SECTION 13
CONFIGURABLE TIMER MODULE (CTM9)

13.1 Introduction

The configurable timer module (CTM9) is a family of timer modules for the Motorola modular microcontroller family (MMF), including the MC68300 (CPU32) and the MC68HC16 (CPU16) families of microcontrollers (MCUs). The timer architecture is modular relative to the number of time-bases (counter submodules), channels (action submodules) and other available general purpose functions (real time clock, RAM, I/O ports, etc.) that can be included.

Please refer to the ***CTM Reference Manual (CTMRM/D)*** for more information.

13.1.1 CTM9 Configuration

The CTM9 is composed of the following submodules:

- 1 free-running counter submodule (FCSM).
- 2 modulus counter submodule (MCSM).
- 4 single action submodule (SASM).
- 4 double action submodule (DASM).
- 4 dedicated PWM submodule (PWMSM).
- 1 bus interface unit submodule (BIUSM).
- 1 counter prescaler submodule (CPSM).

Figure 13-1 and **Table 13-1** show respectively a block diagram and a table representation of the CTM9 configuration.

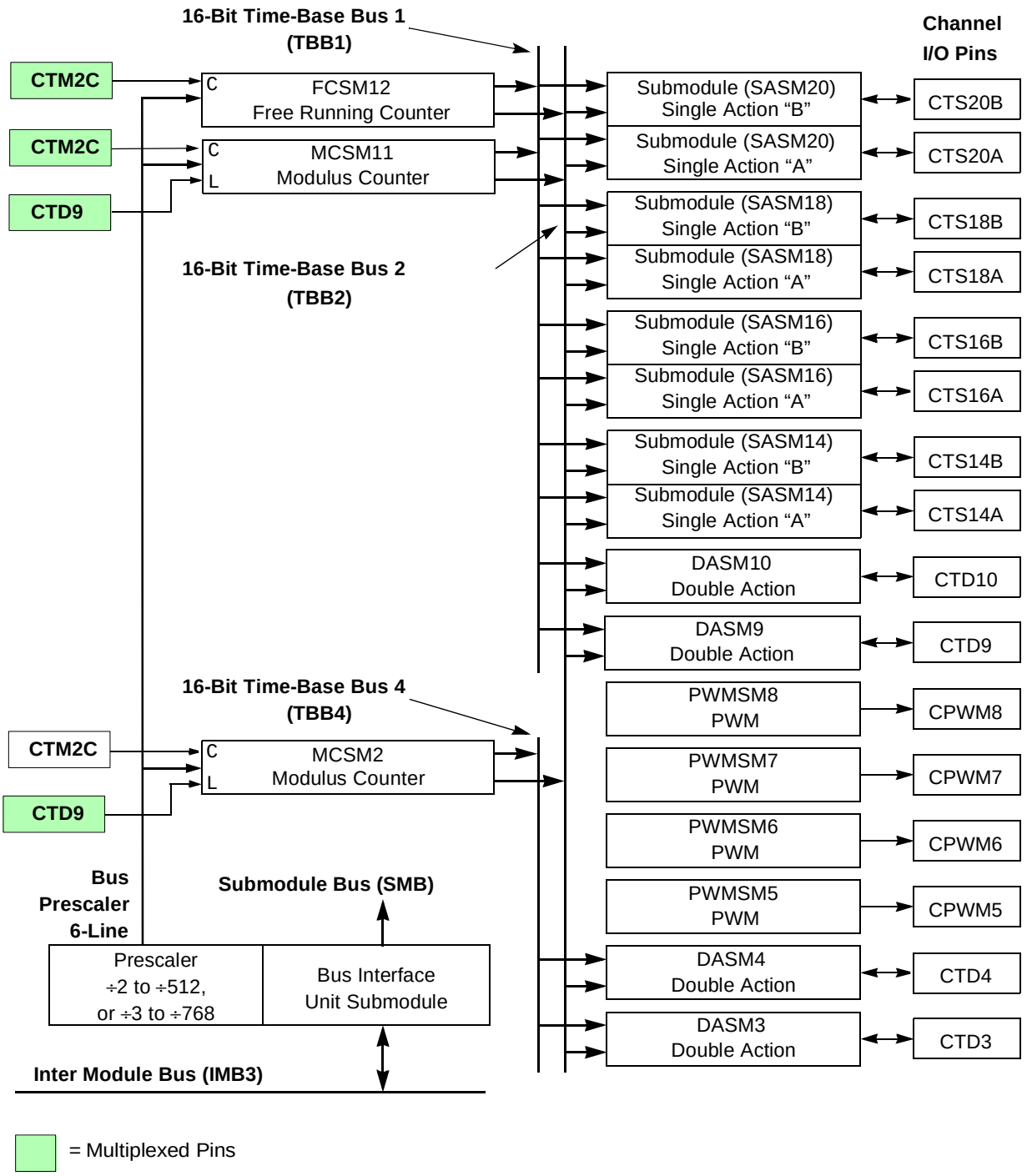


Figure 13-1 Configurable Timer Module, CTM9 Block Diagram

Table 13-1 CTM9 Configuration Description

Submodule Type	Submodule Number	Connected to: tbb 1 (A)	tbb 2 (B)	tbb 3 (B)	tbb 4 (A)	Binary Interrupt Vector Number	Submodule Base Address	Pin Function	Input Pin Name	Output Pin Name
BIUSM+CPSM	0						0xYF F200			
MCSM	2[1]	0	1	0	1	0bxx000010[2]	0xYF F210[3]	Clock	CTM2C	
								Load	CTD9	
DASM	3	0	1	0	1	0bxx000011	0xYF F218	Channel	CTD3	CTD3
DASM	4	0	1	0	1	0bxx000100	0xYF F220	Channel	CTD4	CTD4
PWMSM	5					0bxx000101	0xYF F228	Channel		CPWM 5
PWMSM	6					0bxx000110	0xYF F230	Channel		CPWM 6
PWMSM	7					0bxx000111	0xYF F238	Channel		CPWM 7
PWMSM	8					0bxx001000	0xYF F240	Channel		CPWM 8
DASM	9	1	1	0	0	0bxx001001	0xYF F248	Channel	CTD9	CTD9
DASM	10	1	1	0	0	0bxx001010	0xYF F250	Channel	CTD10	CTD10
MCSM	11	1	1	0	0	0bxx001011	0xYF F258	Clock	CTM2C	
								Load	CTD9	
FCSM	12	1	1	0	0	0bxx001100	0xYF F260	Clock	CTM2C	
SASM	14	1	1	0	0	0bxx001110	0xYF F270	Channel A	CTS14 A	CTS14 A
						0bxx001111		Channel B	CTS14 B	CTS14 B
SASM	16	1	1	0	0	0bxx010000	0xYF F280	Channel A	CTS16 A	CTS16 A
						0bxx010001		Channel B	CTS16 B	CTS16 B
SASM	18	1	1	0	0	0bxx010010	0xYF F290	Channel A	CTS18 A	CTS18 A
						0bxx010011		Channel B	CTS18 B	CTS18 B
SASM	20	1	1	0	0	0bxx010100	0xYF F2A0	Channel A	CTS20 A	CTS20 A
						0bxx010101		Channel B	CTS20 B	CTS20 B

NOTES:

1. Interrupt arbitration priority goes in descending order with submodule #2 having the highest priority and the last interrupting submodule in the table having the lowest priority.
2. xx are the two VECT[7:6] bits contained in the BIUSM.
3. Y=m111, where m is the state of the modmap bit of the MCR of the SIM (Y=0x7 or 0xF).

13.1.2 CTM9 Pins and Naming Convention

The CTM9 uses 17 pins. The usage of these pins is shown in **Figure 13-1** and **Table 13-1**. The CTM9 digital input and output pin names are composed of three sections according to the following convention:

