	
7H	TB1ST
8H	TB1RG1H
9H	TB1RG1L
AH	TB1RG0H
BH	TB1RG0L
CH	TB1CP1H
DH	TB1CP1L
EH	TB1CP0H
FH	TB1CP0L

Address	Mnemonic
FFFFF160H	TB2FFCR
1H	TB2MOD
2H	TB2CR
3H	TB2RUN
4H	(reserved)
5H	(reserved)
6H	
7H	TB2ST
8H	TB2RG1H
9H	TB2RG1L
AH	TB2RG0H
BH	TB2RG0L
CH	TB2CP1H
DH	TB2CP1L
EH	TB2CP0H
FH	TB2CP0L

Address	Mnemonic
FFFFF170H	TB3FFCR
1H	TB3MOD
2H	TB3CR
3H	TB3RUN
4H	(reserved)
5H	(reserved)
6H	
7H	TB3ST
8H	TB3RG1H
9H	TB3RG1L
AH	TB3RG0H
BH	TB3RG0L
CH	TB3CP1H
DH	TB3CP1L
EH	TB3CP0H
FH	TB3CP0L

Big-Endian

Address	Mnemonic
FFFFF180H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF190H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF1A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF1B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF1C0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF1D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF1E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF1F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF200H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF210H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF220H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF230H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-Endian

Address	Mnemonic
FFFFF240H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[6] I^2C/SIO

Address	Mnemonic
FFFFF250H	SBICR2/SR
1H	I2CAR
2H	SBIDBR
3H	SBICR1
4H	SBICR0
5H	
6H	SBIBR1
7H	SBIBR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[7] UART/SIO

Address	Mnemonic
FFFFF260H	BR0CR
1H	SC0MOD0
2H	SC0CR
3H	SC0BUF
4H	
5H	SC0MOD2
6H	SC0MOD1
7H	BR0ADD
8H	BR1CR
9H	SC1MOD0
AH	SC1CR
BH	SC1BUF
CH	
DH	SC1MOD2
EH	SC1MOD1
FH	BR1ADD

Address	Mnemonic
FFFFF270H	BR2CR
1H	SC2MOD0
2H	SC2CR
3H	SC2BUF
4H	
5H	SC2MOD2
6H	SC2MOD1
7H	BR2ADD
8H	BR3CR
9H	SC3MOD0
AH	SC3CR
BH	SC3BUF
CH	
DH	SC3MOD2
EH	SC3MOD1
FH	BR3ADD

Address	Mnemonic
FFFFF280H	BR4CR
1H	SC4MOD0
2H	SC4CR
3H	SC4BUF
4H	
5H	SC4MOD2
6H	SC4MOD1
7H	BR4ADD
8H	BR5CR
9H	SC5MOD0
AH	SC5CR
BH	SC5BUF
CH	
DH	SC5MOD2
EH	SC5MOD1
FH	BR5ADD

Address	Mnemonic
FFFFF290H	BR6CR
1H	SC6MOD0
2H	SC6CR
3H	SC6BUF
4H	
5H	SC6MOD2
6H	SC6MOD1
7H	BR6ADD
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-Endian

[8] 10-Bit ADC

Address	Mnemonic
FFFFF300H	ADREG19H
1H	ADREG19L
2H	ADREG08H
3H	ADREG08L
4H	ADREG3BH
5H	ADREG3BL
6H	ADREG2AH
7H	ADREG2AL
8H	ADREG5DH
9H	ADREG5DL
AH	ADREG4CH
BH	ADREG4CL
CH	ADREG7FH
DH	ADREG7FL
EH	ADREG6EH
FH	ADREG6EL

Address	Mnemonic
FFFFF310H	
1H	
2H	
3H	
4H	
5H	
6H	ADCOMREGH
7H	ADCOMREGL
8H	ADMOD3
9H	(reserved)
AH	ADMOD1
BH	ADMOD0
CH	ADCLK
DH	
EH	
FH	ADMOD4

[9] KWUP

Address	Mnemonic
FFFFF360H	KWUPST3
1H	KWUPST2
2H	KWUPST1
3H	KWUPST0
4H	KWUPST7
5H	KWUPST6
6H	KWUPST5
7H	KWUPST4
8H	KWUPSTB
9H	KWUPSTA
AH	KWUPST9
BH	KWUPST8
CH	
DH	
EH	KWUPSTD
FH	KWUPSTC

Address	Mnemonic
FFFFF370H	
1H	
2H	KWUPCNT
3H	KWUPCLR
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] 32-Bit Input Capture

Address	Mnemonic
FFFFF400H	
1H	TBTCR
2H	TBTRUN
3H	TCCR
4H	TBTCAP3
5H	TBTCAP2
6H	TBTCAP1
7H	TBTCAP0
8H	TCG1ST
9H	TCG1IM
AH	TCG0ST
BH	TCG0IM
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFF410H	
1H	
2H	
3H	CAP0CR
4H	TCCAP0HH
5H	TCCAP0HL
6H	TCCAP0LH
7H	TCCAP0LL
8H	
9H	
AH	
BH	CAP1CR
CH	TCCAP1HH
DH	TCCAP1HL
EH	TCCAP1LH
FH	TCCAP1LL

Address	Mnemonic
FFFFF420H	
1H	
2H	
3H	CAP2CR
4H	TCCAP2HH
5H	TCCAP2HL
6H	TCCAP2LH
7H	TCCAP2LL
8H	
9H	
AH	
BH	CAP3CR
CH	TCCAP3HH
DH	TCCAP3HL
EH	TCCAP3LH
FH	TCCAP3LL

Address	Mnemonic
FFFFF430H	
1H	
2H	
3H	CAP4CR
4H	TCCAP4HH
5H	TCCAP4HL
6H	TCCAP4LH
7H	TCCAP4LL
8H	
9H	
AH	
BH	CAP5CR
CH	TCCAP5HH
DH	TCCAP5HL
EH	TCCAP5LH
FH	TCCAP5LL

[11] 32-Bit Output Compare

Address	Mnemonic
FFFFF440H	
1H	
2H	
3H	CAP6CR
4H	TCCAP6HH
5H	TCCAP6HL
6H	TCCAP6LH
7H	TCCAP6LL
8H	
9H	
AH	
BH	CAP7CR
CH	TCCAP7HH
DH	TCCAP7HL
EH	TCCAP7LH
FH	TCCAP7LL

