# 32-bit RISC Microcontroller

# TXZ Family

## Reference Manual
## 12-bit Analog to Digital Converter (ADC-B)

## Revision 2.0

**2018-03**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

# Contents

Preface ..... 5
Related document ..... 5
Conventions ..... 6
Terms and Abbreviations ..... 8
1. Outlines ..... 9
2. Block Diagram ..... 11
3. Function and Operation ..... 13
3.1. Clock Supply ..... 13
3.2. Conversion Operation by General Purpose Start up Factor ..... 14
3.2.1. Operation ..... 14
3.2.2. Control Registers ..... 15
3.3. Conversion Operation by PMD Trigger ..... 17
3.3.1. Operation ..... 17
3.3.2. Control Registers ..... 18
3.4. Conversion Stop ..... 19
3.5. Start up Priority ..... 20
3.6. AD Monitor Function ..... 21
3.7. Analog Reference Voltage ..... 23
3.8. Conversion Time ..... 24
3.8.1. Conversion timing ..... 24
3.9. Trigger Switching Control ..... 25
4. Registers ..... 26
4.1. List of Registers ..... 26
4.2. Details of Registers ..... 29
4.2.1. *[ADxCR0]* (Control Register0) ..... 29
4.2.2. *[ADxCR1]* (Control Register1) ..... 29
4.2.3. *[ADxST]* (Status Register) ..... 30
4.2.4. *[ADxCLK]* (Conversion Clock Setting Register) ..... 31
4.2.5. *[ADxMOD0]* (Mode Setting Register0) ..... 31
4.2.6. *[ADxMOD1]* (Mode Setting Register1) ..... 32
4.2.7. *[ADxMOD2]* (Mode Setting Register2) ..... 32
4.2.8. *[ADxCMPEN]* (Monitor function Enable Register) ..... 33
4.2.9. *[ADxCMPCR0]* (Monitor function Setting Register0) ..... 33
4.2.10. *[ADxCMPCR1]* (Monitor function Setting Register1) ..... 34
4.2.11. *[ADxCMP0]* (Conversion Result Comparison Register0) ..... 35
4.2.12. *[ADxCMP1]* (Conversion Result Comparison Register1) ..... 35
4.2.13. PMD Trigger Control Registers ..... 36
4.2.13.1. *[ADxPSEL0]* (PMD Trigger Program Number Selection Register0) ..... 36
4.2.13.2. *[ADxPINTS0]* (PMD Trigger Interrupt Selection Register0) ..... 36
4.2.13.3. *[ADxPREGS0]* (PMD Trigger Storage Selection Register0) ..... 37
4.2.13.4. *[ADxPREGS1]* (PMD Trigger Storage Selection Register1) ..... 38

4.2.13.5. *[ADxPSET0]* (PMD Trigger Program Register0) ..... 38
4.2.14. *[ADxTSET0]* (General Purpose Start up Factor Program Register0) ..... 40
4.2.15. *[ADxREG0]* (Conversion Result Storage Register0) ..... 41
4.2.16. *[ADxPROSEL]* (Program restart selection register) ..... 42
4.2.17. *[ADxPFLG]* (Startup priority status register) ..... 44
4.2.18. *[ADxPINT]* (Priority interrupt control register) ..... 45
4.2.19. *[SHxTRGPAT]* (Trigger switching pattern setting register) ..... 46
5. Usage example ..... 47
5.1. Single conversion ..... 47
5.2. PMD trigger conversion ..... 48
5.2.1. PMD (3-shunt), ADC × 1 ..... 48
5.2.2. PMD (1-shunt), ADC × 1 ..... 49
6. Precaution ..... 50
7. Revision History ..... 51
RESTRICTIONS ON PRODUCT USE ..... 53

# List of Figures

Figure 1.1 Related figure of ADC and another peripheral function ..... 10
Figure 2.1 ADC Block diagram ..... 11
Figure 3.1 General purpose start up factor and its corresponding operation ..... 14
Figure 3.2 PMD start up factor and its operation ..... 17
Figure 3.3 AD monitor function (Determination condition: Continuous count) ..... 22
Figure 3.4 AD monitor function (Determination condition: Accumulated count) ..... 22
Figure 3.5 Configuration of Analog reference voltage ..... 23
Figure 3.6 Example Conversion timing ..... 24
Figure 5.1 3-shunt example ..... 48
Figure 5.2 1-shunt example ..... 49

# List of Tables

Table 2.1 List of Signals ..... 12
Table 3.1 Start up factor and interrupt / DMA request ..... 15
Table 3.2 Operation when the start up factor occurs during the conversion ..... 20
Table 3.3 Monitor function and interrupt ..... 21
Table 3.4 Trigger combination patterns ..... 25
Table 5.1 ADC setting in 3-shunt ..... 48
Table 5.2 ADC setting in 1-shunt ..... 49
Table 7.1 Revision History ..... 51

# Preface

**Related document**

| Document name |
|---|
| Exception |
| Clock Control and Operation Mode |
| Product Information |
| Advanced Programmable Motor Control Circuit |
| Advanced Vector Engine Plus |

## Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> =0x01 (hexadecimal), ***[XYZn]***<VW> =1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value.
  In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

The Flash memory uses the Super Flash® technology under the license of Silicon Storage Technology, Inc.
Super Flash® is registered trademark of Silicon Storage Technology, Inc.

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

## Terms and Abbreviations

Some of abbreviations used in this document are as follows:

| | |
|---|---|
| ADC | Analog to Digital Converter |
| A-PMD | Advanced Programmable Motor Control Circuit |
| A-VE+ | Advanced Vector Engine Plus |
| TRGSEL | Trigger Selection circuit |

# 1. Outlines

The 12-bit analog to digital converter (ADC) can convert multiple analog inputs (AINx00 to AINxn) to digital in each unit. The function list is shown as follows.

| Function classification | Function | Operation explanation |
|---|---|---|
| AD conversion | Conversion resolution | 12bits |
| | Conversion time | 4.5 ≤ AVDD5 ≤ 5.5 [V]: 0.5, 0.62, 0.85[μs] at ADCLK=120[MHz]<br>2.7 ≤ AVDD5 < 4.5 [V]: 2.0[μs] at ADCLK=120[MHz] |
| | Store conversion result | 24 conversion result storage registers. |
| Start conversion | Start up by General Purpose Factor | Selectable start up factor: Software start up (Continuous conversion, Single conversion) and general purpose trigger.<br>There is a conversion program (Note) that can perform AD conversions up to 24 with general purpose factor. |
| | Start up by PMD trigger | Each of the twelve PMD triggers can select and execute one of nine PMD trigger programs (Note).<br>Each PMD trigger program can perform up to 4 AD conversions at each conversion program. |
| | Trigger switching control | The execution priority can be changed by switching 13 triggers (PMD trigger and general trigger).<br>Selectable Trigger combination: 16 patterns |
| Conversion status | Status flags | Flag showing that the AD conversion is executing.<br>Flag showing that the program is executing (for each trigger).<br>Conversion result storage flag (for each conversion result storage register).<br>Conversion result overrun flag (for each conversion result storage register). |
| Interrupt | - | PMD trigger program completion (4 interrupts).<br>General purpose trigger program completion.<br>Software single conversion program completion.<br>Software continuous conversion program completion.<br>Priority losing factor occurs (the program conversion is suspended).<br>Monitor function interrupt (2 interrupts). |
| Monitor conversion result | AD monitor function | Each ADC unit has 2 channels of monitor function.<br>Selectable conversion result storage register to be monitored.<br>Selectable detection method:<br>Whether the target register value is larger or smaller than the comparison register.<br>Selectable number of detections.<br>Continuous count and accumulated count can be selected. |

Note: Conversion program can specify conversion channel (analog input) and enable / disable of interrupt. There are multiple programs. Each is started with the start up factor / trigger.

Figure 1.1 shows the connection relationships with the peripheral functions that are linked with the ADC.

The AD conversion can be executed with the PMD trigger synchronized with the motor drive timing of the Advanced Programmable Motor Control Circuit (hereafter, abbreviated as PMD) and the general purpose timer trigger. Execution of the OVV protection in PMD and activation of general purpose timer are possible with the AD monitor function.