<submodule prefix><submodule number><submodule suffix (optional)>

The pin prefix and suffix for the different submodules used in the CTM9 are as follows:

- FCSM pin name:
  <submodule prefix>: “CTF”
  <submodule suffix>: none
  For example an FCSM placed as submodule number n would have its corresponding input clock pin called: CTFn
- MCSM pin name:
  <submodule prefix>: “CTM”
  <submodule suffix>: C for the Clock pin
  <submodule suffix>: L for the Load pin
  For example an MCSM placed as submodule number n would have its corresponding input clock pin named: CTMnC and its input load pin called CTMnL.
- SASM pin name:
  <submodule prefix>: “CTS”
  <submodule suffix>: A for channel A I/O
  <submodule suffix>: B for channel B I/O
  For example an SASM placed as submodule number n would have its corresponding channel A I/O pin named: CTSnA and its channel B I/O pin called: CTSnB.
- DASM pin name:
  <submodule prefix>: “CTD”
  <submodule suffix>: none
  For example a DASM placed as submodule number n would have its corresponding I/O pin called: CTDn
- PWMSM pin name:
  <submodule prefix>: “CPWM”
  <submodule suffix>: none
  For example a PWMSM placed as submodule number n would have its corresponding output pin called: CPWMn

In the CTM9, some pins are multiplexed between submodules. **Table 13-1** shows using the same pin names for the inputs or outputs which are connected together.

13.2 Free Running Counter Submodule (FCSM)

The free-running counter submodule (FCSM) has a 16-bit up counter with an associated clock source selector, selectable time-base bus drivers, software writable control registers, software readable status bits, and interrupt logic. When the 16-bit up counter overflows from 0xFFFF to 0x0000, an optional overflow interrupt is available to the software. The current state of the 16-bit counter is the primary output of the counter submodules. The software selects which, if any, time-base bus is to be driven by the

16-bit counter. A software control register selects whether the clock input to the counter is one of the taps from the prescaler or an input pin. The polarity of the external input pin is also programmable. The free-running counter submodule operation is comparable to the MC68HC11 counter.

A block diagram of the FCSM is shown in **Figure 13-2**. The main components of the FCSM are a 16-bit loadable free-running up-counter, a clock selector, a time base bus driver and an interrupt interface.

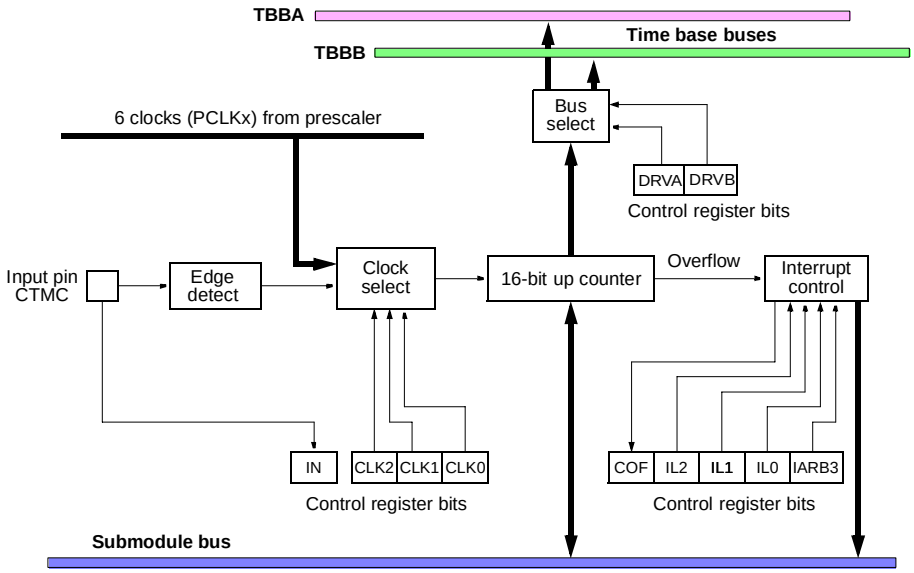


Figure 13-2 FCSM Block Diagram

NOTE

In order to be able to count, the FCSM requires the CPSM clock signals to be present. On coming out of reset, the FCSM will not count internal or external events until the prescaler in the CPSM starts running (when the software sets the PRUN bit). This allows all counters in the CTM submodules to be synchronized.