Address	Mnemonic
FFFFF450H	TCCMP0HH
1H	TCCMP0HL
2H	TCCMP0LH
3H	TCCMP0LL
4H	TCCMP1HH
5H	TCCMP1HL
6H	TCCMP1LH
7H	TCCMP1LL
8H	TCCMP2HH
9H	TCCMP2HL
AH	TCCMP2LH
BH	TCCMP2LL
CH	TCCMP3HH
DH	TCCMP3HL
EH	TCCMP3LH
FH	TCCMP3LL

Address	Mnemonic
FFFFF460H	TCCMP4HH
1H	TCCMP4HL
2H	TCCMP4LH
3H	TCCMP4LL
4H	TCCMP5HH
5H	TCCMP5HL
6H	TCCMP5LH
7H	TCCMP5LL
8H	TCCMP6HH
9H	TCCMP6HL
AH	TCCMP6LH
BH	TCCMP6LL
CH	TCCMP7HH
DH	TCCMP7HL
EH	TCCMP7LH
FH	TCCMP7LL

Address	Mnemonic
FFFFF470H	CMPCTL3
1H	CMPCTL2
2H	CMPCTL1
3H	CMPCTL0
4H	CMPCTL7
5H	CMPCTL6
6H	CMPCTL5
7H	CMPCTL4
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-Endian

[12] INTC

Address	Mnemonic
FFFFE000H	IMC0H
1H	
2H	IMC0L
3H	
4H	IMC1H
5H	
6H	IMC1L
7H	
8H	IMC2H
9H	
AH	IMC2L
BH	
CH	IMC3H
DH	
EH	IMC3L
FH	

Address	Mnemonic
FFFFE010H	IMC4H
1H	
2H	IMC4L
3H	
4H	IMC5H
5H	
6H	IMC5L
7H	
8H	IMC6H
9H	
AH	IMC6L
BH	
CH	IMC7H
DH	
EH	IMC7L
FH	

Address	Mnemonic
FFFFE020H	IMC8H
1H	
2H	IMC8L
3H	
4H	IMC9H
5H	
6H	IMC9L
7H	
8H	IMCAH
9H	
AH	IMCAL
BH	
CH	IMCBH
DH	
EH	IMCBL
FH	

Address	Mnemonic
FFFFE030H	IMCCH
1H	
2H	IMCCL
3H	
4H	IMCDH
5H	
6H	IMCDL
7H	
8H	IMCEH
9H	
AH	IMCEL
BH	
CH	IMCFH
DH	
EH	IMCFL
FH	

Address	Mnemonic
FFFFE040H	HIVR
1H	
2H	LIVR
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE050H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE060H	INTCLR
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-Endian

[13] DAMC

Address	Mnemonic
FFFFE200H	CCR0H
1H	
2H	CCR0L
3H	
4H	CSR0H
5H	
6H	CSR0L
7H	
8H	SAR0H
9H	
AH	SAR0L
BH	
CH	DAR0H
DH	
EH	DAR0L
FH	

Address	Mnemonic
FFFFE210H	BCR0H
1H	
2H	BCR0L
3H	
4H	
5H	
6H	
7H	
8H	DTCR0H
9H	
AH	DTCR0L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE220H	CCR1H
1H	
2H	CCR1L
3H	
4H	CSR1H
5H	
6H	CSR1L
7H	
8H	SAR1H
9H	
AH	SAR1L
BH	
CH	DAR1H
DH	
EH	DAR1L
FH	

Address	Mnemonic
FFFFE230H	BCR1H
1H	
2H	BCR1L
3H	
4H	
5H	
6H	
7H	
8H	DTCR1H
9H	
AH	DTCR1L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE240H	CCR2H
1H	
2H	CCR2L
3H	
4H	CSR2H
5H	
6H	CSR2L
7H	
8H	SAR2H
9H	
AH	SAR2L
BH	
CH	DAR2H
DH	
EH	DAR2L
FH	

Address	Mnemonic
FFFFE250H	BCR2H
1H	
2H	BCR2L
3H	
4H	
5H	
6H	
7H	
8H	DTCR2H
9H	
AH	DTCR2L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE260H	CCR3H
1H	
2H	CCR3L
3H	
4H	CSR3H
5H	
6H	CSR3L
7H	
8H	SAR3H
9H	
AH	SAR3L
BH	
CH	DAR3H
DH	
EH	DAR3L
FH	

Address	Mnemonic
FFFFE270H	BCR3H
1H	
2H	BCR3L
3H	
4H	
5H	
6H	
7H	
8H	DTCR3H
9H	
AH	DTCR3L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE280H	CCR4H
1H	
2H	CCR4L
3H	
4H	CSR4H
5H	
6H	CSR4L
7H	
8H	SAR4H
9H	
AH	SAR4L
BH	
CH	DAR4H
DH	
EH	DAR4L
FH	

Address	Mnemonic
FFFFE290H	BCR4H
1H	
2H	BCR4L
3H	
4H	
5H	
6H	
7H	
8H	DTCR4H
9H	
AH	DTCR4L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE2A0H	CCR5H
1H	
2H	CCR5L
3H	
4H	CSR5H
5H	
6H	CSR5L
7H	
8H	SAR5H
9H	
AH	SAR5L
BH	
CH	DAR5H
DH	
EH	DAR5L
FH	

Address	Mnemonic
FFFFE2B0H	BCR5H
1H	
2H	BCR5L
3H	
4H	
5H	
6H	
7H	
8H	DTCR5H
9H	
AH	DTCR5L
BH	
CH	
DH	
EH	
FH	

Big-Endian

Address	Mnemonic
FFFFE2C0H	CCR6H
1H	
2H	CCR6L
3H	
4H	CSR6H
5H	
6H	CSR6L
7H	
8H	SAR6H
9H	
AH	SAR6L
BH	
CH	DAR6H
DH	
EH	DAR6L
FH	

Address	Mnemonic
FFFFE2D0H	BCR6H
1H	
2H	BCR6L
3H	
4H	
5H	
6H	
7H	
8H	DTCR6H
9H	
AH	DTCR6L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE2E0H	CCR7H
1H	
2H	CCR7L
3H	
4H	CSR7H
5H	
6H	CSR7L
7H	
8H	SAR7H
9H	
AH	SAR7L
BH	
CH	DAR7H
DH	
EH	DAR7L
FH	

Address	Mnemonic
FFFFE2F0H	BCR7H
1H	
2H	BCR7L
3H	
4H	
5H	
6H	
7H	
8H	DTCR7H
9H	
AH	DTCR7L
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE300H	DCRH
1H	
2H	DCRL
3H	
4H	RSRH
5H	
6H	RSRL
7H	
8H	
9H	
AH	
BH	
CH	DHRH
DH	
EH	DHRL
FH	