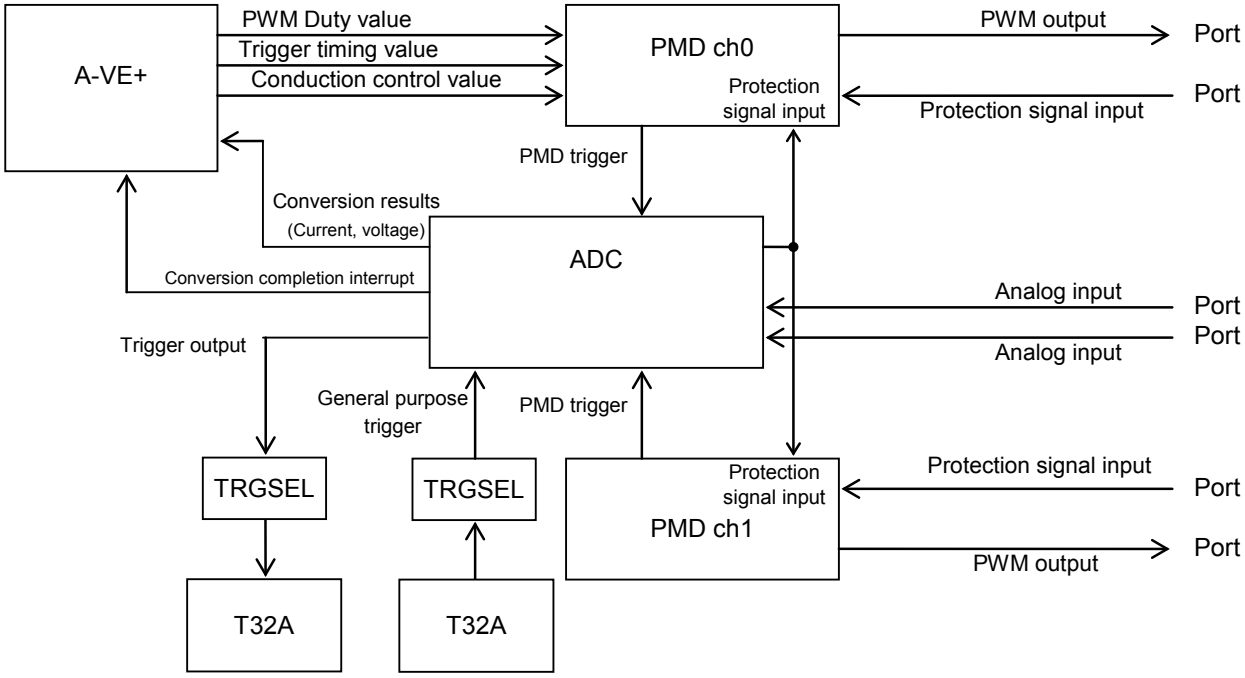



**Figure 1.1 Related figure of ADC and another peripheral function**

# 2. Block Diagram

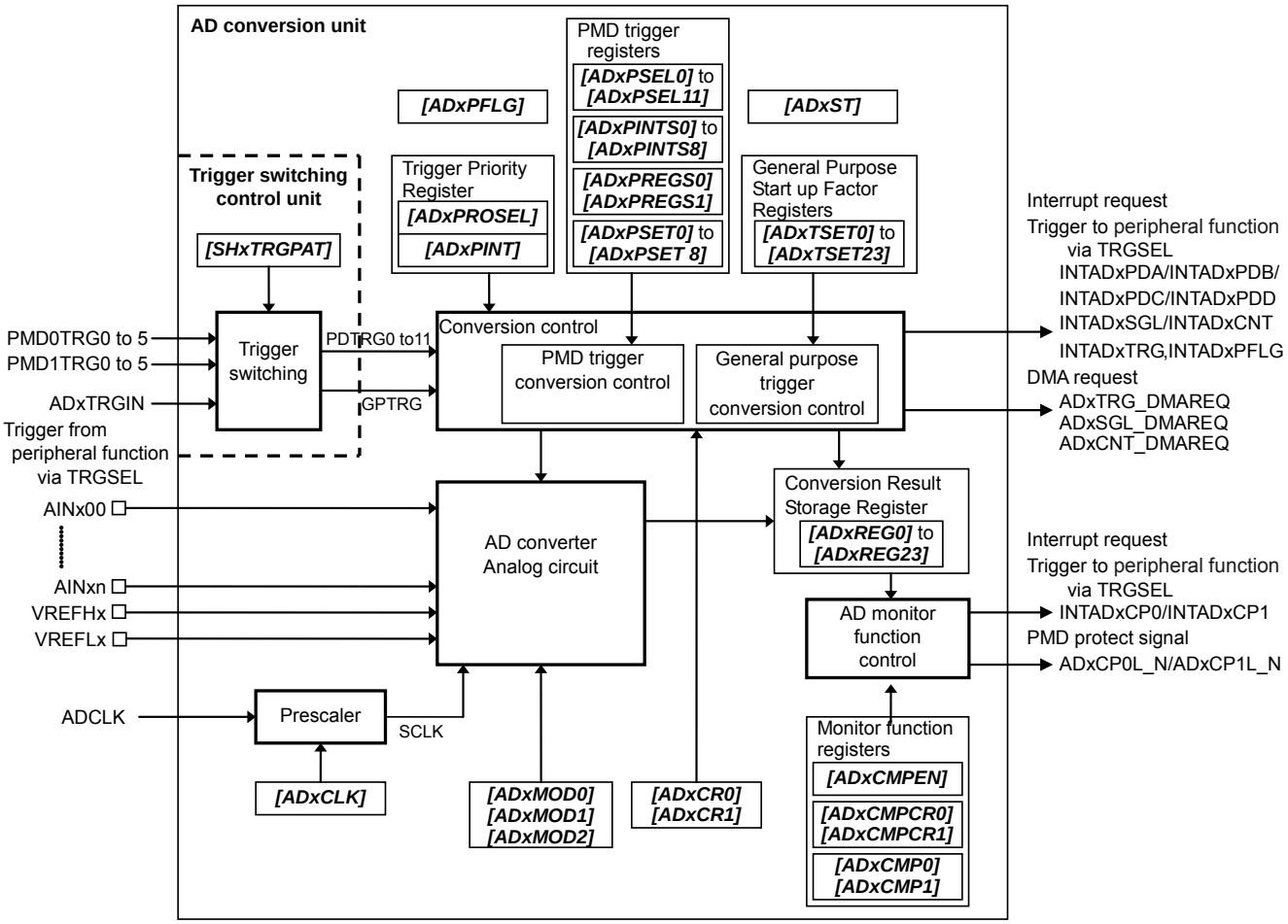



**Figure 2.1 ADC Block diagram**

**Table 2.1 List of Signals**

| No | Signal name | | I/O | Related Reference manual |
|---|---|---|---|---|
| 1 | ADCLK | Conversion clock for ADC | Input | Clock Control and Operation Mode |
| 2 | AINx00 to AINxn | Analog input pin | Input | Product Information |
| 3 | VREFHx | Reference power pin for analog | Input | Product Information |
| 4 | VREFLx | Reference GND pin for analog | Input | Product Information |
| 5 | PMD0TRG0 to 5 | PMD0 trigger | Input | Product Information |
| 6 | PMD1TRG0 to 5 | PMD1 trigger | Input | Product Information |
| 7 | ADxTRGIN | General purpose trigger | Input | Product Information |
| 8 | ADxCP0L_N | Monitor function0 output for PMD protect function | Output | Product Information |
| 9 | ADxCP1L_N | Monitor function1 output for PMD protect function | Output | Product Information |
| 10 | INTADxPDA | PMD trigger interrupt A | Output | Exception |
| 11 | INTADxPDB | PMD trigger interrupt B | Output | Exception |
| 12 | INTADxPDC | PMD trigger interrupt C | Output | Exception |
| 13 | INTADxPDD | PMD trigger interrupt D | Output | Exception |
| 14 | INTADxPFLG | Priority interrupt | Output | Exception |
| 15 | INTADxTRG | General purpose trigger interrupt | Output | Exception, Product Information |
| 16 | INTADxSGL | Single conversion interrupt | Output | Exception, Product Information |
| 17 | INTADxCNT | Continuous conversion interrupt | Output | Exception, Product Information |
| 18 | INTADxCP0 | Monitor function interrupt0 | Output | Exception, Product Information |
| 19 | INTADxCP1 | Monitor function interrupt1 | Output | Exception, Product Information |
| 20 | ADxTRG_DMAREQ | General purpose trigger DMA request | Output | Product Information |
| 21 | ADxSGL_DMAREQ | Single conversion DMA request | Output | Product Information |
| 22 | ADxCNT_DMAREQ | Continuous conversion DMA request | Output | Product Information |

# 3. Function and Operation

The ADC is triggered to start the conversion by the software start up (Software trigger) or the trigger signal from PMD, a timer, and others.

## 3.1. Clock Supply

When you use ADC, please supply the AD conversion unit clock / Trigger switching control unit clock / Conversion clock. Therefore, please set an applicable clock enable bit to "1" (clock supply) in Clock supply and stop register A or B for fsys (***[CGFSYSENA]***, ***[CGFSYSENB]***), Clock supply and stop register for fc (***[CGFCEN]***), and Clock supply and stop register for ADC and TRACE (***[CGSPCLKEN]***).

An applicable register and the bit position vary according to a product. Therefore, the register may not exist with the product. Please refer to "Clock Control and Operation Mode" of the reference manual for details.

When attempting to stop supplying the clock, make sure to check whether the AD conversion is stopping. Note that when the MCU enters STOP mode, make sure to check whether the AD conversion is stopping as well.

## 3.2. Conversion Operation by General Purpose Start up Factor

The factor of the general purpose start up is the GPTRG input or the software start up factor. The software starts up the single conversion or the continuous conversion.

### 3.2.1. Operation

When the conversion is triggered by the general purpose start up factor, the conversion executes according to the setting in the general start up factor program register which is prepared for each conversion result register.

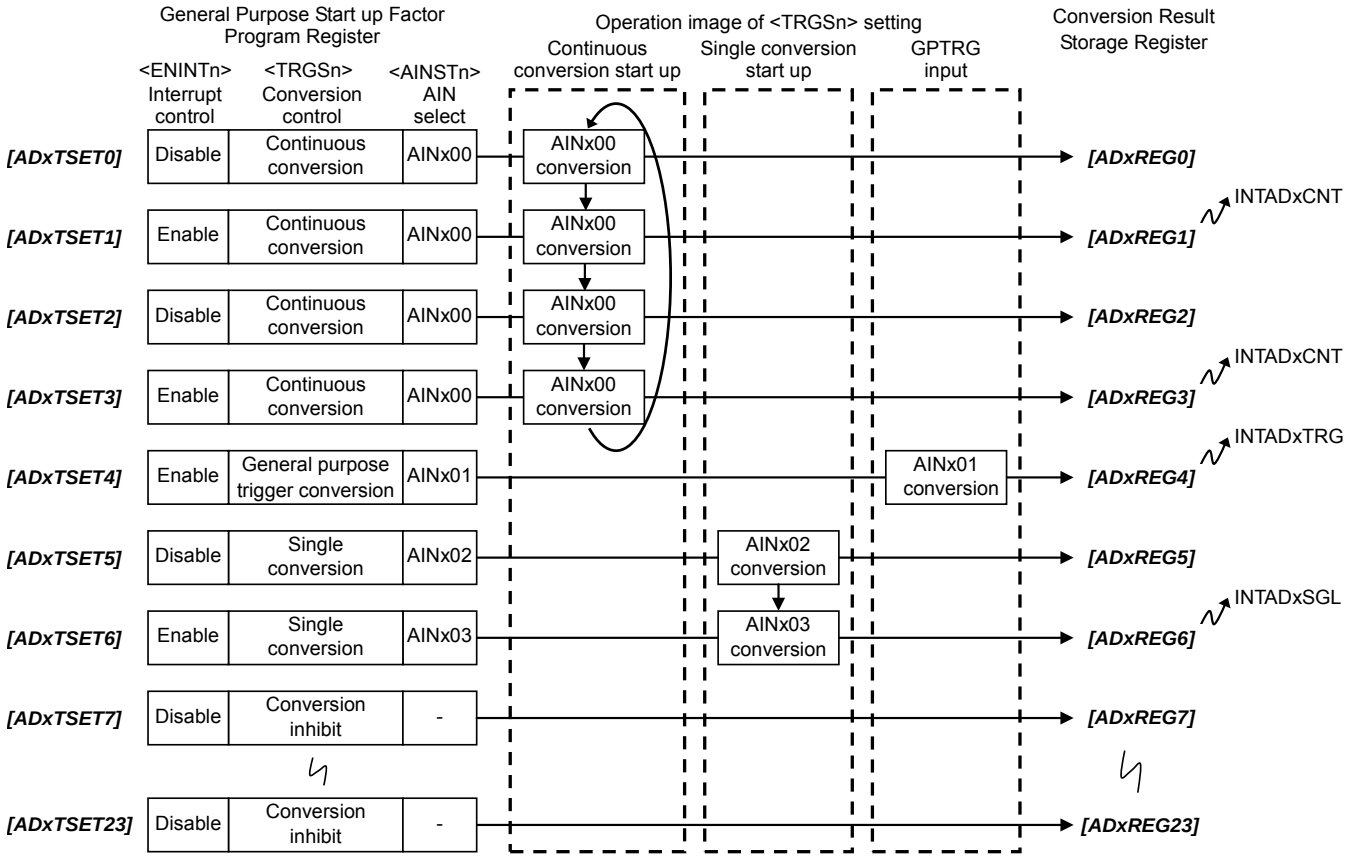



**Figure 3.1 General purpose start up factor and its corresponding operation**

The start up factor select (Conversion control)<TRGSn>, the AIN select <AINSTn>, and the interrupt enable or disable (Interrupt control)<ENINTn> are programmed to the general purpose start up factor program register (***[ADxTSETn]***). When the start up factor occurs, the specified conversions are executed from the smallest number of the register.

The continuous conversion repeats the specified conversion. The single conversion executes the specified conversion only once. The general purpose trigger conversion executes the specified conversion once when the general purpose trigger is received.

With a general purpose start up, when the conversion of the interrupt specified by ***[ADxTSETn]***<ENINTn> ends, different interrupts (INTADxTRG, INTADxSGL, INTADxCNT) are generated for each start up factor (general purpose trigger conversion, single conversion, continuous conversion).

A DMA request can be generated for each general purpose start up factor. When a DMA request control (***[ADxCR1]***<CNTDMEN><SGLDMEN><TRGDMEN>) of the start up factor that generates the interrupt is set to "1", a DMA request and an interrupt request are generated simultaneously.

**Table 3.1 Start up factor and interrupt / DMA request**

| Start up factor | Interrupt | DMA request |
|---|---|---|
| General purpose trigger conversion | General purpose trigger program completion (INTADxTRG) | General purpose trigger DMA request (ADxTRG_DMAREQ) |
| Single conversion operation (software) | Software single conversion program completion (INTADxSGL) | Single conversion DMA request (ADxSGL_DMAREQ) |
| Continuous conversion operation (software) | Software continuous conversion program completion (INTADxCNT) | Continuous conversion DMA request (ADxCNT_DMAREQ) |

### 3.2.2. Control Registers

- General purpose start up factor program register (***[ADxTSET0]*** to ***[ADxTSET23]***)
  The general purpose start up factor program register is prepared for each conversion result storage register. The AIN select <AINSTn>, the conversion control <TRGSn>, and the interrupt control <ENINTn> are set to ***[ADxTSETn]***.

- Mode setting register0 (***[ADxMOD0]***)
  When using the ADC, set "1" to ***[ADxMOD0]***<DACON>. And the interval of 3[μs] are necessary for the stabilization.

- Control register0 (***[ADxCR0]***)
  When the AD conversion can be started, after setting, ***[ADxCR0]***<ADEN> should be set to "1".
  The software single conversion or the software continuous conversion is enabled by setting ***[ADxCR0]*** <SGL> or <CNT> to "1", respectively. When the continuous conversion should be stopped, <CNT> is set to "0".

- Control register1 (***[ADxCR1]***)
  ***[ADxCR1]***<TRGEN> enables the trigger, and then the program start up is done by the general purpose trigger. The conversion starts when a trigger is received.
  ***[ADxCR1]***<SGLDMEN><CNTDMEN><TRGDMEN> are set to "1" to enable the DMA request generation.

Note: ***[ADxCR1]*** register must be set while ***[ADxCR0]***<ADEN>=0.

For start AD conversion by the general purpose start up factor, please set up as below sequence.