13.2.1 The FCSM Counter

The FCSM counter section comprises a 16-bit register and a 16-bit up-counter. Reading the register transfers the contents of the counter to the data bus, while a write to the register loads the counter with the new value. Overflow of the counter is defined to be the transition from 0xFFFF to 0x0000. An overflow condition causes the COF flag bit in the FCSMSIC register to be set.

NOTE

Reset presets the counter register to 0x0000. Writing 0x0000 to the counter register while the counter's value is 0xFFFF does not set the

COF flag and does not generate an interrupt request.

13.2.2 FCSM Clock Sources

The user can choose from eight software selectable counter clock sources:

- Six prescaler outputs (PCLKx)
- Input pin rising edge detection on the input pin CTMC
- Input pin falling edge detection on the input pin CTMC

The clock source is selected by the CLK[2:0] bits in the FCSM status, interrupt and control register FCSMSIC (see **13.2.7.1 FCSMSIC — FCSM Status/Interrupt/Control Register**). When the CLK[2:0] bits are being changed, internal circuitry ensures that spurious edges occurring on the CTMC pin do not affect the FCSM.

Note that the read-only IN bit of the FCSMSIC register reflects the state of the input pin CTMC. The input pin is Schmitt triggered and is synchronized with the system clock (f_{SYS}).

13.2.3 FCSM External Event Counting

When an external clock source (on the input pin) is selected, the FCSM is in the event counter mode. The counter can simply count the number of events occurring on the input pin. Alternatively, the FCSM can be programmed to generate an interrupt when a predefined number of events have been counted; this is done by presetting the counter with the two's complement value of the desired number of events. When using the external clock source, the maximum guaranteed external frequency is $f_{SYS}/4$.

13.2.4 The FCSM Time Base Bus Driver

The DRVA and DRVB bits in the FCSMSIC register select the time base buses to be driven (see **13.2.7.1 FCSMSIC — FCSM Status/Interrupt/Control Register**).

WARNING

It is not recommended that the two time base buses be driven at the same time.

13.2.5 FCSM Interrupts

A valid FCSM interrupt can be generated when the COF bit in the FCSMSIC register is set (as a result of the counter overflowing). If the interrupt priority level of the FCSM is non-zero, as defined by the three IL bits in the FCSMSIC register, a valid interrupt request will occur on the IMB.

13.2.6 Freeze Action on the FCSM

When the IMB FREEZE signal is recognized, the FCSM counter stops counting and remains set at its current value. When the FREEZE signal is negated, the counter starts incrementing from its current value, as if nothing had happened. All registers are accessible during freeze.

During freeze, the IN bit in the FCSMSIC register continues to reflect the state of the signal on the input pin CTMC (see **13.2.7.1 FCSMSIC — FCSM Status/Interrupt/ Control Register**).

13.2.7 FCSM Registers

The FCSM register map comprises four 16-bit register locations. As shown in **Table 13-2**, the register block contains two FCSM registers and two reserved registers. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no effect. In CTM implementations featuring multiple FCSMs, each FCSM has its own set of registers.

NOTE

All register addresses in this section are offsets from the base address of the FCSM.

Table 13-2 FCSM Register Map

Address	15 — 8	7 — 0
0xYF F260	Status, interrupt and control register (FCSMSIC)	
0xYF F262	Counter register(FCSMCNT)	

13.2.7.1 FCSMSIC — FCSM Status/Interrupt/Control Register

FMSMSIC — FCSM Status/Interrupt Control Register **0xYF F260**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
COF	IL2	IL1	IL0	IARB3	0	DRVA	DRVB	IN	0	0	0	0	CLK2	CLK1	CLK0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13-3 FMSMSIC Bit Settings

Bit(s)	Name	Description
15	COF	Counter overflow flag. This status flag bit indicates whether or not a counter overflow has occurred. An overflow is defined to be the transition of the counter from 0xFFFF to 0x0000. If the IL field is non-zero, an interrupt request is generated when the COF bit is set. The flag clearing mechanism will work only if no flag setting event occurs between the read and write operations; if a COF setting event occurs between the read and write operations, the COF bit will not be cleared. 0 = Counter overflow has not occurred. 1 = Counter overflow has occurred.
14:12	IL[2:0]	Interrupt level. The three interrupt level bits are read/write control bits that select the priority level of interrupt requests made by the FCSM. These bits can be read or written at any time and are cleared by reset. 000 = Interrupt disabled 001 = Interrupt level 1 (lowest) 010 = Interrupt level 2 011 = Interrupt level 3 100 = Interrupt level 4 101 = Interrupt level 5 110 = Interrupt level 6 111 = Interrupt level 7 (highest)
11	IARB3	The read/write IARB3 bit works in conjunction with the IARB[2:0] field in the BIUSM module configuration register. Each module that generates interrupt requests on the IMB must have a unique value in the arbitration field (IARB). This interrupt arbitration identification number is used to arbitrate for the IMB when modules generate simultaneous interrupts of the same priority.
10	—	Reserved
9:8	DRV[A:B]	Drive time base bus. DRVA and DRVB are read/write bits that control the connection of the FCSM to the time base buses A and B. These bits are cleared by reset. It is recommended that the two time base buses **not** be driven at the same time. 00 = Neither time base bus A nor time base bus B is driven. 01 = Time base bus B is driven 10 = Time base bus A is driven 11 = Both time base bus A and time base bus B are driven
7	IN	Input pin status. This read-only status bit reflects the logic state of the FCSM input pin CTMC. Writing a ‘zero’ or a ‘one’ to this bit has no effect. Reset has no effect on this bit.
6:3	—	Reserved
2:0	CLK[2:0]	Counter clock select. These read/write control bits select one of six internal clock signals (PCLKx) or one of two external conditions on the input pin (rising edge or falling edge). The maximum frequency of the external clock signals is $f_{SYS}/4$. 000 = Prescaler output 1 (/2 or /3) 001 = Prescaler output 2 (/4 or /6) 010 = Prescaler output 3 (/8 or /12) 011 = Prescaler output 4 (/16 or /24) 100 = Prescaler output 5 (/32 or /48) 101 = Prescaler output 6 (/64 to /512 or /96 to /768) 110 = CTMC pin input, negative edge 111 = CTMC pin input, positive edge

13.2.7.2 FCSMCNT — FCSM Counter Register

FCSMCNT — FCSM Counter Register **0xYF F262**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FCSM counter register is a read/write register; it is cleared by reset.