Address	Mnemonic
FFFFE310H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE320H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE330H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE340H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE350H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE360H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE370H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big- Endian

[14] CS/Wait Controller

Address	Mnemonic
FFFFE400H	BMA0
1H	
2H	
3H	
4H	BMA1
5H	
6H	
7H	
8H	BMA2
9H	
AH	
BH	
CH	BMA3
DH	
EH	
FH	

Address	Mnemonic
FFFFE410H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE480H	B01CS
1H	
2H	
3H	
4H	B23CS
5H	
6H	
7H	
8H	BEXCS
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE490H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] CG

Address	Mnemonic
FFFFEE00H	SYSCR3
1H	SYSCR2
2H	SYSCR1
3H	SYSCR0
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFEE10H	IMCGA3
1H	IMCGA2
2H	IMCGA1
3H	IMCGA0
4H	IMCGB3
5H	IMCGB2
6H	IMCGB1
7H	IMCGB0
8H	IMCGC3
9H	IMCGC2
AH	IMCGC1
BH	IMCGC0
CH	IMCGD3
DH	IMCGD2
EH	IMCGD1
FH	IMCGD0

Address	Mnemonic
FFFFEE20H	EICRCG
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFEE40H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] Flash Control

Address	Mnemonic
FFFFE510H	SEQMOD
1H	
2H	
3H	
4H	SEQCNT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Mnemonic
FFFFE520H	FLCS
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] ROM Correction

Address	Mnemonic
FFFFE540H	ADDREG0
1H	
2H	
3H	
4H	ADDREG1
5H	
6H	
7H	
8H	ADDREG2
9H	
AH	
BH	
CH	ADDREG3
DH	
EH	
FH	

Address	Mnemonic
FFFFE550H	ADDREG4
1H	
2H	
3H	
4H	ADDREG5
5H	
6H	
7H	
8H	ADDREG6
9H	
AH	
BH	
CH	ADDREG7
DH	
EH	
FH	

Little-Endian

[1] PORT

ADR	Mnemonic
FFFFF000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF010H	
1H	
2H	P2
3H	
4H	P2CR
5H	P2FC
6H	
7H	
8H	P3
9H	
AH	P3CR
BH	P3FC
CH	
DH	
EH	P4
FH	

ADR	Mnemonic
FFFFF020H	P4CR
1H	P4FC
2H	
3H	
4H	
5H	
6H	
7H	
8H	P5
9H	P6
AH	
BH	
CH	P5CR
DH	P5FC
EH	P6CR
FH	P6FC

ADR	Mnemonic
FFFFF030H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF040H	P7
1H	P8
2H	P9
3H	PA
4H	
5H	
6H	
7H	PACR
8H	
9H	
AH	
BH	PAFC
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF050H	PB
1H	PC
2H	PD
3H	PE
4H	PBCR
5H	PCCR
6H	PDCR
7H	PECR
8H	PBFC
9H	PCFC
AH	PDFC
BH	PEFC
CH	
DH	PCODE
EH	PDODE
FH	PEODE

ADR	Mnemonic
FFFFF060H	PF
1H	PG
2H	PH
3H	PI
4H	PFCR
5H	PGCR
6H	PHCR
7H	PICR
8H	PFFC
9H	PGFC
AH	PHFC
BH	PIFC
CH	PFODE
DH	
EH	
FH	

ADR	Mnemonic
FFFFF070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF0C0H	PJ
1H	PK
2H	PL
3H	PM
4H	PJCR
5H	PKCR
6H	PLCR
7H	PMCR
8H	PJFC
9H	PKFC
AH	PLFC
BH	PMFC
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF0D0H	PN
1H	PO
2H	PP
3H	
4H	PNCR
5H	POCR
6H	PPCR
7H	
8H	PNFC
9H	POFC
AH	PPFC
BH	
CH	PNODE
DH	
EH	
FH	

ADR	Mnemonic
FFFFF0E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF0F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	(reserved)
DH	(reserved)
EH	(reserved)
FH	(reserved)

Little-Endian

[2] WDT

ADR	Mnemonic
FFFFF090H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] RTC

ADR	Mnemonic
FFFFF0A0H	RTCCR
1H	
2H	
3H	
4H	RTCREG
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF0B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[4] 8-Bit Timer

ADR	Mnemonic
FFFFF100H	TA01RUN
1H	TA01CR
2H	TA0REG
3H	TA1REG
4H	TA01MOD
5H	TA1FFCR
6H	TAG0IM
7H	TAG0ST
8H	TA23RUN
9H	TA23CR
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	(reserved)
FH	(reserved)

ADR	Mnemonic
FFFFF110H	TA45RUN
1H	TA45CR
2H	TA4REG
3H	TA5REG
4H	TA45MOD
5H	TA5FFCR
6H	TAG1IM
7H	TAG1ST
8H	TA67RUN
9H	TA67CR
AH	TA6REG
BH	TA7REG
CH	TA67MOD
DH	TA7FFCR
EH	(reserved)
FH	(reserved)

ADR	Mnemonic
FFFFF120H	TA89RUN
1H	TA89CR
2H	TA8REG
3H	TA9REG
4H	TA89MOD
5H	TA9FFCR
6H	TAG2IM
7H	TAG2ST
8H	TAABRUN
9H	TAABCR
AH	TAAREG
BH	TABREG
CH	TAABMOD
DH	TABFFCR
EH	(reserved)
FH	(reserved)

ADR	Mnemonic
FFFFF130H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] 16-Bit Timer

ADR	Mnemonic
FFFFF140H	TB0RUN
1H	TB0CR
2H	TB0MOD
3H	TB0FFCR
4H	TB0ST
5H	
6H	(reserved)
7H	(reserved)
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

ADR	Mnemonic
FFFFF150H	TB1RUN
1H	TB1CR
2H	TB1MOD
3H	TB1FFCR
4H	TB1ST
5H	
6H	(reserved)
7H	(reserved)
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

ADR	Mnemonic
FFFFF160H	TB2RUN
1H	TB2CR
2H	TB2MOD
3H	TB2FFCR
4H	TB2ST
5H	
6H	(reserved)
7H	(reserved)
8H	TB2RG0L
9H	TB2RG0H
AH	TB2RG1L
BH	TB2RG1H
CH	TB2CP0L
DH	TB2CP0H
EH	TB2CP1L
FH	TB2CP1H

ADR	Mnemonic
FFFFF170H	TB3RUN
1H	TB3CR
2H	TB3MOD
3H	TB3FFCR
4H	TB3ST
5H	
6H	(reserved)
7H	(reserved)
8H	TB3RG0L
9H	TB3RG0H
AH	TB3RG1L
BH	TB3RG1H
CH	TB3CP0L
DH	TB3CP0H
EH	TB3CP1L
FH	TB3CP1H