- Single conversion
  (1) Set interrupt to use INTADxSGL.
  (2) Set "1" to ***[ADxMOD0]***<DACON>.
  (3) Wait at least 3[μs].
  (4) Set ***[ADxTSETn]***. AIN selection <AINSTn>= arbitrary, conversion control <TRGSn>=10, interrupt control <ENINTn>=1.
  (5) To perform the Single conversion using multiple channels, change the AIN selection and set (4) again.
  (6) Set "1" to ***[ADxCR0]***<ADEN>.

(7) Set "1" to ***[ADxCR0]***<SGL>, starts the conversion.
(8) When conversion is complete, INTADxSGL will be generated. Read ***[ADxREGn]*** in the interrupt service routine.
(9) Repeat steps (7) to (8).

- Continuous conversion
  (1) Set interrupt to use INTADxCNT.
  (2) Set "1" to ***[ADxMOD0]***<DACON>.
  (3) Wait at least 3[μs].
  (4) Set ***[ADxTSETn]***. AIN selection <AINSTn>= arbitrary, conversion control <TRGSn>=01, interrupt control <ENINTn>=1.
  (5) To perform the continuous conversion using multiple channels, change the AIN selection and set (4) again.
  (6) Set "1" to ***[ADxCR0]***<ADEN>.
  (7) Set "1" to ***[ADxCR0]***<CNT>, starts the conversion.
  (8) When conversion is complete, INTADxCNT will be generated. Read ***[ADxREGn]*** in the interrupt service routine.
  (9) Repeat steps (8).

- General purpose trigger conversion
  (1) Set interrupt to use INTADxTRG.
  (2) Set "1" to ***[ADxMOD0]***<DACON>.
  (3) Wait at least 3[μs].
  (4) Set "1" to ***[ADxCR1]***<TRGEN>.
  (5) Sets which trigger to use for the general purpose trigger (ADxTRGIN). (Note)
  (6) Set ***[ADxTSETn]***. AIN selection <AINSTn>= arbitrary, conversion control <TRGSn>=11, interrupt control <ENINTn>=1.
  (7) To activate the general purpose trigger using multiple channels, change the AIN selection and set (6) again.
  (8) Set "1" to ***[ADxCR0]***<ADEN>.
  (9) When you input a trigger, conversion starts.
  (10) When conversion is complete, INTADxTRG will be generated. Read ***[ADxREGn]*** in the interrupt service routine.
  (11) Repeat steps (9) to (10).

Note: For details of the signal connected to the general purpose trigger (ADxTRGIN), refer to "Product Information" of the reference manual.

## 3.3. Conversion Operation by PMD Trigger

### 3.3.1. Operation

The conversion is started by the PDTRGn (n=0 to 11). PDTRGn are triggers from the PMD. (Note1) (Note2)

The programmed conversion operation is executed by the PDTRGn. Each PDTRGn selects one program from among 9 programs available.

One program can execute 4-time conversions at maximum. The conversion result is stored to the selected register group in units of 4 registers.

One interrupt from among the INTADxPDA, INTADxPDB, INTADxPDC, and INTADxPDD can be generated at the program completion.

Note1: For details of the PMD, refer to "Advanced Programmable Motor Control Circuit" of the reference manual.

Note2: For the connections of each product, refer to "Product Information" of the reference manual.

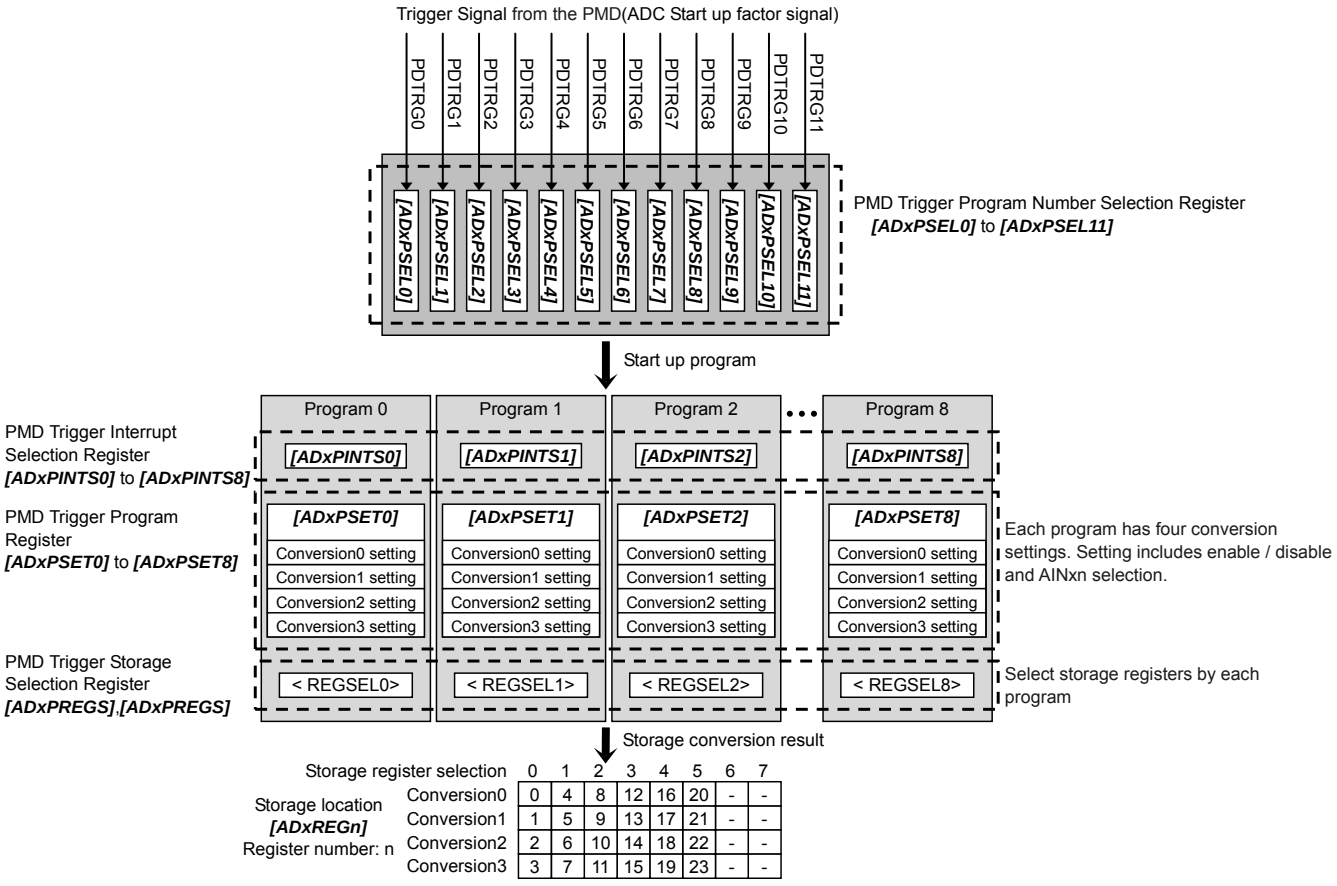



**Figure 3.2 PMD start up factor and its operation**

### 3.3.2. Control Registers

The following registers should be set for the conversion started by the PMD trigger.

- Mode setting register0 (***[ADxMOD0]***)
  When using the ADC, set "1" to ***[ADxMOD0]***<DACON>.And the interval of 3[μs] are necessary for the stabilization.

- PMD trigger program number selection register (***[ADxPSEL0]*** to ***[ADxPSEL11]***)
  Each register sets the trigger enable/disable and the number of the specified program (0 to 8) for one corresponding trigger out of 12 triggers.
  12 registers (***[ADxPSEL0]*** to ***[ADxPSEL11]***) are prepared for the 12 PMD triggers (PDTRG0 to PDTRG11), respectively.

- PMD trigger program register (***[ADxPSET0]*** to ***[ADxPSET8]***)
  This register sets the corresponding program enable or disable, the converted analog input channel and the Advanced Vector Engine Plus(hereafter, abbreviated as VE). Each program can be set to 4-time conversions at maximum.

- PMD trigger interrupt selection register (***[ADxPINTS0]*** to ***[ADxPINTS8]***)
  An interrupt can be generated at the program completion. The PMD trigger interrupt selection register selects the interrupt enable or disable, and the interrupt INTADxPDA, INTADxPDB, INTADxPDC, or INTADxPDD.
  9 registers (***[ADxPINTS0]*** to ***[ADxPINTS8]***) are prepared for the 9 programs, respectively.

- PMD trigger storage selection register (***[ADxPREGS0]***, ***[ADxPREGS1]***)
  The storage destination of the conversion result of each program can be selected. The storage destination is selected from among the group of the conversion result storage register 0 to 3, 4 to 7, 8 to 11, 12 to 15, 16 to 19, and 20 to 23.

  Note: When the VE is used, the conversion result storage register 0 to 3 should be used.

- Control register0 (***[ADxCR0]***)
  When the AD conversion can be started, after setting, ***[ADxCR0]***<ADEN> should be set to "1".

For start AD conversion by PMD trigger, please setup as below sequence.

(1) Set interrupt to use INTADxPDA, INTADxPDB, INTADxPDC, or INTADxPDD.
(2) Set "1" to ***[ADxMOD0]***<DACON>.
(3) Wait at least 3[μs].
(4) Set ***[ADxPSELn]***. Trigger control <PENSn>=1, Program Number <PMDSn>= arbitrary.
(5) Set ***[ADxPSETn]***. AIN selection <AINSPm>= arbitrary, Phase selection (for Vector Engine) <UVWISn> arbitrary, Conversion control<ENSPn>=1.
(6) Set ***[ADxPINTSn]***. Interrupt selection <INTSELn>= INTADxPDA, INTADxPDB, INTADxPDC, or INTADxPDD.
(7) Set ***[ADxPREGS0]*** or ***[ADxPREGS1]***. Register selection <REGSELn>= arbitrary.
(8) Set "1" to ***[ADxCR0]***<ADEN>.
(9) Conversion starts by the trigger (PDTRGn) that is PMD generated.
(10) When conversion program is complete, Interrupt (INTADxPDA, INTADxPDB, INTADxPDC, or INTADxPDD) will be generated. Read ***[ADxREGn]*** to ***[ADxREGn+3]*** in the interrupt service routine.
(11) Repeat steps (9) to (10).

## 3.4. Conversion Stop

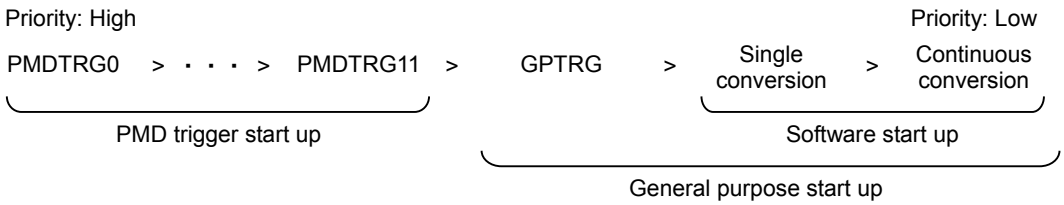
When ***[ADxCR0]***<ADEN> is set to "0", the conversion stops immediately. If the continuous conversion is enabled, ***[ADxCR0]***<CNT> should be also set to "0".

When the conversion stops completely, all bits in ***[ADxST]*** become "0". The registers other than ***[ADxST]*** keep their data, as well as the conversion result registers. Before the next conversion is enabled, the conversion result registers should be read to clear the corresponding flags.

When stopping ADCLK, AD conversion stop processing should be performed. Please confirm that ***[ADxST]***<ADBF> becomes "0" and stop ADCLK.

## 3.5. Start up Priority

The start up factors are prioritized as follows:




If multiple start up factors occur at the same time, the conversion program with the highest priority factor is executed and other factors are suspended.

If a higher priority factor is generated, the current conversion program execution is suspended and the conversion program of the higher priority factor executes. If a lower priority factor is generated, its conversion program execution is suspended.

A start up factor specified to be canceled with the program restart selection register ***[ADxPROSEL]*** is canceled, not aborted or reserved. Start up factors of losing priority are confirmed by the start up priority status register ***[ADxPFLG]***.

The priority interrupt control register ***[ADxPINT]*** can specify occurrence of a priority interrupt INTADxPFLG in case that priority is lost.

When the start up factor is generated again during execution of the conversion program of the same start up factor, the factor is ignored.

The execution status of the conversion program can be checked by ***[ADxST]***<CNTF><SNGF><TRGF><PMDF>. For the software start up factors, it should be confirmed whether the corresponding flags are "0". Then, the start up is certainly executed.