13.3 Modulus Counter Submodule (MCSM)

The modulus counter submodule (MCSM) is an enhancement of the free-running counter. A modulus register gives the additional flexibility of recycling the counter at a count other than 64K clock cycles. The state of the modulus register is transferred to the counter under three conditions:

1. When an overflow occurs.
2. When an appropriate transition occurs on the external load pin.
3. When the program writes to the counter register. In this case, the value is first written into the modulus register and immediately transferred to the counter.

Software can also write a value to the modulus register for later loading into the counter with one of the two first criteria. A block diagram of the MCSM is shown in **Figure 13-3**.

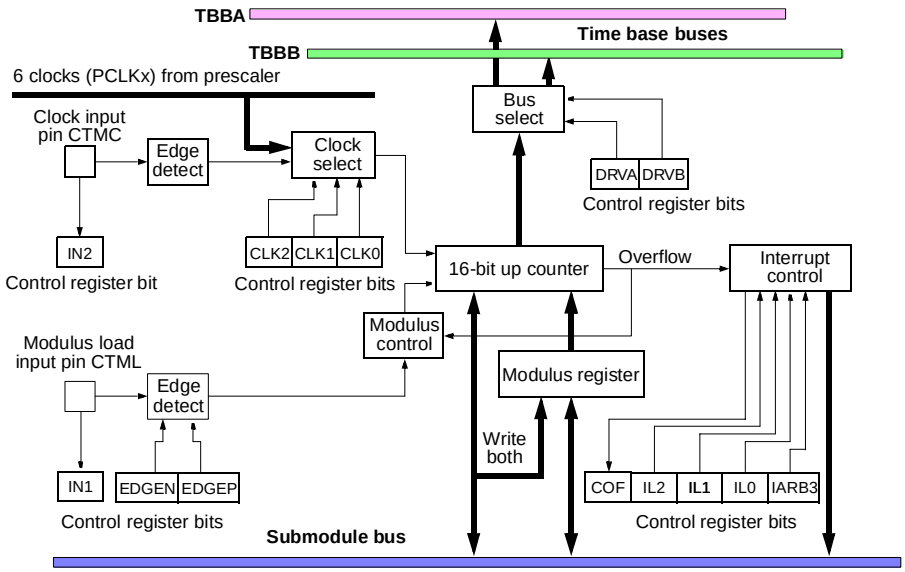


Figure 13-3 MCSM Block Diagram

The main components of the MCSM are a 16-bit modulus latch, a 16-bit loadable up-counter, counter loading logic, a clock selector, a time base bus driver and an interrupt interface.

NOTE

In order to be able to count, the MCSM requires the CPSM clock signals to be present. On coming out of reset, the MCSM will not count internal or external events until the prescaler in the CPSM starts running (when the software sets the PRUN bit). This allows all counters in the CTM submodules to be synchronized.

13.3.1 The MCSM Modulus Latch

The 16-bit modulus latch is a read/write register that is used to reload the counter automatically with a predetermined value. The contents of the modulus latch register can be read at any time. Writing to the register loads the modulus latch with the new value. This value is then transferred to the counter register on the next hardware load of that counter. However, writing to the corresponding counter register loads the modulus latch and the counter register immediately with the new value. The modulus latch register is cleared to 0x0000 by reset.

13.3.2 The MCSM Counter

The counter is composed of a 16-bit read/write register associated with a 16-bit incrementer. Reading the counter transfers the contents of the counter register to the data bus; writing to the counter loads the modulus latch and the counter register immediately with the new value. The counter can be clocked with different clock sources (see **13.3.3 MCSM Clock Sources**).

NOTE

Reset presets the counter register to 0x0000. Writing 0x0000 to the counter register while its value is 0xFFFF does not set the COF flag and does not generate an interrupt.

13.3.2.1 Loading the MCSM Counter Register

The counter register can be loaded by writing directly to it. The counter register is also loaded from the modulus latch each time a counter overflow occurs and the COF flag bit in the MCSM status/interrupt/control register (MCSMSIC) is set.

NOTE

When the modulus latch is loaded with 0xFFFF, the overflow flag is set on every counter clock pulse.

Loading of the counter register from the modulus register can also be triggered by an external event on the modulus load pin CTML. The edge on the CTML pin that triggers the loading of the counter register is selected by bits EDGEN and EDGEP in the MCSMSIC register. Hardware is provided to prevent the occurrence of spurious edges while changing the EDGEN and EDGEP bits. Reset clears the EDGEN and EDGEP

bits to zero, thereby preventing a signal on the CTML pin from loading the counter register until EDGEN and EDGEP have been initialized by the software. The modulus load input pin CTML is Schmitt triggered and synchronized to the system clock (f_{SYS}).

NOTE

The read-only IN1 bit of the MCSMSIC reflects the state of the input pin CTML.

13.3.2.2 Using the MCSM as a Free-Running Counter

The MCSM is a modulus counter. However it can be made to behave like a free-running counter by loading the modulus register with the value 0x0000.

13.3.3 MCSM Clock Sources

The User can choose from eight software selectable counter clock sources:

- Six prescaler outputs (PCLKx)
- Input pin rising edge detection on the input pin CTMC
- Input pin falling edge detection on the input pin CTMC

The clock source is selected by the CLK[2:0] bits in the MCSM status, interrupt and control register MCSMSIC (see **13.3.9 MCSMSIC — MCSM Status/Interrupt/Control Register**). When the CLK[2:0] bits are being changed, internal circuitry ensures that spurious edges occurring on the CTMC pin do not affect the MCSM. The clock input pin CTMC is Schmitt triggered and is synchronized with the system clock (f_{SYS}).

NOTE

The read-only IN2 bit of the MCSMSIC register reflects the state of the input pin CTMC.