Little-Endian

ADR	Mnemonic
FFFFF180H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF190H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF1A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF1B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF1C0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF1D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF1E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF1F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF200H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF210H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF220H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFF230H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-Endian

ADR	Mnemonic
FFFFF240H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[6] I^2C/SIO

ADR	Mnemonic
FFFFF250H	SBICR1
1H	SBIDBR
2H	I2CAR
3H	SBICR2/SR
4H	SBIBR0
5H	SBIBR1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[7] UART/SIO

ADR	Mnemonic
FFFFF260H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	SC0MOD2
7H	
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	SC1MOD2
FH	

ADR	Mnemonic
FFFFF270H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	SC2MOD2
7H	
8H	SC3BUF
9H	SC3CR
AH	SC3MOD0
BH	BR3CR
CH	BR3ADD
DH	SC3MOD1
EH	SC3MOD2
FH	

ADR	Mnemonic
FFFFF280H	SC4BUF
1H	SC4CR
2H	SC4MOD0
3H	BR4CR
4H	BR4ADD
5H	SC4MOD1
6H	SC4MOD2
7H	
8H	SC5BUF
9H	SC5CR
AH	SC5MOD0
BH	BR5CR
CH	BR5ADD
DH	SC5MOD1
EH	SC5MOD2
FH	

ADR	Mnemonic
FFFFF290H	SC6BUF
1H	SC6CR
2H	SC6MOD0
3H	BR6CR
4H	BR6ADD
5H	SC6MOD1
6H	SC6MOD2
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-Endian

[8] 10-Bit ADC

ADR	Mnemonic
FFFFF300H	ADREG08L
1H	ADREG08H
2H	ADREG19L
3H	ADREG19H
4H	ADREG2AL
5H	ADREG2AH
6H	ADREG3BL
7H	ADREG3BH
8H	ADREG4CL
9H	ADREG4CH
AH	ADREG5DL
BH	ADREG5DH
CH	ADREG6EL
DH	ADREG6EH
EH	ADREG7FL
FH	ADREG7FH

ADR	Mnemonic
FFFFF310H	
1H	
2H	
3H	
4H	ADCOMREGL
5H	ADCOMREGH
6H	
7H	
8H	ADMOD0
9H	ADMOD1
AH	(reserved)
BH	ADMOD3
CH	ADMOD4
DH	
EH	
FH	ADCLK

[9] KWUP

ADR	Mnemonic
FFFFF360H	KWUPST0
1H	KWUPST1
2H	KWUPST2
3H	KWUPST3
4H	KWUPST4
5H	KWUPST5
6H	KWUPST6
7H	KWUPST7
8H	KWUPST8
9H	KWUPST9
AH	KWUPSTA
BH	KWUPSTB
CH	KWUPSTC
DH	KWUPSTD
EH	
FH	

ADR	Mnemonic
FFFFF370H	KWUPCLR
1H	KWUPCNT
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] 32-Bit Input Capture

ADR	Mnemonic
FFFFF 400H	TCCR
1H	TBTRUN
2H	TBTCR
3H	
4H	TBTCAP0
5H	TBTCAP1
6H	TBTCAP2
7H	TBTCAP3
8H	TCG0IM
9H	TCG0ST
AH	TCG1IM
BH	TCG1ST
CH	TCG2IM
DH	TCG2ST
EH	
FH	

ADR	Mnemonic
FFFFF410H	CAP0CR
1H	
2H	
3H	
4H	TCCAP0LL
5H	TCCAP0LH
6H	TCCAP0HL
7H	TCCAP0HH
8H	CAP1CR
9H	
AH	
BH	
CH	TCCAP1LL
DH	TCCAP1LH
EH	TCCAP1HL
FH	TCCAP1HH

ADR	Mnemonic
FFFFF420H	CAP2CR
1H	
2H	
3H	
4H	TCCAP2LL
5H	TCCAP2LH
6H	TCCAP2HL
7H	TCCAP2HH
8H	CAP3CR
9H	
AH	
BH	
CH	TCCAP3LL
DH	TCCAP3LH
EH	TCCAP3HL
FH	TCCAP3HH

ADR	Mnemonic
FFFFF430H	CAP4CR
1H	
2H	
3H	
4H	TCCAP4LL
5H	TCCAP4LH
6H	TCCAP4HL
7H	TCCAP4HH
8H	CAP5CR
9H	
AH	
BH	
CH	TCCAP5LL
DH	TCCAP5LH
EH	TCCAP5HL
FH	TCCAP5HH

[11] 32-Bit Output Compare

ADR	Mnemonic
FFFFF 440H	CAP6CR
1H	
2H	
3H	
4H	TCCAP6LL
5H	TCCAP6LH
6H	TCCAP6HL
7H	TCCAP6HH
8H	CAP7CR
9H	
AH	
BH	
CH	TCCAP7LL
DH	TCCAP7LH
EH	TCCAP7HL
FH	TCCAP7HH

ADR	Mnemonic
FFFFF450H	TCCMP0LL
1H	TCCMP0LH
2H	TCCMP0HL
3H	TCCMP0HH
4H	TCCMP1LL
5H	TCCMP1LH
6H	TCCMP1HL
7H	TCCMP1HH
8H	TCCMP2LL
9H	TCCMP2LH
AH	TCCMP2HL
BH	TCCMP2HH
CH	TCCMP3LL
DH	TCCMP3LH
EH	TCCMP3HL
FH	TCCMP3HH

ADR	Mnemonic
FFFFF460H	TCCMP4LL
1H	TCCMP4LH
2H	TCCMP4HL
3H	TCCMP4HH
4H	TCCMP5LL
5H	TCCMP5LH
6H	TCCMP5HL
7H	TCCMP5HH
8H	TCCMP6LL
9H	TCCMP6LH
AH	TCCMP6HL
BH	TCCMP6HH
CH	TCCMP7LL
DH	TCCMP7LH
EH	TCCMP7HL
FH	TCCMP7HH

ADR	Mnemonic
FFFFF470H	CMPCTL0
1H	CMPCTL1
2H	CMPCTL2
3H	CMPCTL3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-Endian

[12] INTC

ADR	Mnemonic
FFFFE000H	IMC0L
1H	
2H	IMC0H
3H	
4H	IMC1L
5H	
6H	IMC1H
7H	
8H	IMC2L
9H	
AH	IMC2H
BH	
CH	IMC3L
DH	
EH	IMC3H
FH	