**Table 3.2 Operation when the start up factor occurs during the conversion**

| | | Later start up factor | | | |
|---|---|---|---|---|---|
| | | **PDTRGn (Note1)** | **General purpose trigger** | **Software Single conversion** | **Software Continuous conversion** |
| Current start up factor during conversion | PDTRGm (Note1) | Continue current factor (Note2) (Note6) | Continue current factor (Note3) (Note6) | Continue current factor (Note3) (Note6) | Continue current factor (Note3) (Note6) |
| | GPTRG | Start later factor (Note5) (Note6) | Continue current factor (Note4) | Continue current factor (Note3) (Note6) | Continue current factor (Note3) (Note6) |
| | Software Single conversion | Start later factor (Note5) (Note6) | Start later factor (Note5) (Note6) | Continue current factor (Note4) | Continue current factor (Note3) (Note6) |
| | Software Continuous conversion | Start later factor (Note5) (Note6) | Start later factor (Note5) (Note6) | Start later factor (Note5) (Note6) | Continue current factor (Note4) |

Note1: m,n = 0 to 11

Note2: In the case of m = n: The later start up factor is ignored.
In the case of m > n: The current start up factor is executed continuously, and the later start up factor is pending execution. The later start up factor will be started after the current start up factor is completed.
In the case of m < n: The later start up factor is started and the current start up factor is interrupted.

Note3: The later start up factor is performed after the current start up factor is completed.

Note4: The later start up factor is ignored.

Note5: The current start up factor is suspended, then the later start up factor is performed. The current start up factor is restarted after the later start up factor is completed.

Note6: If restart cancel setting (***[ADxPROSEL]***<PROSELn>=1) is specified, a suspended program is not restarted.

## 3.6. AD Monitor Function

The AD monitor function generates an interrupt if the AD conversion result is larger than the set value or smaller. It is possible to detect whether the AD conversion result is within the range of two set values or to detect whether the AD conversion result is out of the range by using this function simultaneously in two channels.

When ***[ADxCMPEN]***<CMP0EN> or <CMP1EN> is set to "1", the corresponding AD monitor function is enabled. The two monitor functions can be enabled simultaneously.

The following description is for ***[ADxCMPCR0]*** (The same for ***[ADxCMPCR1]***).

***[ADxCMPCR0]***<REGS0[4:0]> sets the conversion result storage register which value should be compared. <ADBIG0> sets the determination condition (bigger or smaller). <CMPCND0> sets the determination count condition. And <CMPCNT0[3:0]> sets the determination count value.

Whenever a conversion result is stored to the target conversion result storage register, the result is compared (bigger or smaller). If the comparison result is the same as the <ADBIG0> setting, the determination counter increments.

The determination count condition is either the continuous count or the accumulated count.

The continuous count condition is as follows: when the status set in <ADBIG0> continues the count times set in <CMPCNT0[3:0]>, the AD monitor function interrupt (INTADxCP0) and the protect signal for the PMD are generated. When it continues exceeding the set-up count number, nothing occurs. If the status is different from the <ADBIG0> status, the counter is cleared.

The accumulated count condition is as follows: when the count of the status set in <ADBIG0> is accumulated and the accumulated value reaches the value set in <CMPCNT0[3:0]>, the AD monitor function interrupt (INTADxCP0) and the protect signal for the PMD are generated, and the counter is cleared. Even when the status is different from the status set in <ADBIG0>, the counter value is maintained. When the value in the conversion result storage register specified by the ***[ADxCMPCR0]*** register is equal to the value in the conversion result comparison register, the counter does not increment and the AD monitor function interrupt and the trigger are not generated.

**Table 3.3 Monitor function and interrupt**

| Monitor function | Interrupt |
|---|---|
| Monitor function Setting Register0 (***[ADxCMPCR0]***) | Monitor function 0 Interrupt (INTADxCP0) |
| Monitor function Setting Register1 (***[ADxCMPCR1]***) | Monitor function 1 Interrupt (INTADxCP1) |

When the AD monitor function is used, the overrun flag ***[ADxREGn]***<ADOVRFn> and the conversion result storage flag ***[ADxREGn]***<ADRFn> are set because the storage register is not read by the software. So, when the AD monitor function is executing, the flags of the corresponding conversion result storage registers should not be used.

Note: The monitor function registers must be set while ***[ADxCR0]***<ADEN>=0.

(1) Determination by Continuous count

- Monitor function setting register0 (***[ADxCMPCR0]*** =0x00000200)
  Conversion result storage register: ***[ADxREG0]***
  Magnitude determination: ***[ADxREG0]***<ADR0> > ***[ADxCMP0]***<AD0CMP0>
  (Larger than the comparison register.)
  Determination count condition: Continuous count
  Magnitude determination count: 3 counts
- AD conversion result comparison register (***[ADxCMP0]***<AD0CMP0>=0x888)
- Monitor function enable register (***[ADxCMPEN]*** =0x00000001)

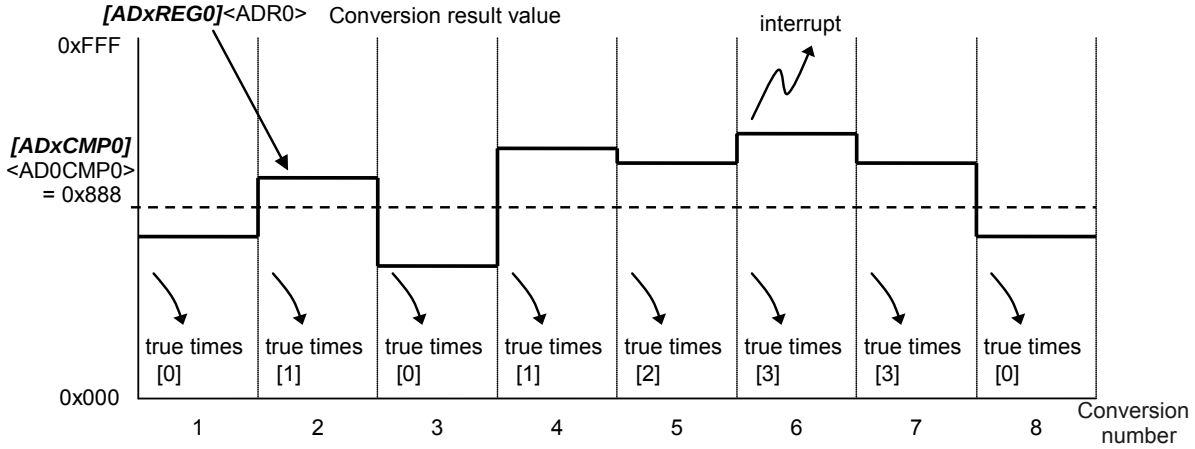



**Figure 3.3 AD monitor function (Determination condition: Continuous count)**

(2) Determination by Accumulated count

- Monitor function setting register (***[ADxCMPCR0]*** =0x00000240)
  Conversion result storage register: ***[ADxREG0]***
  Magnitude determination: ***[ADxREG0]***<ADR0> > ***[ADxCMP0]***<AD0CMP0>
  (Larger than the comparison register.)
  Determination count condition: Accumulated count
  Magnitude determination count: 3 counts
- AD conversion result comparison register (***[ADxCMP0]***<AD0CMP0>=0x888)
- Monitor function enable register (***[ADxCMPEN]*** =0x00000001)

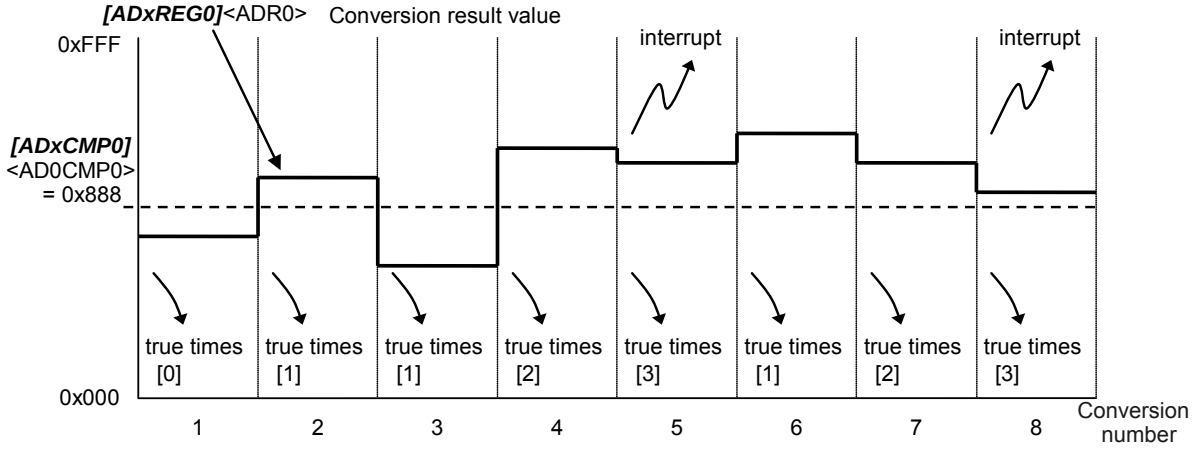



**Figure 3.4 AD monitor function (Determination condition: Accumulated count)**

## 3.7. Analog Reference Voltage

Analog reference pins VREFHx and VREFLx in the ADC unit are connected to a High level and a Low level, respectively. When ***[ADxMOD0]***<RCUT> is set to "1", the switch between VREFHx and VREFLx is turned on only during the conversion to reduce the power consumption.

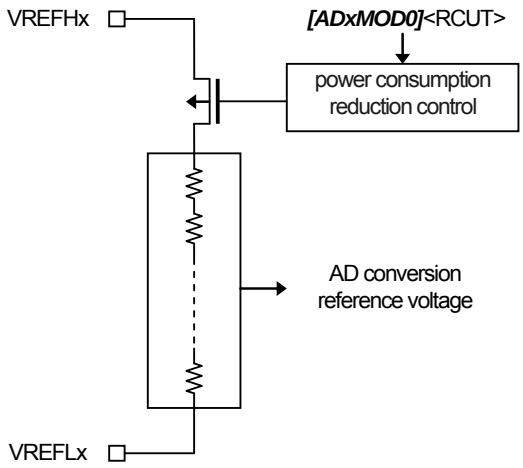



**Figure 3.5 Configuration of Analog reference voltage**

## 3.8. Conversion Time

### 3.8.1. Conversion timing

Figure 3.6 shows the conversion timing.

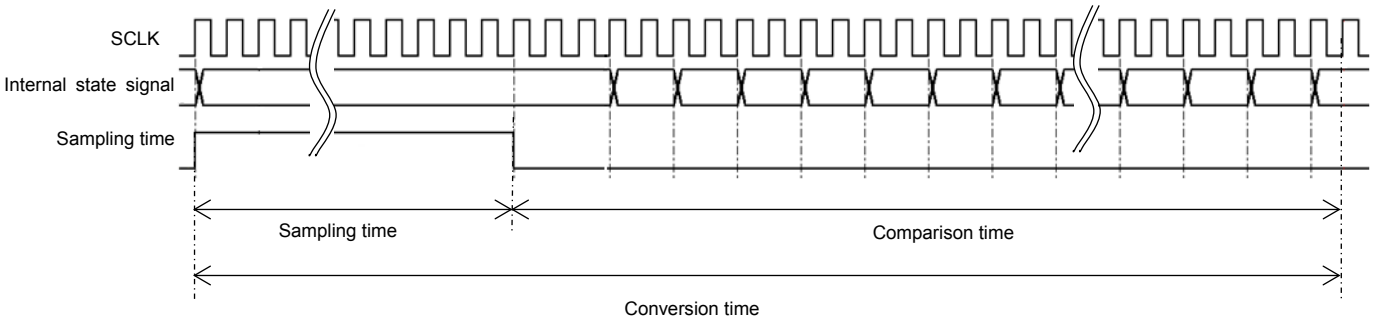



**Figure 3.6 Example Conversion timing**

## 3.9. Trigger Switching Control

The trigger control block outputs trigger signals to the ADC in different combinations. The combination is selected with the ***[SHxTRGPAT]*** register. (Refer to "Table 3.4 Trigger combination patterns".)