13.3.4 MCSM External Event Counting

When an external clock source (on the CTMC input pin) is selected, the MCSM is in the event counter mode. The counter can simply count the number of events occurring on the input pin. Alternatively, the MCSM can be programmed to generate an interrupt when a predefined number of events have been counted; this is done by presetting the counter with the two's complement value of the desired number of events. When using the external clock source, the maximum external guaranteed frequency is $f_{SYS}/4$.

13.3.5 The MCSM Time Base Bus Driver

The DRVA and DRVB bits in the MCSMSIC register select the time base buses to be driven (see **13.3.9 MCSMSIC — MCSM Status/Interrupt/Control Register**).

13.3.6 MCSM interrupts

A valid MCSM interrupt can be generated when the COF bit in the MCSMSIC register is set as a result of the counter overflowing. If the interrupt priority level of the MCSM is non-zero, as defined by the three IL bits in the MCSMSIC register, a valid interrupt request will occur on the IMB.

13.3.7 Freeze Action on the MCSM

When the IMB FREEZE signal is recognized, the MCSM counter stops counting and remains set at its last value. When the FREEZE signal is negated, the counter starts incrementing from its last value, as if nothing had happened. All registers are accessible during freeze.

During freeze, the IN1 and IN2 bits in the MCSMSIC continue to reflect the states of the signals on the input pins (see **13.3.9 MCSMSIC — MCSM Status/Interrupt/Control Register**).

13.3.8 MCSM Registers

The MCSM register map comprises four 16-bit register locations. As shown in **Table 13-4**, the register block contains three FCSM registers and one reserved register. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no effect. In CTM implementations featuring multiple MCSMs, each MCSM has its own set of registers.

Table 13-4 MCSM Register Map

Address	15 — 8	7 — 0
0xYF F210	MCSM2 status/interrupt/control register (MCSM2SIC)	
0xYF F212	MCSM2 counter (MCSM2CNT)	
0xYF F214	MCSM2 modulus latch (MCSM2ML)	
0xYF F258	MCSM11 status/interrupt/control register (MCSM11SIC)	
0xYF F25A	MCSM11 counter (MCSM11CNT)	
0xYF F25C	MCSM11 modulus latch (MCSM11ML)	

13.3.9 MCSMSIC — MCSM Status/Interrupt/Control Register

MCSM2SIC — MCSM Status/Interrupt Control Register **0xYF F210**
MCSM11SIC **0xYF F258**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
COF	IL2	IL1	IL0	IARB3	0	DRVA	DRVB	IN2	IN1	EDGEN	EDGEP	0	CLK 2	CLK 1	CLK 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13-5 MCSMSIC Bit Settings

Bit(s)	Name	Description
15	COF	Counter overflow flag. This status flag bit indicates whether or not a counter overflow has occurred. An overflow of the MCSM counter is defined to be the transition of the counter from 0xFFFF to 0xxxxx, where 0xxxxx is the value contained in the modulus latch. If the IL field is non-zero, an interrupt request is generated when the COF bit is set. This flag bit is set only by the hardware and cleared only by the software or by a system reset. To clear the flag, the software must first read the bit (as '1') then write a '0' to the bit. The flag clearing mechanism will work only if no flag setting event occurs between the read and write operations; if a COF setting event occurs between the read and write operations, the COF bit will not be cleared. 0 = Counter overflow has not occurred 1 = Counter overflow has occurred.
14:12	IL[2:0]	Interrupt level. The three interrupt level bits are read/write control bits that select the priority level of interrupt requests made by the MCSM. These bits can be read or written at any time and are cleared by reset. 000 = Interrupt disabled. 001 = Interrupt level 1 (lowest). 010 = Interrupt level 2. 011 = Interrupt level 3. 100 = Interrupt level 4. 101 = Interrupt level 5. 110 = Interrupt level 6. 111 = Interrupt level 7 (highest).
11	IARB3	Interrupt arbitration bit 3. The read/write IARB3 bit works in conjunction with the IARB[2:0] field in the BIUSM module configuration register. Each module that generates interrupt requests on the IMB must have a unique value in the arbitration field (IARB). This interrupt arbitration identification number is used to arbitrate for the IMB when modules generate simultaneous interrupts of the same priority.
10	—	Reserved
9:8	DRV{A:B]	Drive time base bus. DRVA and DRVB are read/write bits that control the connection of the MCSM to the time base buses A and B. 00 = Neither time base bus A nor time base bus B is driven. 01 = Time base bus B is driven. 10 = Time base bus A is driven. 11 = Both time base bus A and time base bus B are driven.
7	IN2	Clock input pin status. This read-only status bit reflects the logic state of the clock input pin CTMC. Writing a 0 or 1 to this bit has no effect. Reset has no effect on this bit.
6	IN1	Modulus load input pin status. This read-only status bit reflects the logic state of the modulus load input pin CTML. Writing a 0 or 1 to this bit has no effect. Reset has no effect on this bit.
5:4	EDGEN, EDGEP	Modulus load edge sensitivity. These read/write bits select the sensitivity of the edge detection circuitry on the modulus load pin CTML. 00 = None 01 = Positive edge only. 10 = Negative edge only. 11 = Positive and negative edge.
3	—	Reserved
2:0	CLK[2:0]	Counter clock select. These read/write control bits select one of six internal clock signals (PCLKx) or one of two external conditions on the input pin (rising edges or falling edges). The maximum frequency of the external clock signals is $f_{SYS}/4$. 000 = Prescaler output 1 (/2 or /3). 001 = Prescaler output 2 (/4 or /6). 010 = Prescaler output 3 (/8 or /12). 011 = Prescaler output 4 (/16 or /24). 100 = Prescaler output 5 (/32 or /48). 101 = Prescaler output 6 (/64 to /768). 110 = CTMC pin input, negative edge. 111 = CTMC pin input, positive edge.