ADR	Mnemonic
FFFFE010H	IMC4L
1H	
2H	IMC4H
3H	
4H	IMC5L
5H	
6H	IMC5H
7H	
8H	IMC6L
9H	
AH	IMC6H
BH	
CH	IMC7L
DH	
EH	IMC7H
FH	

ADR	Mnemonic
FFFFE020H	IMC8L
1H	
2H	IMC8H
3H	
4H	IMC9L
5H	
6H	IMC9H
7H	
8H	IMCAL
9H	
AH	IMCAH
BH	
CH	IMCBL
DH	
EH	IMCBH
FH	

ADR	Mnemonic
FFFFE030H	IMCCL
1H	
2H	IMCCH
3H	
4H	IMCDL
5H	
6H	IMCDH
7H	
8H	IMCEL
9H	
AH	IMCEH
BH	
CH	IMCFL
DH	
EH	IMCFH
FH	

ADR	Mnemonic
FFFFE040H	HIVR
1H	
2H	LIVR
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE050H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE060H	INTCLR
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-Endian

[13] DAMC

ADR	Mnemonic
FFFFE200H	CCR0L
1H	
2H	CCR0H
3H	
4H	CSR0L
5H	
6H	CSR0H
7H	
8H	SAR0L
9H	
AH	SAR0H
BH	
CH	DAR0L
DH	
EH	DAR0H
FH	

ADR	Mnemonic
FFFFE210H	BCR0L
1H	
2H	BCR0H
3H	
4H	
5H	
6H	
7H	
8H	DTCR0L
9H	
AH	DTCR0H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE220H	CCR1L
1H	
2H	CCR1H
3H	
4H	CSR1L
5H	
6H	CSR1H
7H	
8H	SAR1L
9H	
AH	SAR1H
BH	
CH	DAR1L
DH	
EH	DAR1H
FH	

ADR	Mnemonic
FFFFE230H	BCR1L
1H	
2H	BCR1H
3H	
4H	
5H	
6H	
7H	
8H	DTCR1L
9H	
AH	DTCR1H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE240H	CCR2L
1H	
2H	CCR2H
3H	
4H	CSR2L
5H	
6H	CSR2H
7H	
8H	SAR2L
9H	
AH	SAR2H
BH	
CH	DAR2L
DH	
EH	DAR2H
FH	

ADR	Mnemonic
FFFFE250H	BCR2L
1H	
2H	BCR2H
3H	
4H	
5H	
6H	
7H	
8H	DTCR2L
9H	
AH	DTCR1H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE260H	CCR3L
1H	
2H	CCR3H
3H	
4H	CSR3L
5H	
6H	CSR3H
7H	
8H	SAR3L
9H	
AH	SAR3H
BH	
CH	DAR3L
DH	
EH	DAR3H
FH	

ADR	Mnemonic
FFFFE270H	BCR3L
1H	
2H	BCR3H
3H	
4H	
5H	
6H	
7H	
8H	DTCR3L
9H	
AH	DTCR3H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE280H	CCR4L
1H	
2H	CCR4H
3H	
4H	CSR4L
5H	
6H	CSR4H
7H	
8H	SAR4L
9H	
AH	SAR4H
BH	
CH	DAR4L
DH	
EH	DAR4H
FH	

ADR	Mnemonic
FFFFE290H	BCR4L
1H	
2H	BCR4H
3H	
4H	
5H	
6H	
7H	
8H	DTCR4L
9H	
AH	DTCR4H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE2A0H	CCR5L
1H	
2H	CCR5H
3H	
4H	CSR5L
5H	
6H	CSR5H
7H	
8H	SAR5L
9H	
AH	SAR5H
BH	
CH	DAR5L
DH	
EH	DAR5H
FH	

ADR	Mnemonic
FFFFE2B0H	BCR5L
1H	
2H	BCR5H
3H	
4H	
5H	
6H	
7H	
8H	DTCR5L
9H	
AH	DTCR5H
BH	
CH	
DH	
EH	
FH	

Little-Endian

ADR	Mnemonic
FFFFE2C0H	CCR6L
1H	
2H	CCR6H
3H	
4H	CSR6L
5H	
6H	CSR6H
7H	
8H	SAR6L
9H	
AH	SAR6H
BH	
CH	DAR6L
DH	
EH	DAR6H
FH	

ADR	Mnemonic
FFFFE2D0H	BCR6L
1H	
2H	BCR6H
3H	
4H	
5H	
6H	
7H	
8H	DTCR6L
9H	
AH	DTCR6H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE2E0H	CCR7L
1H	
2H	CCR7H
3H	
4H	CSR7L
5H	
6H	CSR7H
7H	
8H	SAR7L
9H	
AH	SAR7H
BH	
CH	DAR7L
DH	
EH	DAR7H
FH	

ADR	Mnemonic
FFFFE2F0H	BCR7L
1H	
2H	BCR7H
3H	
4H	
5H	
6H	
7H	
8H	DTCR7L
9H	
AH	DTCR7H
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE300H	DCRL
1H	
2H	DCRH
3H	
4H	RSRL
5H	
6H	RSRH
7H	
8H	
9H	
AH	
BH	
CH	DHRL
DH	
EH	DHRH
FH	

ADR	Mnemonic
FFFFE310H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE320H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE330H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE340H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE350H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE360H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE370H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-Endian

[14] CS/Wait Controller

ADR	Mnemonic
FFFFE400H	BMA0
1H	
2H	
3H	
4H	BMA1
5H	
6H	
7H	
8H	BMA2
9H	
AH	
BH	
CH	BMA3
DH	
EH	
FH	

ADR	Mnemonic
FFFFE410H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE480H	B01CS
1H	
2H	
3H	
4H	B23CS
5H	
6H	
7H	
8H	BEXCS
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE490H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] CG

ADR	Mnemonic
FFFFEE00H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	SYSCR3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFEE10H	IMCGA0
1H	IMCGA1
2H	IMCGA2
3H	IMCGA3
4H	IMCGB0
5H	IMCGB1
6H	IMCGB2
7H	IMCGB3
8H	IMCGC0
9H	IMCGC1
AH	IMCGC2
BH	IMCGC3
CH	IMCGD0
DH	IMCGD1
EH	IMCGD2
FH	IMCGD3

ADR	Mnemonic
FFFFEE20H	EICRCG
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFEE40H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] Flash Control

ADR	Mnemonic
FFFFE510H	SEQMOD
1H	
2H	
3H	
4H	SEQCNT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Mnemonic
FFFFE520H	FLCS
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] ROM Correction