**Table 3.4 Trigger combination patterns**

| Trigger switching control output | <TRGPAT1> | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 00 | | | | 01 | | | | 10 | | | | 11 | | | |
| | <TRGPAT0> | | | | | | | | | | | | | | | |
| | 00 | 01 | 10 | 11 | 00 | 01 | 10 | 11 | 00 | 01 | 10 | 11 | 00 | 01 | 10 | 11 |
| PDTRG0 | T00 | | T10 | | T00 | | T10 | | T00 | | T10 | | T00 | | T10 | |
| PDTRG1 | T01 | | T11 | | T01 | | T11 | | T01 | | T11 | | T01 | | T11 | |
| PDTRG2 | T02 | | T12 | | T02 | | T12 | | T02 | | T12 | | T02 | | T12 | |
| PDTRG3 | T03 | T10 | T13 | T00 | T03 | T10 | T13 | T00 | T03 | T10 | T13 | T00 | HT | | | |
| PDTRG4 | T04 | T11 | T14 | T01 | T04 | T11 | T14 | T01 | T04 | T11 | T14 | T01 | T03 | T10 | T13 | T00 |
| PDTRG5 | T05 | T12 | T15 | T02 | T05 | T12 | T15 | T02 | T05 | T12 | T15 | T02 | T04 | T11 | T14 | T01 |
| PDTRG6 | T10 | T03 | T00 | T13 | T10 | T03 | T00 | T13 | HT | | | | T05 | T12 | T15 | T02 |
| PDTRG7 | T11 | T04 | T01 | T14 | T11 | T04 | T01 | T14 | T10 | T03 | T00 | T13 | T10 | T03 | T00 | T13 |
| PDTRG8 | T12 | T05 | T02 | T15 | T12 | T05 | T02 | T15 | T11 | T04 | T01 | T14 | T11 | T04 | T01 | T14 |
| PDTRG9 | T13 | | T03 | | HT | | | | T12 | T05 | T02 | T15 | T12 | T05 | T02 | T15 |
| PDTRG10 | T14 | | T04 | | T13 | | T03 | | T13 | | T03 | | T13 | | T03 | |
| PDTRG11 | T15 | | T05 | | T14 | | T04 | | T14 | | T04 | | T14 | | T04 | |
| GPTRG | HT | | | | T15 | | T05 | | T15 | | T05 | | T15 | | T05 | |

Note1: T0x,T1x, and HT in the table indicate trigger inputs described as follows:
T00 to T05: PMD0TRG 0 to 5
T10 to T15: PMD1TRG 0 to 5
HT: ADxTRGIN

Note2: Output: Trigger names (used to specify the register) that are used in trigger switching unit/ AD conversion unit after a trigger combination has been changed.

Note3: Interlocking operations of the ADC and VE can only be used in <TRGPAT1>=00.

# 4. Registers

## 4.1. List of Registers

The control registers and their addresses are shown as follows.

| Function | | Channel/Unit | Base address |
|---|---|---|---|
| 12-bit Analog to Digital Converter | ADC | Unit A | 0x400B8800 |
| Sample and hold circuit | SH | Unit A | 0x400B8700 |

- ADC registers

| Register Name | | Address (Base+) |
|---|---|---|
| Control Register0 | ***[ADxCR0]*** | 0x0000 |
| Control Register1 | ***[ADxCR1]*** | 0x0004 |
| Status Register | ***[ADxST]*** | 0x0008 |
| Conversion Clock Setting Register | ***[ADxCLK]*** | 0x000C |
| Mode Setting Register0 | ***[ADxMOD0]*** | 0x0010 |
| Mode Setting Register1 | ***[ADxMOD1]*** | 0x0014 |
| Mode Setting Register2 | ***[ADxMOD2]*** | 0x0018 |
| Monitor function Enable Register | ***[ADxCMPEN]*** | 0x0020 |
| Monitor function Setting Register0 | ***[ADxCMPCR0]*** | 0x0024 |
| Monitor function Setting Register1 | ***[ADxCMPCR1]*** | 0x0028 |
| Conversion Result Comparison Register0 | ***[ADxCMP0]*** | 0x002C |
| Conversion Result Comparison Register1 | ***[ADxCMP1]*** | 0x0030 |
| PMD Trigger Program Number Selection Register0 | ***[ADxPSEL0]*** | 0x0040 |
| PMD Trigger Program Number Selection Register1 | ***[ADxPSEL1]*** | 0x0044 |
| PMD Trigger Program Number Selection Register2 | ***[ADxPSEL2]*** | 0x0048 |
| PMD Trigger Program Number Selection Register3 | ***[ADxPSEL3]*** | 0x004C |
| PMD Trigger Program Number Selection Register4 | ***[ADxPSEL4]*** | 0x0050 |
| PMD Trigger Program Number Selection Register5 | ***[ADxPSEL5]*** | 0x0054 |
| PMD Trigger Program Number Selection Register6 | ***[ADxPSEL6]*** | 0x0058 |
| PMD Trigger Program Number Selection Register7 | ***[ADxPSEL7]*** | 0x005C |
| PMD Trigger Program Number Selection Register8 | ***[ADxPSEL8]*** | 0x0060 |
| PMD Trigger Program Number Selection Register9 | ***[ADxPSEL9]*** | 0x0064 |
| PMD Trigger Program Number Selection Register10 | ***[ADxPSEL10]*** | 0x0068 |
| PMD Trigger Program Number Selection Register11 | ***[ADxPSEL11]*** | 0x006C |
| PMD Trigger Interrupt Selection Register0 | ***[ADxPINTS0]*** | 0x0070 |
| PMD Trigger Interrupt Selection Register1 | ***[ADxPINTS1]*** | 0x0074 |
| PMD Trigger Interrupt Selection Register2 | ***[ADxPINTS2]*** | 0x0078 |
| PMD Trigger Interrupt Selection Register3 | ***[ADxPINTS3]*** | 0x007C |
| PMD Trigger Interrupt Selection Register4 | ***[ADxPINTS4]*** | 0x0080 |
| PMD Trigger Interrupt Selection Register5 | ***[ADxPINTS5]*** | 0x0084 |
| PMD Trigger Interrupt Selection Register6 | ***[ADxPINTS6]*** | 0x0088 |
| PMD Trigger Interrupt Selection Register7 | ***[ADxPINTS7]*** | 0x008C |
| PMD Trigger Interrupt Selection Register8 | ***[ADxPINTS8]*** | 0x0090 |
| PMD Trigger Storage Selection Register0 | ***[ADxPREGS0]*** | 0x0094 |
| PMD Trigger Storage Selection Register1 | ***[ADxPREGS1]*** | 0x0098 |
| PMD Trigger Program Register0 | ***[ADxPSET0]*** | 0x00A0 |
| PMD Trigger Program Register1 | ***[ADxPSET1]*** | 0x00A4 |
| PMD Trigger Program Register2 | ***[ADxPSET2]*** | 0x00A8 |
| PMD Trigger Program Register3 | ***[ADxPSET3]*** | 0x00AC |
| PMD Trigger Program Register4 | ***[ADxPSET4]*** | 0x00B0 |
| PMD Trigger Program Register5 | ***[ADxPSET5]*** | 0x00B4 |

| Register Name | | Address (Base+) |
|---|---|---|
| PMD Trigger Program Register6 | ***[ADxPSET6]*** | 0x00B8 |
| PMD Trigger Program Register7 | ***[ADxPSET7]*** | 0x00BC |
| PMD Trigger Program Register8 | ***[ADxPSET8]*** | 0x00C0 |
| General Purpose Start up Factor Program Register0 | ***[ADxTSET0]*** | 0x00C4 |
| General Purpose Start up Factor Program Register1 | ***[ADxTSET1]*** | 0x00C8 |
| General Purpose Start up Factor Program Register2 | ***[ADxTSET2]*** | 0x00CC |
| General Purpose Start up Factor Program Register3 | ***[ADxTSET3]*** | 0x00D0 |
| General Purpose Start up Factor Program Register4 | ***[ADxTSET4]*** | 0x00D4 |
| General Purpose Start up Factor Program Register5 | ***[ADxTSET5]*** | 0x00D8 |
| General Purpose Start up Factor Program Register6 | ***[ADxTSET6]*** | 0x00DC |
| General Purpose Start up Factor Program Register7 | ***[ADxTSET7]*** | 0x00E0 |
| General Purpose Start up Factor Program Register8 | ***[ADxTSET8]*** | 0x00E4 |
| General Purpose Start up Factor Program Register9 | ***[ADxTSET9]*** | 0x00E8 |
| General Purpose Start up Factor Program Register10 | ***[ADxTSET10]*** | 0x00EC |
| General Purpose Start up Factor Program Register11 | ***[ADxTSET11]*** | 0x00F0 |
| General Purpose Start up Factor Program Register12 | ***[ADxTSET12]*** | 0x00F4 |
| General Purpose Start up Factor Program Register13 | ***[ADxTSET13]*** | 0x00F8 |
| General Purpose Start up Factor Program Register14 | ***[ADxTSET14]*** | 0x00FC |
| General Purpose Start up Factor Program Register15 | ***[ADxTSET15]*** | 0x0100 |
| General Purpose Start up Factor Program Register16 | ***[ADxTSET16]*** | 0x0104 |
| General Purpose Start up Factor Program Register17 | ***[ADxTSET17]*** | 0x0108 |
| General Purpose Start up Factor Program Register18 | ***[ADxTSET18]*** | 0x010C |
| General Purpose Start up Factor Program Register19 | ***[ADxTSET19]*** | 0x0110 |
| General Purpose Start up Factor Program Register20 | ***[ADxTSET20]*** | 0x0114 |
| General Purpose Start up Factor Program Register21 | ***[ADxTSET21]*** | 0x0118 |
| General Purpose Start up Factor Program Register22 | ***[ADxTSET22]*** | 0x011C |
| General Purpose Start up Factor Program Register23 | ***[ADxTSET23]*** | 0x0120 |
| Conversion Result Storage Register0 | ***[ADxREG0]*** | 0x0144 |
| Conversion Result Storage Register1 | ***[ADxREG1]*** | 0x0148 |
| Conversion Result Storage Register2 | ***[ADxREG2]*** | 0x014C |
| Conversion Result Storage Register3 | ***[ADxREG3]*** | 0x0150 |
| Conversion Result Storage Register4 | ***[ADxREG4]*** | 0x0154 |
| Conversion Result Storage Register5 | ***[ADxREG5]*** | 0x0158 |
| Conversion Result Storage Register6 | ***[ADxREG6]*** | 0x015C |
| Conversion Result Storage Register7 | ***[ADxREG7]*** | 0x0160 |
| Conversion Result Storage Register8 | ***[ADxREG8]*** | 0x0164 |
| Conversion Result Storage Register9 | ***[ADxREG9]*** | 0x0168 |
| Conversion Result Storage Register10 | ***[ADxREG10]*** | 0x016C |
| Conversion Result Storage Register11 | ***[ADxREG11]*** | 0x0170 |
| Conversion Result Storage Register12 | ***[ADxREG12]*** | 0x0174 |
| Conversion Result Storage Register13 | ***[ADxREG13]*** | 0x0178 |
| Conversion Result Storage Register14 | ***[ADxREG14]*** | 0x017C |
| Conversion Result Storage Register15 | ***[ADxREG15]*** | 0x0180 |
| Conversion Result Storage Register16 | ***[ADxREG16]*** | 0x0184 |
| Conversion Result Storage Register17 | ***[ADxREG17]*** | 0x0188 |
| Conversion Result Storage Register18 | ***[ADxREG18]*** | 0x018C |
| Conversion Result Storage Register19 | ***[ADxREG19]*** | 0x0190 |
| Conversion Result Storage Register20 | ***[ADxREG20]*** | 0x0194 |
| Conversion Result Storage Register21 | ***[ADxREG21]*** | 0x0198 |
| Conversion Result Storage Register22 | ***[ADxREG22]*** | 0x019C |
| Conversion Result Storage Register23 | ***[ADxREG23]*** | 0x01A0 |
| Program restart selection register | ***[ADxPROSEL]*** | 0x01C4 |
| Startup priority status register | ***[ADxPFLG]*** | 0x01C8 |
| Priority interrupt control register | ***[ADxPINT]*** | 0x01CC |

- SH registers

| Register Name | | Address (Base+) |
|---|---|---|
| Trigger switching patter setting register | ***[SHxTRGPAT]*** | 0x000C |