13.3.10 MCSMCNT — MCSM Counter Register

MCSM2CNT — MCSM Counter Register **0xYF F212**
MCSM11CNT **0xYF F25A**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.11 MCSMML — MCSM Modulus Latch Register

MCSM2ML — MCSM Modulus Latch Register **0xYF F214**
MCSM11ML **0xYF F25C**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4 Single-Action Submodule (SASM)

The single action submodule provides an input capture and an output compare for each of two bidirectional pins. All of the functions associated with one pin comprise a SASM channel. Each channel includes a 16-bit equality comparator and one 16-bit register for saving an input capture value or for holding an output compare value. The input edge detector associated with each pin is programmable to cause the capture function to occur on the rising or falling edge. The output flip flop is set by the software to either toggle when an output compare occurs or to transfer a software provided bit value to the output pin. In either the input capture mode or the output compare mode, a software interrupt may be programmed to occur for each channel. Software selection is provided to select which of the two incoming time-base buses is used for input captures or output compares on each channel. Each channel operates independently. However, interrupt and interrupt priority logic are shared by both SASM channels. See **Figure 13-4** for a SASM block diagram.

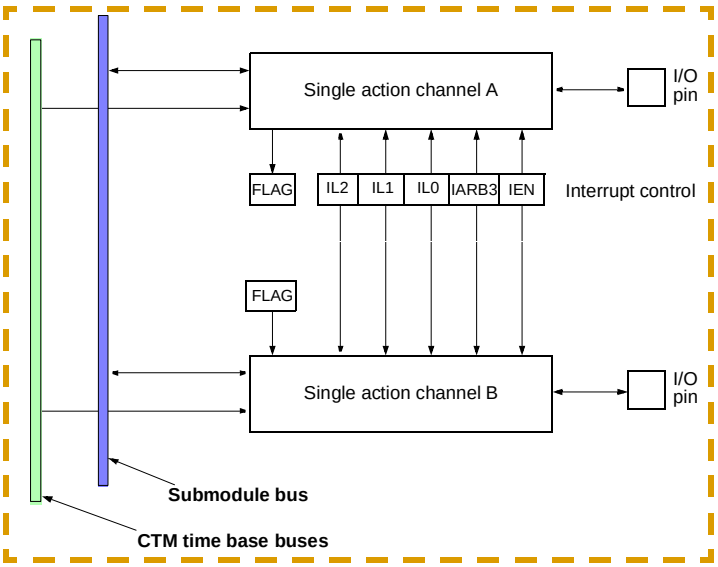


Figure 13-4 SASM Block Diagram

Each SASM channel comprises:

- A time base bus selector (which selects the time base bus to be used by that channel for all timing functions),
- A 16-bit data register (which can be read by the software at any time and which is used for both input capture and output compare functions),
- A 16-bit comparator (which continuously compares the 16-bit value in the data register with the time base bus),
- An output flip-flop (which holds the logic level to be sent to the output pin when a successful output compare occurs),
- An input edge detector (which detects the rising or falling edge that will trigger the input capture function),
- Several status and control bits in the status/interrupt/control register SICA or SICB,
- An interrupt section.

NOTE

During reset the output of the output flip-flop is cleared (i.e., to ‘0’).

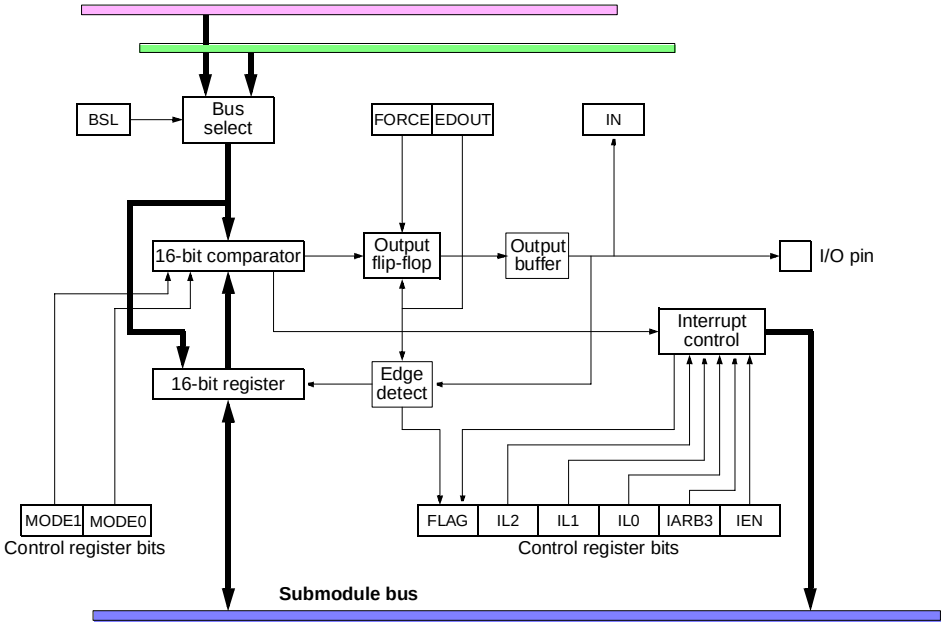


Figure 13-5 SASM Block Diagram (Channel A)

13.4.1 SASM Modes of Operation

Each SASM channel can operate in four different modes:

1. Input capture (IC) (i.e. either as input capture on a rising or falling edge or as a read-only input port)
2. Output compare (OC)
3. Output compare and toggle (OCT)
4. Output port (OP)

NOTE

For a channel operating in IC mode, the IN bit in the SIC register reflects the logic state of the corresponding input pin (after being Schmitt triggered and synchronized). When a channel is operating in OC, OCT or OP mode, the IN bit in the SIC register reflects the logic state of the output of the output flip-flop.

13.4.1.1 Clearing and Using the FLAG Bits

To clear a FLAG bit, the software must first read the channel's SIC register, then write a zero to the FLAG bit. These two steps do not have to be done on consecutive instructions. This clearing sequence must be used in every mode of operation. Writing a one to the FLAG bit has no effect.

WARNING

To avoid spurious interrupts, and to make sure that the FLAG bit is set according to the newly selected mode, the following sequence of operations should be adopted when changing mode:

1. Disable SASM interrupts
2. Change mode
3. Reset the corresponding FLAG bit
4. Re-enable SASM interrupts (if desired)

NOTE

When changing between output modes (OP, OC or OCT), it is not necessary to follow this procedure, as in these modes the FLAG bit merely indicates to the software that the compare value may be updated.