ADR	Mnemonic
FFFFE540H	ADDREG0
1H	
2H	
3H	
4H	ADDREG1
5H	
6H	
7H	
8H	ADDREG2
9H	
AH	
BH	
CH	ADDREG3
DH	
EH	
FH	

ADR	Mnemonic
FFFFE550H	ADDREG4
1H	
2H	
3H	
4H	ADDREG5
5H	
6H	
7H	
8H	ADDREG6
9H	
AH	
BH	
CH	ADDREG7
DH	
EH	
FH	

24. I/O Port Equivalent-Circuit Diagrams

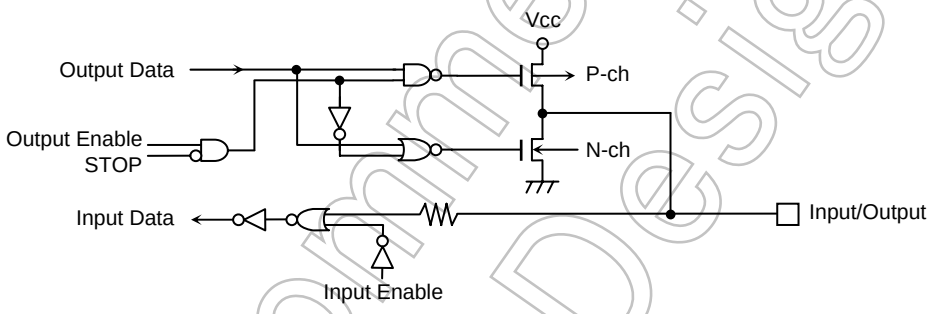
- How to read circuit diagrams

  The circuit diagrams in this chapter are drawn using the same gate symbols as for the 74HCxx Series standard CMOS logic ICs.

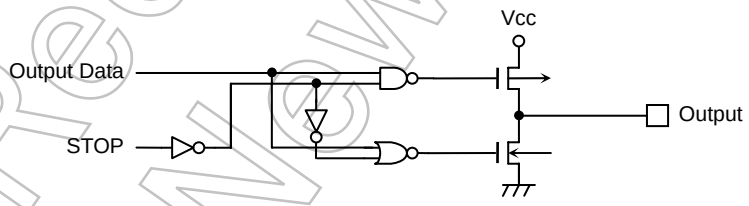
  The signal named STOP has a unique function. This signal goes active-high if the CPU sets the HALT bit when the STBY[1:0] field in the SYSCR2 register is programmed to 01 (i.e., STOP mode) and the Drive Enable (DRVE) bit in the same register is cleared. If the DRVE bit is set, the STOP signal remains inactive (at logic 0).

- The input protection circuit has a resistor in the range of several tens to several hundreds of ohms.

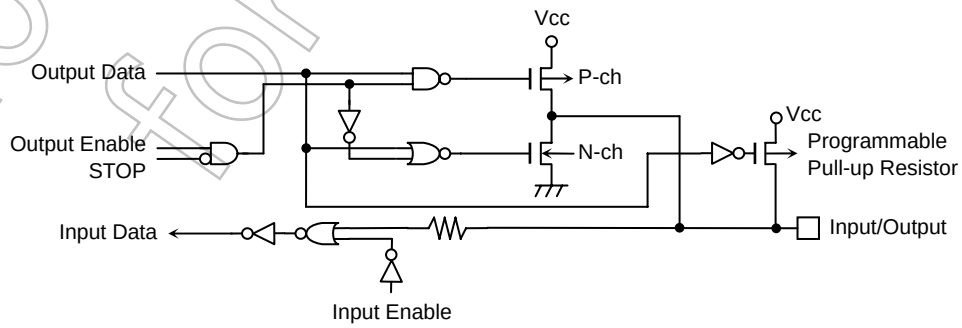
■ P0 (D0–D7/AD0–AD7), P1 (D8–D15/AD8–AD15, A8–A15), P2 (A16–23, A0–7), P37,P44, P5 (A0–A7), P6 (A8–A15), PA, PB, PI7, PC1, PC4, PC7, PD2, PD5, PE1, PF2–PF7, PG, PH, PJ1–PJ4, PL, PM, PN1, PN3–PN7, PO, PP



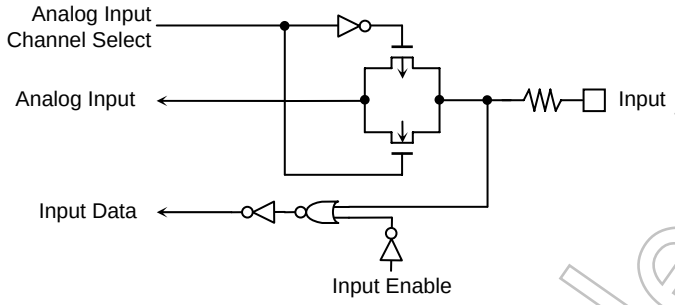
■ P30 ($\overline{RD}$), P31 ($\overline{WR}$), DCLK, PCST3–PCST0, SDAO/TPC, TDO



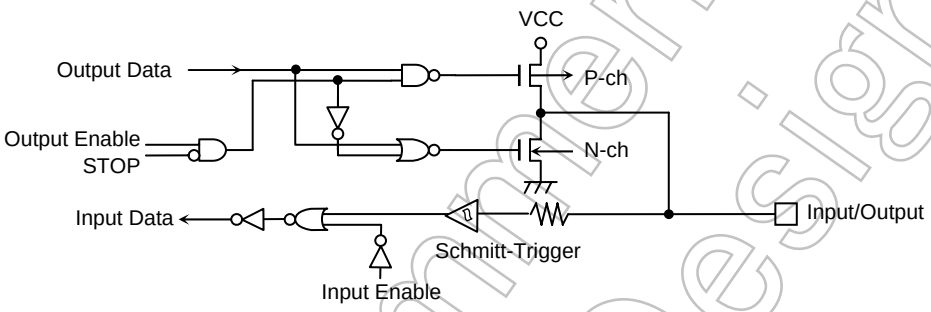
■ P32–P36, P40–P43



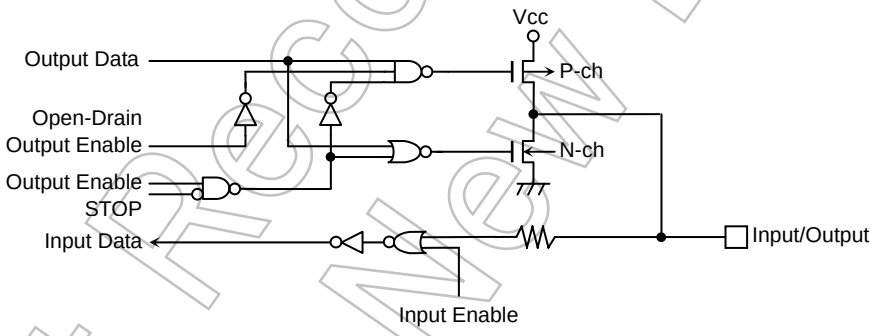
■ P7 (AN0–AN7), P8 (AN8–AN15), P9 (AN16–AN23)