## 4.2. Details of Registers

### 4.2.1. *[ADxCR0]* (Control Register0)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7 | ADEN | 0 | R/W | ADC control.<br>0: Disabled.<br>1: Enabled.<br>When "1" is set, the conversion is enabled. When "0" is set, the conversion stops. |
| 6:2 | - | 0 | R | Read as "0". |
| 1 | SGL | 0 | W | Single conversion control<br>0: Don't care<br>1: Conversion start.<br>When "1" is set, the single conversion program starts to execute.<br>If this bit is read, "0" is returned. |
| 0 | CNT | 0 | R/W | Continuous conversion control<br>0: Disabled.<br>1: Enabled.<br>When "1" is set, the continuous conversion starts to execute. This bit should be set to "1" when ***[ADxST]***<CNTF> is "0" (a continuous conversion program does not execute). |

### 4.2.2. *[ADxCR1]* (Control Register1)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:7 | - | 0 | R | Read as "0". |
| 6 | CNTDMEN | 0 | R/W | Continuous conversion DMA request control<br>0: Disabled.<br>1: Enabled. |
| 5 | SGLDMEN | 0 | R/W | Single conversion DMA request control<br>0: Disabled.<br>1: Enabled. |
| 4 | TRGDMEN | 0 | R/W | General purpose trigger DMA request control<br>0: Disabled.<br>1: Enabled. |
| 3:1 | - | 0 | R | Read as "0". |
| 0 | TRGEN | 0 | R/W | General purpose trigger start up control<br>0: Disabled.<br>1: Enabled. |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.3. *[ADxST]* (Status Register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7 | ADBF | 0 | R | AD operation flag<br>0: Stop (ADCLK can be stopped.)<br>1: Executing (ADCLK cannot be stopped.)<br>Before ADCLK is stopped, this bit should be confirmed to be "0". |
| 6:4 | - | 0 | R | Read as "0". |
| 3 | CNTF | 0 | R | Continuous conversion program flag<br>0: Stop<br>1: Executing<br>When the request is received, this bit becomes "1". When the last conversion result is stored, this bit becomes "0". |
| 2 | SNGF | 0 | R | Single conversion program flag<br>0: Stop<br>1: Executing<br>When the request is received, this bit becomes "1". When the last conversion result is stored, this bit becomes "0". |
| 1 | TRGF | 0 | R | General purpose trigger program flag<br>0: Stop<br>1: Executing<br>When the request is received, this bit becomes "1". When the last conversion result is stored, this bit becomes "0". |
| 0 | PMDF | 0 | R | PMD trigger program flag<br>0: Stop<br>1: Executing<br>When the request is received, this bit becomes "1". When the last conversion result is stored, this bit becomes "0". |

### 4.2.4. *[ADxCLK]* (Conversion Clock Setting Register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:7 | - | 0 | R | Read as "0". |
| 6:3 | EXAZ[3:0] | 0000 | R/W | AIN sampling time selection<br>Write as "0000". |
| 2:0 | VADCLK[2:0] | 000 | R/W | AD prescaler output (SCLK) selection<br>4.5 ≤ AVDD5 ≤ 5.5 [V]:<br>Write as "000" (ADCLK/1).<br>2.7 ≤ AVDD5 < 4.5 [V]:<br>Write as "010" (ADCLK/4). |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.5. *[ADxMOD0]* (Mode Setting Register0)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:4 | - | 0 | R | Read as "0". |
| 3 | AZFSH | 0 | R/W | Clock mode control<br>4.5 ≤ AVDD5 ≤ 5.5 [V]:<br>AD conversion time 0.5[μs] at SCLK=120M[Hz]:<br>Write as "0".<br>AD conversion time 0.62[μs] / 0.85[μs] at SCLK=120M[Hz]:<br>Write as "1".<br>2.7 ≤ AVDD5 < 4.5 [V]:<br>AD conversion time 2.0[μs] at SCLK=30M[Hz]:<br>Write as "0". |
| 2 | REFBSEL | 0 | R/W | VREFHx / AINxn switching control<br>0: Used as VREFHx pin (VREFHx pin is independent from AVDD5 pin)<br>1: Used as AINxn pin (Shared with AVDD5 and VREFHx pins; or AINxn pin)<br>This pin is shared with VREFHx and AINxn pins. This setting is selectable. |
| 1 | RCUT | 1 | R/W | Low power mode selection<br>0: Normal operation<br>1: Low power operation (Energized between VREFHx and VREFLx only during the conversion) |
| 0 | DACON | 0 | R/W | DAC control<br>0: OFF<br>1: ON<br>When the ADC is used, <DACON> should be set to "1". |

Note1: This register must be set while ***[ADxCR0]***<ADEN>=0.

Note2: After ***[ADxMOD0]***<DACON> is set to "1", the interval of 3[μs] are necessary for the stabilization.

### 4.2.6. *[ADxMOD1]* (Mode Setting Register1)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:0 | MOD1[31:0] | 0x00304000 | R/W | 4.5 ≤ AVDD5 ≤ 5.5 [V]:<br>AD conversion time 0.5[μs] at SCLK=120M[Hz]:<br>Write as "0x00304000".<br>AD conversion time 0.62[μs] at SCLK=120M[Hz]:<br>Write as "0x00304001".<br>AD conversion time 0.85[μs] at SCLK=120M[Hz]:<br>Write as "0x00304111".<br><br>2.7 ≤ AVDD5 < 4.5 [V]:<br>AD conversion time 2.0[μs] at SCLK=30M[Hz]:<br>Write as "0x00304000". |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.7. *[ADxMOD2]* (Mode Setting Register2)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:0 | MOD2[31:0] | 0x00000000 | R/W | 4.5 ≤ AVDD5 ≤5.5 [V]:<br>AD conversion time 0.5[μs] at SCLK=120M[Hz]:<br>Write as "0x00000000".<br>AD conversion time 0.62[μs] at SCLK=120M[Hz]:<br>Write as "0x00000060".<br>AD conversion time 0.85[μs] at SCLK=120M[Hz]:<br>Write as "0x00000000".<br><br>2.7 ≤ AVDD5 < 4.5 [V]:<br>AD conversion time 2.0[μs] at SCLK=30M[Hz]:<br>Write as "0x00000070". |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.8. *[ADxCMPEN]* (Monitor function Enable Register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:2 | - | 0 | R | Read as "0". |
| 1 | CMP1EN | 0 | R/W | AD monitor function1<br>0: Disabled.<br>1: Enabled. |
| 0 | CMP0EN | 0 | R/W | AD monitor function0<br>0: Disabled.<br>1: Enabled. |

### 4.2.9. *[ADxCMPCR0]* (Monitor function Setting Register0)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:12 | - | 0 | R | Read as "0". |
| 11:8 | CMPCNT0[3:0] | 0000 | R/W | Comparison count<br>0000: 1 1000: 9<br>0001: 2 1001: 10<br>0010: 3 1010: 11<br>0011: 4 1011: 12<br>0100: 5 1100: 13<br>0101: 6 1101: 14<br>0110: 7 1110: 15<br>0111: 8 1111: 16 |
| 7 | - | 0 | R | Read as "0". |
| 6 | CMPCND0 | 0 | R/W | Determination condition<br>0: Continuous count<br>1: Accumulated count |
| 5 | ADBIG0 | 0 | R/W | Magnitude determination setting<br>0: Conversion result specified by <REGS0> > ***[ADxCMP0]*** (Larger than the comparison register)<br>1: Conversion result specified by <REGS0> < ***[ADxCMP0]*** (Smaller than the comparison register) |
| 4:0 | REGS0[4:0] | 00000 | R/W | Compared conversion result storage register<br>00000: ADxREG0 01000: ADxREG8 10000: ADxREG16<br>00001: ADxREG1 01001: ADxREG9 10001: ADxREG17<br>00010: ADxREG2 01010: ADxREG10 10010: ADxREG18<br>00011: ADxREG3 01011: ADxREG11 10011: ADxREG19<br>00100: ADxREG4 01100: ADxREG12 10100: ADxREG20<br>00101: ADxREG5 01101: ADxREG13 10101: ADxREG21<br>00110: ADxREG6 01110: ADxREG14 10110: ADxREG22<br>00111: ADxREG7 01111: ADxREG15 10111: ADxREG23<br>11000 to 11111: Inhibited setting. |

Note: This register must be set while ***[ADxCMPEN]***<CMP0EN>=0.

### 4.2.10. *[ADxCMPCR1]* (Monitor function Setting Register1)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:12 | - | 0 | R | Read as "0". |
| 11:8 | CMPCNT1[3:0] | 0000 | R/W | Comparison count<br>0000: 1 1000: 9<br>0001: 2 1001: 10<br>0010: 3 1010: 11<br>0011: 4 1011: 12<br>0100: 5 1100: 13<br>0101: 6 1101: 14<br>0110: 7 1110: 15<br>0111: 8 1111: 16 |
| 7 | - | 0 | R | Read as "0". |
| 6 | CMPCND1 | 0 | R/W | Determination condition<br>0: Continuous count<br>1: Accumulated count |
| 5 | ADBIG1 | 0 | R/W | Magnitude determination setting<br>0: Conversion result specified by <REGS1> > ***[ADxCMP1]*** (Larger than the comparison register)<br>1: Conversion result specified by <REGS1> < ***[ADxCMP1]*** (Smaller than the comparison register) |
| 4:0 | REGS1[4:0] | 00000 | R/W | Compared conversion result storage register<br>00000: ADxREG0 01000: ADxREG8 10000: ADxREG16<br>00001: ADxREG1 01001: ADxREG9 10001: ADxREG17<br>00010: ADxREG2 01010: ADxREG10 10010: ADxREG18<br>00011: ADxREG3 01011: ADxREG11 10011: ADxREG19<br>00100: ADxREG4 01100: ADxREG12 10100: ADxREG20<br>00101: ADxREG5 01101: ADxREG13 10101: ADxREG21<br>00110: ADxREG6 01110: ADxREG14 10110: ADxREG22<br>00111: ADxREG7 01111: ADxREG15 10111: ADxREG23<br>11000 to 11111: Inhibited setting. |

Note: This register must be set while ***[ADxCMPEN]***<CMP1EN>=0.

### 4.2.11. *[ADxCMP0]* (Conversion Result Comparison Register0)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:4 | AD0CMP0[11:0] | 0x000 | R/W | AD conversion result comparison value storage<br>The value compared with the AD conversion result is set. |
| 3:0 | - | 0 | R | Read as "0". |

Note: This register must be set while ***[ADxCMPEN]***<CMP0EN>=0.

### 4.2.12. *[ADxCMP1]* (Conversion Result Comparison Register1)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:4 | AD0CMP1[11:0] | 0x000 | R/W | AD conversion result comparison value storage<br>The value compared with the AD conversion result is set. |
| 3:0 | - | 0 | R | Read as "0". |

Note: This register must be set while ***[ADxCMPEN]***<CMP1EN>=0.

## 4.2.13. PMD Trigger Control Registers

### 4.2.13.1. *[ADxPSEL0]* (PMD Trigger Program Number Selection Register0)

The following is an example of ***[ADxPSEL0]***. ***[ADxPSEL1]*** to ***[ADxPSEL11]*** have the same configuration.

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7 | PENS0 | 0 | R/W | PDTRG0 trigger control (Note2)<br>0: Disabled.<br>1: Enabled. |
| 6:4 | - | 0 | R | Read as "0". |
| 3:0 | PMDS0[3:0] | 0000 | R/W | Program number selection<br>0000: Program 0<br>0001: Program 1<br>0010: Program 2<br>0011: Program 3<br>0100: Program 4<br>0101: Program 5<br>0110: Program 6<br>0111: Program 7<br>1000: Program 8<br>1001 to 1111: Reserved |

Note1: This register must be set while ***[ADxCR0]***<ADEN>=0.

Note2: For details of the PMD, please refer to "Advanced Programmable Motor Control Circuit" of the reference manual.

### 4.2.13.2. *[ADxPINTS0]* (PMD Trigger Interrupt Selection Register0)

The following is an example of ***[ADxPINTS0]***. ***[ADxPINTS1]*** to ***[ADxPINTS8]*** have the same configuration.