13.4.1.2 Input Capture (IC) Mode

In IC mode, the 16-bit counter value on the selected time base bus is 'captured' when a triggering event occurs on the channel's input pin. Triggering of the input capture circuitry is done by a rising or falling edge on the input pin; the polarity of the triggering edge is selected by the EDOUT bit. The logic level on the input pin can be read by software via the IN bit in the channel's SIC register.

In IC mode, the input pin is Schmitt triggered and the input signal is synchronized to the system clock (f_{SYS}). The IN bit reflects the state present on the input pin (after being Schmitt triggered and synchronized).

When an input capture occurs, the count value on the selected time base bus is latched into the channel's 16-bit data register. At the same time, the FLAG bit in the SIC register is set to indicate that an input capture has occurred.

The FLAG bit must be reset by software (see **13.4.1.1 Clearing and Using the FLAG Bits**). If the interrupt is serviced, the FLAG bit should be cleared by the servicing routine before returning from that routine. If a subsequent input capture event occurs while the FLAG bit is set, the new captured counter value is latched, and the FLAG bit remains unchanged.

In IC mode, the value of the EDOUT bit is permanently transferred to the output flip-flop. This value will be output on the pin when the mode is changed to one of the output modes.

13.4.1.3 Output Compare (OC) Mode

In OC mode, the state of an output pin is changed when a successful output compare occurs; an interrupt may also be generated. The output compare circuitry performs a comparison between the 16-bit register and the selected time base bus. When a match is found, the EDOUT bit value is transferred to the output flip-flop. At the same time,

the FLAG bit is set to indicate to the processor that a match has occurred. Depending on the state of the IEN bit, an interrupt can be generated when the FLAG bit is set. The FLAG bit must be reset by software (see **13.4.1.1 Clearing and Using the FLAG Bits**). If the interrupt is serviced, the FLAG bit should be cleared by the servicing routine before returning from that routine. If a subsequent output compare occurs while the FLAG bit is set, the output compare function occurs normally, and the FLAG bit remains set.

An output compare match can be simulated in software by writing a one to the FORCE bit. Setting the FORCE bit forces the EDOUT bit value onto the pin as if an output compare had occurred. In this case, the FLAG bit is not affected. Only if a genuine output compare occurs while doing a force, will the FLAG bit be set to signify that the compare has occurred.

In OC mode, the IN bit value reflects the logic state on the output of the output flip-flop.

13.4.1.4 Output Compare and Toggle (OCT) Mode

In OCT mode, the state of an output pin is toggled each time a successful output compare occurs; an interrupt may also be generated. The output compare circuitry performs a comparison between the 16-bit register and the selected time base bus. When a match is found, the output flip-flop is toggled to the opposite state. At the same time, the FLAG bit is set to indicate to the processor that the output compare has occurred. Depending on the state of the IEN bit, an interrupt can be generated when the FLAG bit is set. The FLAG bit must be reset by software (see **13.4.1.1 Clearing and Using the FLAG Bits**). If the interrupt is serviced, the FLAG bit should be cleared by the servicing routine before returning from that routine. If a subsequent output compare occurs while the FLAG bit is set, the output toggles, and the FLAG bit remains set.

An output compare match can be simulated in software by writing a one to the FORCE bit. Setting the FORCE bit forces the output flip flop to toggle as if an output compare had occurred. In this case, the FLAG bit is not affected. Only if a genuine output compare occurs while doing a force, will the FLAG bit be set to signify that the compare has occurred.

In OCT mode, the IN bit reflects the logic state on the output of the output flip-flop.

13.4.1.5 Output Port (OP) mode

In OP mode the channel's input/output pin is used as a single output port pin. The output compare function is still available, but for internal operation only, and does not affect the state of the output pin. An interrupt may also be generated when a compare occurs. The state of the output pin always reflects the value of the EDOUT bit in the channel's SIC register. Reading the EDOUT bit returns the last value written to it.

The internal compare feature compares the 16-bit register with the selected time base bus. The output compare circuitry performs a comparison between the 16-bit register and the selected time base bus. When a match is found, the FLAG bit is set to indicate to the processor that the output compare has occurred. Depending on the state of the IEN bit, an interrupt can be generated when the FLAG bit is set. The FLAG bit must be

reset by software (see **13.4.1.1 Clearing and Using the FLAG Bits**). If the interrupt is serviced, the FLAG bit should be cleared by the servicing routine before returning from that routine. If a subsequent output compare occurs while the FLAG bit is set, the internal output compare functions normally, and the FLAG bit remains set.

In OP mode, the IN bit value reflects the logic state on the output of the output flip-flop.

13.4.2 SASM Interrupts

Each channel in the dual-channel SASM has separately enabled and initiated interrupts and they each have their own unique vector number and address. However, they are both assigned to the same interrupt level and arbitration priority by the IL[2:0] and IARB3 bits in the SICA register.

A valid SASM interrupt is recognized when the FLAG bit is set, the corresponding IEN bit is set and the interrupt level defined by bits IL[2:0] is not equal to zero.

The FLAG bit is a status bit that indicates, when set, that an input capture or output compare has occurred on the corresponding single action channel.

The relative priority of these sources of interrupt is fixed and channel A has a higher priority than channel B.

13.4.3 Freeze Action on the SASM

When the IMB FREEZE signal is recognized, the SASM input capture and output compare functions are halted. As soon as the FREEZE signal is negated, SASM actions resume as if nothing had happened. During freeze, the IN bits of the SIC registers (SICA and SICB) are readable and return the levels present at the input pins if an input mode is in operation, or the output value if an output mode is in operation (see **13.4.4.1 SICA — SASM Status/Interrupt Control Register A** and **13.4.4.3 SICB — SASM Status/Interrupt Control Register B**). When one of the output modes is in operation, the force output function remains available, allowing the software to output the desired level (a useful feature for debugging). All SASM registers are accessible during freeze.