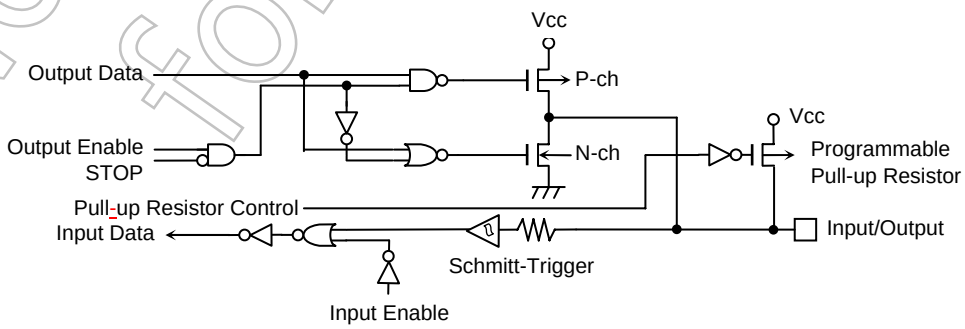
■ PI0–PI6 ($\overline{\text{ADTRG}}$, INT1–INTA), PJ0 (INT0)



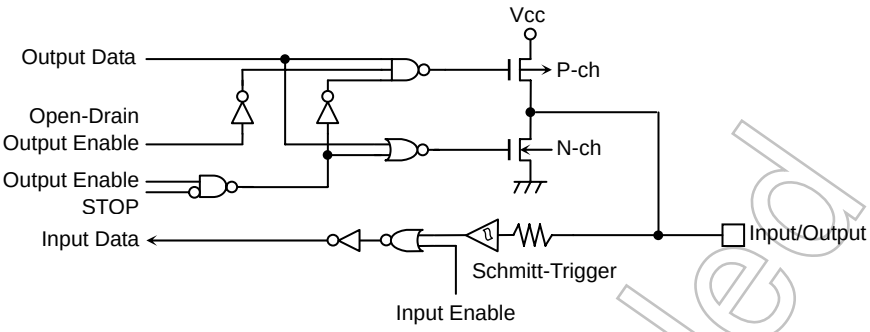
■ PC0, PC2, PC3,PC5, PC6, PD0, PD1, PD3, PD4, PD6, PE0, PE2, PN0, PN2



■ PD7, PE3–PE7, PK

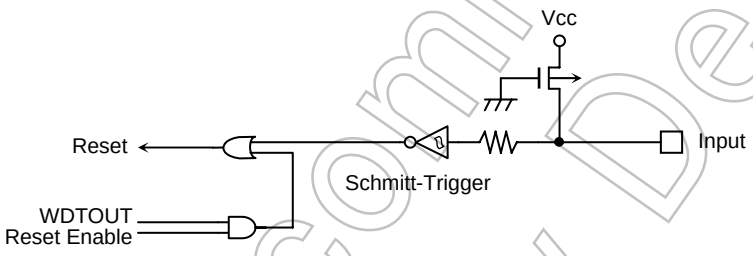


■ PF0, PF1

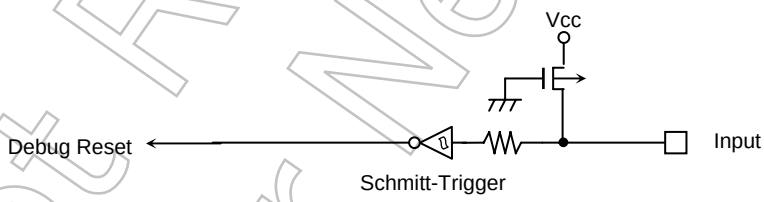


■ $\overline{\text{NMI}}$, BW0–BW1, $\overline{\text{PLLOFF}}$, RSTPUP

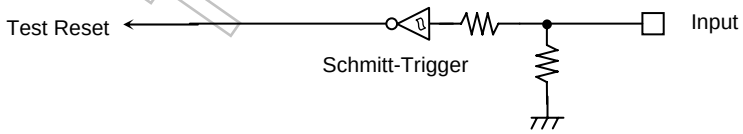
■ $\overline{\text{RESET}}$



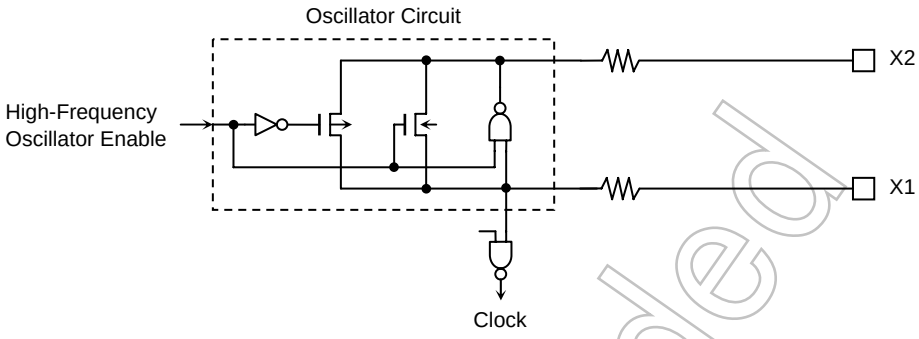
■ $\overline{\text{DRESET}}$, $\overline{\text{DBGE}}$, SDI/$\overline{\text{DINT}}$, TCK, TMS, TDI



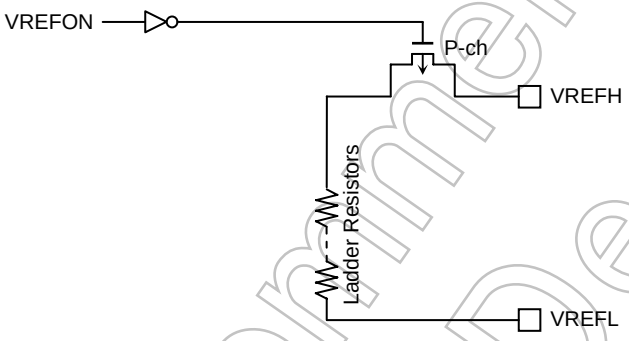
■ $\overline{\text{TRST}}$



■ X1, X2



■ VREFH, VREFL



25. Notations, Precautions and Restrictions

25.1 Notations and Terms

(1) I/O register fields are often referred to as <*register_mnemonic*>.<*field_name*> for the interest of brevity. For example, TRUN.T0RUN means the T0RUN bit in the TRUN register.