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:3 | - | 0 | R | Read as "0". |
| 2:0 | INTSEL0[2:0] | 000 | R/W | Interrupt selection<br>000: No interrupts.<br>001: INTADxPDA<br>010: INTADxPDB<br>011: INTADxPDC<br>100: INTADxPDD<br>101 to 111: No interrupts.<br>This field selects an interrupt for the program 0. |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.13.3. *[ADxPREGS0]* (PMD Trigger Storage Selection Register0)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31 | - | 0 | R | Read as "0". |
| 30:28 | REGSEL7[2:0] | 000 | R/W | Program 7 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 27 | - | 0 | R | Read as "0". |
| 26:24 | REGSEL6[2:0] | 000 | R/W | Program 6 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 23 | - | 0 | R | Read as "0". |
| 22:20 | REGSEL5[2:0] | 000 | R/W | Program 5 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 19 | - | 0 | R | Read as "0". |
| 18:16 | REGSEL4[2:0] | 000 | R/W | Program 4 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 15 | - | 0 | R | Read as "0". |
| 14:12 | REGSEL3[2:0] | 000 | R/W | Program 3 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 11 | - | 0 | R | Read as "0". |
| 10:8 | REGSEL2[2:0] | 000 | R/W | Program 2 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 7 | - | 0 | R | Read as "0". |
| 6:4 | REGSEL1[2:0] | 000 | R/W | Program 1 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | REGSEL0[2:0] | 000 | R/W | Program 0 conversion result storage register selection<br>000: ADxREG0 to 3 100: ADxREG16 to 19<br>001: ADxREG4 to 7 101: ADxREG20 to 23<br>010: ADxREG8 to 11 110: Inhibited setting.<br>011: ADxREG12 to 15 111: Inhibited setting. |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.13.4. *[ADxPREGS1]* (PMD Trigger Storage Selection Register1)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:3 | - | 0 | R | Read as "0". |
| 2:0 | REGSEL8[2:0] | 000 | R/W | Program 8 conversion result storage register selection<br>000: ADxREG0 to 3<br>001: ADxREG4 to 7<br>010: ADxREG8 to 11<br>011: ADxREG12 to 15<br>100: ADxREG16 to 19<br>101: ADxREG20 to 23<br>110: Inhibited setting.<br>111: Inhibited setting. |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.13.5. *[ADxPSET0]* (PMD Trigger Program Register0)

The following is an example of ***[ADxPSET0]***. ***[ADxPSET1]*** to ***[ADxPSET8]*** have the same configuration.

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31 | ENSP03 | 0 | R/W | Conversion 3 setting: Conversion control<br>0: Disabled.<br>1: Enabled. |
| 30:29 | UVWIS03[1:0] | 00 | R/W | Conversion 3 setting: Phase select (for Vector Engine)<br>00: Not specified<br>01: U<br>10: V<br>11: W |
| 28:24 | AINSP03[4:0] | 00000 | R/W | Conversion 3 setting: AIN selection(Note)<br>00000: AINx00 01000: AINx08 10000: AINx16<br>00001: AINx01 01001: AINx09 10001: AINx17<br>00010: AINx02 01010: AINx10 10010: AINx18<br>00011: AINx03 01011: AINx11 10011: AINx19<br>00100: AINx04 01100: AINx12 10100: AINx20<br>00101: AINx05 01101: AINx13 10101: AINx21<br>00110: AINx06 01110: AINx14 10110: AINx22<br>00111: AINx07 01111: AINx15 10111: AINx23<br>11000 to 11111: Inhibited setting. |
| 23 | ENSP02 | 0 | R/W | Conversion 2 setting: Conversion control<br>0: Disabled.<br>1: Enabled. |
| 22:21 | UVWIS02[1:0] | 00 | R/W | Conversion 2 setting: Phase select (for Vector Engine)<br>00: Not specified<br>01: U<br>10: V<br>11: W |
| 20:16 | AINSP02[4:0] | 00000 | R/W | Conversion 2 setting: AIN selection(Note)<br>00000: AINx00 01000: AINx08 10000: AINx16<br>00001: AINx01 01001: AINx09 10001: AINx17<br>00010: AINx02 01010: AINx10 10010: AINx18<br>00011: AINx03 01011: AINx11 10011: AINx19<br>00100: AINx04 01100: AINx12 10100: AINx20<br>00101: AINx05 01101: AINx13 10101: AINx21<br>00110: AINx06 01110: AINx14 10110: AINx22<br>00111: AINx07 01111: AINx15 10111: AINx23<br>11000 to 11111: Inhibited setting. |

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 15 | ENSP01 | 0 | R/W | Conversion 1 setting: Conversion control<br>0: Disabled.<br>1: Enabled. |
| 14:13 | UVWIS01[1:0] | 00 | R/W | Conversion 1 setting: Phase select (for Vector Engine)<br>00: Not specified<br>01: U<br>10: V<br>11: W |
| 12:8 | AINSP01[4:0] | 00000 | R/W | Conversion 1 setting: AIN selection(Note)<br>00000: AINx00 01000: AINx08 10000: AINx16<br>00001: AINx01 01001: AINx09 10001: AINx17<br>00010: AINx02 01010: AINx10 10010: AINx18<br>00011: AINx03 01011: AINx11 10011: AINx19<br>00100: AINx04 01100: AINx12 10100: AINx20<br>00101: AINx05 01101: AINx13 10101: AINx21<br>00110: AINx06 01110: AINx14 10110: AINx22<br>00111: AINx07 01111: AINx15 10111: AINx23<br>11000 to 11111: Inhibited setting. |
| 7 | ENSP00 | 0 | R/W | Conversion 0 setting: Conversion control<br>0: Disabled.<br>1: Enabled. |
| 6:5 | UVWIS00[1:0] | 00 | R/W | Conversion 0 setting: Phase select (for Vector Engine)<br>00: Not specified<br>01: U<br>10: V<br>11: W |
| 4:0 | AINSP00[4:0] | 00000 | R/W | Conversion 0 setting: AIN selection(Note)<br>00000: AINx00 01000: AINx08 10000: AINx16<br>00001: AINx01 01001: AINx09 10001: AINx17<br>00010: AINx02 01010: AINx10 10010: AINx18<br>00011: AINx03 01011: AINx11 10011: AINx19<br>00100: AINx04 01100: AINx12 10100: AINx20<br>00101: AINx05 01101: AINx13 10101: AINx21<br>00110: AINx06 01110: AINx14 10110: AINx22<br>00111: AINx07 01111: AINx15 10111: AINx23<br>11000 to 11111: Inhibited setting. |

Note: The AIN which the product does not have is inhibited to be set (Refer to "Product Information" of the reference manual).

### 4.2.14. *[ADxTSET0]* (General Purpose Start up Factor Program Register0)

The following is an example of ***[ADxTSET0]***. ***[ADxTSET1]*** to ***[ADxTSET23]*** have the same configuration.

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7 | ENINT0 | 0 | R/W | Conversion Result Storage Register0 setting: Interrupt control<br>0: Disabled.<br>1: Enabled. |
| 6:5 | TRGS0[1:0] | 00 | R/W | Conversion Result Storage Register0 setting: Conversion control<br>00: Conversion is inhibited.<br>01: Continuous conversion.<br>10: Single conversion.<br>11: General purpose trigger conversion. |
| 4:0 | AINST0[4:0] | 00000 | R/W | Conversion Result Storage Register0 setting: AIN selection(Note)<br>00000: AINx00 01000: AINx08 10000: AINx16<br>00001: AINx01 01001: AINx09 10001: AINx17<br>00010: AINx02 01010: AINx10 10010: AINx18<br>00011: AINx03 01011: AINx11 10011: AINx19<br>00100: AINx04 01100: AINx12 10100: AINx20<br>00101: AINx05 01101: AINx13 10101: AINx21<br>00110: AINx06 01110: AINx14 10110: AINx22<br>00111: AINx07 01111: AINx15 10111: AINx23<br>11000 to 11111: Inhibited setting. |

Note: The AIN which the product does not have is inhibited to be set (Refer to "Product Information" of the reference manual).

### 4.2.15. *[ADxREG0]* (Conversion Result Storage Register0)

The following is an example of ***[ADxREG0]***. ***[ADxREG1]*** to ***[ADxREG23]*** have the same configuration.

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:30 | - | 0 | R | Read as "0". |
| 29 | ADOVRF_M0 | 0 | R | Mirror bit of overrun flag <ADOVRF0> |
| 28 | ADRF_M0 | 0 | R | Mirror bit of AD conversion result storage flag <ADRF0> |
| 27:16 | ADR_M0[11:0] | 0x000 | R | Mirror bit of AD conversion result <ADR0>.<br>The AD conversion result is read from the lower 12 bits in the upper half word of ***[ADxREG0]*** register. |
| 15:4 | ADR0[11:0] | 0x000 | R | AD conversion result is stored.<br>The AD conversion result is read from the upper 12 bits in the lower half word of ***[ADxREG0]*** register. |
| 3:2 | - | 0 | R | Read as "0". |
| 1 | ADOVRF0 | 0 | R | Overrun flag<br>0: Not occurred.<br>1: Occurred.<br>This flag is set to "1", when an AD conversion result is overwritten before the ***[ADxREG0]*** register is read. This flag is cleared to "0" when it is read. |
| 0 | ADRF0 | 0 | R | AD conversion result storage flag<br>0: No conversion results are stored.<br>1: A conversion result is stored.<br>This flag is set to "1" when an AD conversion value is stored. This flag is cleared to "0" when it is read. |

### 4.2.16. *[ADxPROSEL]* (Program restart selection register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:15 | - | 0 | R | Read as "0". |
| 14 | PROSEL14 | 0 | R/W | Selects to restart or cancel continuous conversion programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 13 | PROSEL13 | 0 | R/W | Selects to restart or cancel single conversion programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 12 | PROSEL12 | 0 | R/W | Selects to restart or cancel GPTRG programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 11 | PROSEL11 | 0 | R/W | Selects to restart or cancel PDTRG11 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 10 | PROSEL10 | 0 | R/W | Selects to restart or cancel PDTRG10 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 9 | PROSEL9 | 0 | R/W | Selects to restart or cancel PDTRG9 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 8 | PROSEL8 | 0 | R/W | Selects to restart or cancel PDTRG8 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 7 | PROSEL7 | 0 | R/W | Selects to restart or cancel PDTRG7 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 6 | PROSEL6 | 0 | R/W | Selects to restart or cancel PDTRG6 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 5 | PROSEL5 | 0 | R/W | Selects to restart or cancel PDTRG5 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 4 | PROSEL4 | 0 | R/W | Selects to restart or cancel PDTRG4 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 3 | PROSEL3 | 0 | R/W | Selects to restart or cancel PDTRG3 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 2 | PROSEL2 | 0 | R/W | Selects to restart or cancel PDTRG2 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 1 | PROSEL1 | 0 | R/W | Selects to restart or cancel PDTRG1 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |
| 0 | PROSEL0 | 0 | R/W | Selects to restart or cancel PDTRG0 programming.<br>0: Restart (Restarts from the point of conversion where pending/suspending had been started)<br>1: Cancel a restart (Canceled at the point of conversion where pending/suspending had been canceled.) |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.17. *[ADxPFLG]* (Startup priority status register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:15 | - | 0 | R | Read as "0". |
| 14 | PFLG14 | 0 | R | Priority status flag for start up factor of continuous conversion<br>0: -<br>1: pending/suspending occurred. |
| 13 | PFLG13 | 0 | R | Priority status flag for start up factor of single conversion<br>0: -<br>1: pending/suspending occurred. |
| 12 | PFLG12 | 0 | R | Priority status flag for start up factor of GPTRG<br>0: -<br>1: pending/suspending occurred. |
| 11 | PFLG11 | 0 | R | Priority status flag for start up factor of PDTRG11<br>0: -<br>1: pending/suspending occurred. |
| 10 | PFLG10 | 0 | R | Priority status flag for start up factor of PDTRG10<br>0: -<br>1: pending/suspending occurred. |
| 9 | PFLG9 | 0 | R | Priority status flag for start up factor of PDTRG9<br>0: -<br>1: pending/suspending occurred. |
| 8 | PFLG8 | 0 | R | Priority status flag for start up factor of PDTRG8<br>0: -<br>1: pending/suspending occurred. |
| 7 | PFLG7 | 0 | R | Priority status flag for start up factor of PDTRG7<br>0: -<br>1: pending/suspending occurred. |
| 6 | PFLG6 | 0 | R | Priority status flag for start up factor of PDTRG6<br>0: -<br>1: pending/suspending occurred. |
| 5 | PFLG5 | 0 | R | Priority status flag for start up factor of PDTRG5<br>0: -<br>1: pending/suspending occurred. |
| 4 | PFLG4 | 0 | R | Priority status flag for start up factor of PDTRG4<br>0: -<br>1: pending/suspending occurred. |
| 3 | PFLG3 | 0 | R | Priority status flag for start up factor of PDTRG3<br>0: -<br>1: pending/suspending occurred. |
| 2 | PFLG2 | 0 | R | Priority status flag for start up factor of PDTRG2<br>0: -<br>1: pending/suspending occurred. |
| 1 | PFLG1 | 0 | R | Priority status flag for start up factor of PDTRG1<br>0: -<br>1: pending/suspending occurred. |
| 0 | PFLG0 | 0 | R | Priority status flag for start up factor of PDTRG0<br>0: -<br>1: pending/suspending occurred. |

Note: If pending/suspending occurs, a flag is set. If the same trigger occurs next time, the previous flag is cleared.

### 4.2.18. *[ADxPINT]* (Priority interrupt control register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:15 | - | 0 | R | Read as "0". |
| 14 | PINT14 | 0 | R/W | Controls a request in case that continuous conversion loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 13 | PINT13 | 0 | R/W | Controls a request in case that single conversion loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 12 | PINT12 | 0 | R/W | Controls a request in case that GPTRG loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 11 | PINT11 | 0 | R/W | Controls a request in case that PDTRG11 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 10 | PINT10 | 0 | R/W | Controls a request in case that PDTRG10 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 9 | PINT9 | 0 | R/W | Controls a request in case that PDTRG9 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 8 | PINT8 | 0 | R/W | Controls a request in case that PDTRG8 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 7 | PINT7 | 0 | R/W | Controls a request in case that PDTRG7 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 6 | PINT6 | 0 | R/W | Controls a request in case that PDTRG6 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 5 | PINT5 | 0 | R/W | Controls a request in case that PDTRG5 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 4 | PINT4 | 0 | R/W | Controls a request in case that PDTRG4 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 3 | PINT3 | 0 | R/W | Controls a request in case that PDTRG3 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 2 | PINT2 | 0 | R/W | Controls a request in case that PDTRG2 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 1 | PINT1 | 0 | R/W | Controls a request in case that PDTRG1 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |
| 0 | PINT0 | 0 | R/W | Controls a request in case that PDTRG0 loses the priority.<br>0: Disables interrupt requests<br>1: Enables interrupt request |

Note: This register must be set while ***[ADxCR0]***<ADEN>=0.

### 4.2.19. *[SHxTRGPAT]* (Trigger switching pattern setting register)

| Bit | Bit Symbol | After Reset | Type | Function |
|---|---|---|---|---|
| 31:4 | - | 0 | R | Read as "0". |
| 3:2 | TRGPAT1[1:0] | 00 | R/W | Selects an upper pattern of trigger combination.<br>(Selects a PMD trigger / ADxTRGIN pattern)<br>00: Upper pattern 0<br>01: Upper pattern 1<br>10: Upper pattern 2<br>11: Upper pattern 3 |
| 1:0 | TRGPAT0[1:0] | 00 | R/W | Selects a lower pattern of trigger combination.<br>(Selects a PMD trigger pattern)<br>00: Lower pattern 0<br>01: Lower pattern 1<br>10: Lower pattern 2<br>11: Lower pattern 3 |

Note1: This register must be set while ***[ADxCR0]***<ADEN>=0.

Note2: For combination patterns, refer to "Table 3.4 Trigger combination patterns".

# 5. Usage example

## 5.1. Single conversion

The single conversion is started by software and enable more than one conversion.

In the following setting example, the conversion results of the two analog inputs (AINx02, AINx03) are saved in two result storage registers (***[ADxREG4]***, ***[ADxREG5]***), and a single conversion interrupt INTADxSGL is generated at the end of second conversion.

- Initial setting
  - ***[ADxMOD0]*** =0x00000001
    DAC ON: <DACON>=1
    Normal operation: <RCUT>=0
  - ***[ADxCLK]*** =0x00000000
    Conversion Clock Setting: Conversion time 0.5[μs] at ADCLK=120[MHz]
  - ***[ADxMOD1]*** =0x00304000
    MODE setting 1: Conversion time 0.5[μs] at ADCLK=120[MHz]
  - ***[ADxMOD2]*** =0x00000000
    MODE setting 2: Conversion time 0.5[μs] at ADCLK=120[MHz]

- Conversion program setting
  - ***[ADxTSET4]*** =0x00000042
    Single conversion: <TRGS4>=10
    AINx02: <AINST4>=00010
    Disable interrupt output: <ENINT4>=0
  - ***[ADxTSET5]*** =0x000000C3
    Single conversion: <TRGS5>=10
    AINx03: <AINST5>=00011
    Enable interrupt output: <ENINT5>=1

- Conversion start setting
  - ***[ADxCR1]*** =0x00000000
    Disable DMA request
  - ***[ADxCR0]*** =0x00000082
    Enable ADC: <ADEN>=1
    Disable continuous conversion: <CNT>=0
    Enable single conversion: <SGL>=1 ; Conversion start

## 5.2. PMD trigger conversion

### 5.2.1. PMD (3-shunt), ADC × 1

The following shows the connection diagram in which PMD channel 0 and ADC unit A are used in 3-shunt.

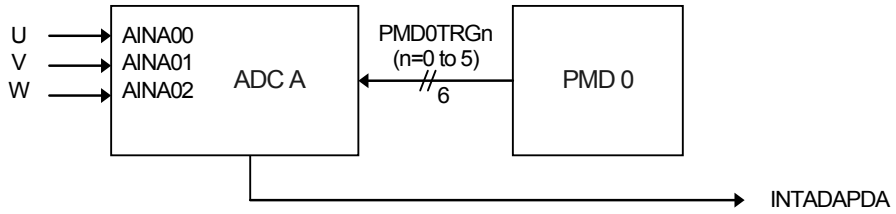



**Figure 5.1 3-shunt example**

The setting example for the ADC is as follows in this case:

**Table 5.1 ADC setting in 3-shunt**

| Program | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Reg0 | U | V | W | V | W | U |
| Reg1 | V | W | U | U | V | W |
| INT | INTADAPDA | INTADAPDA | INTADAPDA | INTADAPDA | INTADAPDA | INTADAPDA |

The 6 trigger inputs PMD0TRG0 to PMD0TRG5 are assigned to the programs 0 to 5 by ***[ADAPSEL0]*** to ***[ADAPSEL5]***.

Reg0 and Reg1 in the table represent ***[ADAPSETn]***[7:0] and ***[ADAPSETn]***[15:8] (n: Program number), respectively. U, V, and W in the table are the motor phases. The corresponding AIN input should be selected for each phase.

When the trigger is received, the AD conversion starts and the execution is done in the order of Reg0 and Reg1. Each conversion result is stored to the conversion result storage register, and the INTADAPDA interrupt is generated.

### 5.2.2. PMD (1-shunt), ADC × 1

The following shows the connection diagram in which PMD channel 0 and ADC unit A are used in 1-shunt.

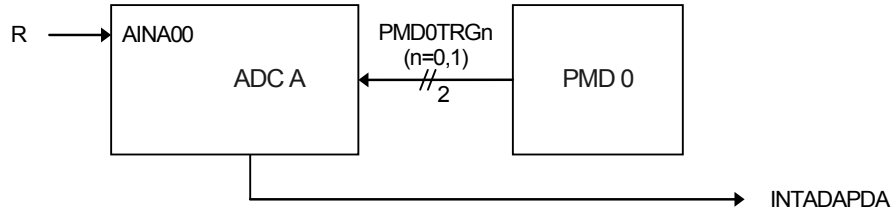



**Figure 5.2 1-shunt example**

The setting example for the ADC is as follows in this case:

**Table 5.2 ADC setting in 1-shunt**

| Trigger | PMD0 | PMD0 |
|---|---|---|
| | 0 | 1 |
| Program | 0 | 1 |
| Reg0 | R | - |
| Reg1 | - | R |
| INT | - | INTADAPDA |

The two trigger signals from PMD0 are assigned to the program numbers, respectively.

Reg0 and Reg1 in the table represent ***[ADAPSETn]***[7:0] and ***[ADAPSETn]***[15:8] (n: Program number), respectively. R in the table is a resistor. It is connected to the corresponding AIN.

When the trigger is received, ADC unit A starts and the conversion result is stored to the conversion result storage register0 and 1. The conversion executes in the order of the program0 and 1. The INTADAPDA interrupt is generated at the conversion completion.

# 6. Precaution

- The AD conversion result may have some variation due to the fluctuation of the power supply and surrounding noises. The data of the output pins should not be changed during AD conversion to prevent from degrading the AD conversion accuracy. The AD conversion accuracy may degrade if the signal on the shared pin with the AD input/output changes or other output pin changes its output during the AD conversion. In the above case, the AD conversion result should be acquired with the mean value of multiple conversion results and other countermeasures.

- Measures should be taken to prevent digital noise from mixing into the analog power supply pins(AVDD5, AVSS) and the reference voltage pins(VREFHx, VREFLx) of the ADC.
  - Insert a bypass capacitor between AVDD5 and AVSS pins, the VREFHx and VREFLx pins. Place the capacitor as close to the terminal as possible.

# 7. Revision History

**Table 7.1 Revision History**

| Revision | Date | Description |
|---|---|---|
| 1.0 | 2018-01-12 | First release |
| 2.0 | 2018-03-20 | - 1.Outlines Table<br>Conversion time/Operation explanation: Modified<br>Store conversion result/Operation explanation: 32→24<br>Start up by General Purpose Factor/Operation explanation: Modified<br>Start up by PMD trigger /Operation explanation: start up→<br>conversion program<br>AD monitor function /Operation explanation: Modified<br>Note : presence / absence →enable / disable<br>- Figure 1.1 : A-PMD→PMD<br>- 2. Block Diagram Title: Configuration<br>- Figure 2.1 : ADxTSET31→ADxTSET23, ADxREG31→ADxREG23<br>Added ADxMOD2<br>Trigger from peripheral function→<br>Trigger from peripheral function via TRGSEL<br>- 3. Function and Operation<br>2nd and subsequent: Separated into "3.1.Clock Supply"<br>Content modified<br>- 3.2. Title: Strat→Start<br>- 3.2.1.Operation "AIN select" → "AIN select <AINSTn>"<br>"(Interrupt control)<AINSTn>"→"(Interrupt control)<ENINTn>"<br>activation→start up<br>DMA request explanation modified.<br>- Figure 3.1: ADxTSET31→ADxTSET23, ADxREG31→ADxREG23<br>- Table 3.1 Title: Factor →Start up factor<br>- 3.2.2.Control Registers 1st term: ADxREG31→ADxREG23<br>general start up→general purpose start up<br>2nd term,3rd term: Modified<br>Before 4th term: Modified<br>5th term: (5),Note modified<br>- Figure 3.2 Lower side: "24 to 31"→ "-"<br>- 3.3.2. Control Registers<br>1st term: Modified<br>2nd term: (PDTRGn (n=0 to 11) → (PDTRG0 to PDTRG11)<br>5th term: Deleted " 24 to 27", "28 to 31"<br>6th item and after : Modified<br>Procedure (5): <UVWISn> of phase select (for Vector Engine)<br>- 3.5.Start up Priority<br>program →conversion program<br>status of the program →execution status of the conversion program<br>later factor →later start up factor, current factor →current start up factor<br>halted →suspended, resumed →restarted<br>- 3.6.AD Monitor Function<br>7th line: whose → which<br>Under Table 3.3: Deleted "For the handling of the interrupts, refer to ..."<br>(1),(2): [15:4]→<AD0CMP0>, <ADxCMP0>→<AD0CMP0><br>- 4.1.List of Registers Deleted:<br>General Purpose Start up Factor Program Register24 to 31<br>Conversion Result Storage Register24 to 31<br>- 4.2.7. Modified Title line<br>- 4.2.9./4.2.10. REGSn/Function<br>11000 to 11111: Modified to "Inhibited setting."<br>- 4.2.13.1. Modified Note2<br>- 4.2.13.3./4.2.13.4 REGSELn/Function<br>110,111: Modified to "Inhibited setting." |

| | | |
|---|---|---|
| | | - 4.2.14. 1st line: ADxPSET1→ADxTSET1, ADxTSET31→ADxTSET23<br>- 4.2.15. 1st line: ADxREG31→ ADxREG23<br>- 5.1.Single conversion<br>Initial setting term: Conversion Clock setting 1→Conversion Clock Setting<br>- 5.2.1. Modified: ***[PMD0TRG0]*** to ***[PMD0TRG5]***→<br>PMD0TRG0 to PMD0TRG5<br>- 5.2.2. Modified: 1-Shunt,1 shunt → 1-shunt<br>- Table 5.2 Modified title<br>ADC unit A in 1-shunt →ADC setting in 1-shunt |

## RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**