(2) fc, fsys, state

fosc: Clock supplied from the X1 and X2 pins

fpll: Clock generated by the on-chip PLL

fc: Clock selected by the $\overline{\text{PLLOFF}}$ pin

fgear: Clock selected by the SYSCR1.GEAR[1:0] bits

fsys: Clock selected by the SYSCR1.SYSCK bit

The fsys cycle is referred to as a state.

In addition, the clock selected by the SYSCR1.FPSEL bit and the prescaler clock source selected by the SYSCR0.PRCK[1:0] bits are referred to as fperiph and ϕT0 respectively.

25.2 Precautions and Restrictions

(1) Processor Revision Identifier

The Process Revision Identifier (PRId) register in the TX19 core of the TMP1962C10B contains 0x0000_2CA1.

(2) BW0–BW1 Pins

The BW0 and BW1 pins must be connected to the DVCC2 pin to ensure that their signal levels do not fluctuate during chip operation.

(3) Oscillator Warm-Up Counter

If an external crystal is utilized, an interrupt signal programmed to bring the TMP1940CYAF out of STOP mode triggers the on-chip warm-up counter. The system clock is not supplied to the on-chip logic until the warm-up counter expires.

(4) Programmable Pullup Resistors

When port pins are configured as input ports, the integrated pull-up resistors can be enabled and disabled under software control. The pull-up resistors are not programmable when port pins are configured as output ports.

The relevant port registers are programmed with the data resister.

(5) External Bus Mastership

The pin states while the bus is granted to an external device are described in Chapter 7, *I/O Ports*.

(6) Watchdog Timer (WDT)

Upon reset, the WDT is enabled. If the watchdog timer function is not required, it must be disabled after reset. When relevant pins are configured as bus arbitration signals, the I/O peripherals including the WDT can operate during external bus mastership.

(7) A/D Converter (ADC)

The ladder resistor network between the VREFH and VREFL pins can be disconnected under software control. This helps to reduce power dissipation, for example, in STOP mode.

(8) Undefined Bits in I/O Registers

Undefined I/O register bits are read as undefined states. Therefore, software must be coded without relying on the states of any undefined bits.

(9) Electrostatic Discharge (ESD) Sensitivity

The following shows ESD sensitivity. Protect the device from static damage during device development or production stage. For a detailed description on ESD, see General Safety Precautions and Usage Considerations.

- **TMP1962C10BXBG**

Specification	Sensitivity
Machine Model: MM	200 V
Human Body Model: HBM	1500 V

- **TMP1962F10AXBG**

Specification	Sensitivity
Machine Model: MM	200 V
Human Body Model: HBM	1200 V

(10) Notations, Precautions and Restrictions

Overflow Exception

Problem:

If an overflow exception caused a jump to the exception handler and the first instruction in that exception handler caused another exception, the EPC register should point to the address of the first instruction in the exception handler. However, the EPC register might contain the address that caused the overflow exception.

- **Problem-Causing Situation:**

  When, with the instruction pipeline full, an overflow exception was taken at the following sequence of instructions and then the first instruction in the overflow exception handler causes another exception

  ADD, ADDI or SUB <= # Instruction that causes an overflow

  Jump or branch instruction <= # Instruction with a delay slot

  Delay slot

Note: Toshiba's compiler uses no instructions that could cause an overflow. Therefore, this problem does not occur.

Workaround:

Don't place a jump or branch instruction immediately following an instruction that could cause an overflow (ADD, ADDI or SUB).

LWL and LWR Instructions

Problem:

The LWL or LWR instruction might provide incorrect results.

- **Problem-Causing Situation #1:**
  a. The destination of a load instruction (LB, LBU, LH, LHU, LW, LWL or LWR) is identical to that of the LWL or LWR instruction.
  b. The instruction pipeline is full. (The load instruction and the LWL or LWR instruction will be executed consecutively.)
  c. The DMAC is programmed for data cache snooping. Once the load instruction is executed, the DMAC initiates a DMA transaction. After it has been serviced, the LWL or LWR instruction is executed.

  This problem occurs when all of these conditions are true.

- **Problem-Causing Situation #2:**
  a. The destination of a load instruction (LB, LBU, LH, LHU, LW, LWL or LWR) is identical to that of the LWL or LWR instruction.
  b. The Doze or Halt bit in the Config register is set to 1 immediately before the load instruction.
  c. The instruction pipeline is full. (The load instruction and the LWL or LWR instruction will be executed consecutively.)
  d. After the load instruction is executed, the processor is put in the STOP, SLEEP or IDLE mode.
  e. After an interrupt signaling brings the processor out of the STOP, SLEEP or IDLE mode, the LWL or LWR instruction is executed.

Note: This applies to the case in which an interrupt signaling does not generate an interrupt upon exit from STOP or IDLE mode. In other words, either the IEc bit in the Status register is cleared (interrupts disabled), or if the IEc bit is set, the priority level of the incoming interrupt signaling is lower than the mask level programmed in the CMask field in the Status register. (Exit from STOP, SLEEP or IDLE mode can be accomplished even with such settings.)

This problem occurs when all of these conditions are true.

Workarounds:

To use the LWL or LWR instruction,

1) Place a NOP between a load instruction and the LWL or LWR instruction, or
2) Disable the data cache snooping of the DMAC before the LWL or LWR instruction is executed. Also, do not put the processor in STOP, SLEEP or IDLE mode before the LWL or LWR instruction is executed.

Overflow Exception When a DSU Probe Is Used

Problem:

It looks as if an overflow exception caused a jump to the reset and nonmaskable exception vector address (0xBFC0_0000).

- **Problem-Causing Situation:**
  When an overflow exception occurs, with the processor connected to a DSU probe

Note: Toshiba's compiler uses no instructions that could cause an overflow. Therefore, this problem does not occur.

Workaround:

Don't place a jump or branch instruction immediately following an instruction that could cause an overflow (ADD, ADDI or SUB).

Malfunction of using BUSREQ signl in External Bus Access mode

[Condition]

1. In External Bus mode, using Auto WAIT insert function (as same as +N wait)
2. Use External Bus request signal Function (BUSREQ).
3. For each target product,Bus setting mode (Multiplex/ separate)、ALE width(short/long)
   . Please refer to following table.

(Exp: ALE Band =1.5CLK, Auto wait = 3)