13.4.4 SASM Registers

The SASM register map comprises eight 16-bit register locations. As shown in **Table 13-6**, the register block contains two SASM registers for each channel and four reserved registers. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no meaning nor effect. All register addresses in this section are specified as offsets from the base address of the SASM. In CTM implementations featuring multiple SASMs, each SASM has its own set of registers.

Table 13-6 SASM Register Map

Address	15 8	7 0
0xYF F270	SASM14 status/interrupt/control register A (S14ICA)	
0xYF F272	SASM14 data register A (S14DATA)	
0xYF F274	SASM14 status/interrupt/control register B (S14ICB)	
0xYF F276	SASM14 data register B (S14DATB)	
0xYF F280	SASM16 status/interrupt/control register A (S16ICA)	
0xYF F282	SASM16 data register A (S16DATA)	
0xYF F284,	SASM16 status/interrupt/control register B (S16ICB)	
0xYF F286	SASM16 data register B (S16DATB)	
0xYF F290	SASM18 status/interrupt/control register A (S18ICA)	
0xYF F292	SASM18 data register A (S18DATA)	
0xYF F294	SASM18 status/interrupt/control register B (S18ICB)	
0xYF F296	SASM18 data register B (S18DATB)	
0xYF F2A0	SASM20 status/interrupt/control register A (S20ICA)	
0xYF F2A2	SASM20 data register A (S20DATA)	
0xYF F2A4	SASM20 status/interrupt/control register B (S20ICB)	
0xYF F2A6	SASM20 data register B (S20DATB)	

13.4.4.1 SICA — SASM Status/Interrupt Control Register A

This register contains the control, interrupt enable and status bits for SASM channel A. It also contains the interrupt priority level bits IL[2:0] and the arbitration priority bit IARB3 for the whole SASM (i.e. common to channels A and B).

S14ICA — SASM Status/Interrupt Control Register A **0xYF F270**
S16ICA **0xYF F280**
S18ICA **0xYF F290**
S20ICA **0xYF F2A0**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
FLAG	IL2	IL1	IL0	IARB3	IEN	0	BSL	IN	0	FORCE	EDOUT	0	0	MODE1	MODE0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13-7 SICA Bit Settings

Bit(s)	Name	Description
15	FLAG	Event flag. The FLAG bit is set whenever an input capture or output compare event occurs. This flag bit is set only by the hardware and cleared only by the software or by a system reset. If the IL field is non-zero, and the IEN bit is set, an interrupt request is generated when the FLAG bit is set. In IC mode, if a subsequent input capture event occurs while the FLAG bit is set, the new value is latched and the FLAG bit remains set. In OC mode, if a subsequent output compare event occurs while the FLAG bit is set, the compare occurs normally and the FLAG bit remains set. In OCT mode, if a subsequent output compare event occurs while the FLAG bit is set, the toggle of the output signal occurs as normal and the FLAG bit remains set. In OP mode, if a subsequent internal compare event occurs while the FLAG bit is set, the compare occurs normally and the FLAG bit remains set. To clear the flag, the software must first read the bit (as '1') then write a '0' to the bit. The flag clearing mechanism will work only if no flag setting event occurs between the read and write operations; if a FLAG setting event occurs between the read and write operations, the FLAG bit will not be cleared. 0 = An input capture or output compare event has not occurred. 1 = An input capture or output compare event has occurred.
14:12	IL[2:0]	Interrupt level. The three interrupt level bits are read/write control bits that select the priority level of interrupt requests made by the SASM. These bits can be read or written at any time and are cleared by reset. These bits affect both SASM channels, not just channel A. 000 = Interrupt disabled. 001 = Interrupt level 1 (lowest). 010 = Interrupt level 2. 011 = Interrupt level 3. 100 = Interrupt level 4. 101 = Interrupt level 5. 110 = Interrupt level 6. 111 = Interrupt level 7 (highest).
11	IARB3	Interrupt arbitration bit 3. The read/write IARB3 bit works in conjunction with the IARB[2:0] field in the BIUSM module configuration register. Each module that generates interrupt requests on the IMB must have a unique value in the arbitration field (IARB). This interrupt arbitration identification number is used to arbitrate for the IMB when modules generate simultaneous interrupts of the same priority. This bit affects both SASM channels, not just channel A.
10	IEN	Interrupt enable. This control bit enables interrupts on channel A when the FLAG bit is set and the IL[2:0] field is non-zero. This bit is cleared by reset 0 = Interrupts disabled. 1 = Interrupts enabled.
9	—	Reserved
8	BSL	Time base bus select. This control bit selects the time base bus to be connected to SASM channel A. This bit is cleared by reset. 0 = Time base bus A selected. 1 = Time base bus B selected.
7	IN	Input pin status. In input mode (IC), the IN bit reflects the logic state present on the corresponding input pin (after being Schmitt triggered and synchronized). In the output modes (OC, OCT and OP), the IN bit value reflects the state of the output of the output flip-flop. The IN bit is a read-only bit; writing to it has no effect. Reset has no effect on this bit.
6	—	Reserved
5	FORCE	Supervisor/user data space. The SUPV bit places the SCIM2E global registers in either supervisor or user data space. The FLAG bit is not affected by the use of the FORCE bit. 0 = No action 1 = Force output flip-flop to behave as if an output compare has just occurred.

Table 13-7 SICA Bit Settings (Continued)

Bit(s)	Name	Description
4	EDOUT	Edge detect and output level. In IC mode, the EDOUT bit is used to select the edge that will trigger the input capture circuitry. In OC mode, the EDOUT bit is used to latch the value to be output to the pin on the next output compare match or when the FORCE bit is set. Internal synchronization ensures that the correct level appears on the output pin when a new value is written to EDOUT and FORCE is set at the same time. Reading EDOUT returns the previous value written. In OCT mode, the EDOUT bit has no effect. However, the force function is still available and will force the value of the EDOUT bit to appear on the output pin. In OP mode, the value of the EDOUT bit is output to the corresponding pin. Reading EDOUT returns the previous value written. 0 = Input capture on falling edge. 1 = Input capture on rising edge.
3:2	—	Reserved
1:0	MODE1, MODE0	SASM operating mode select. These control bits select the mode of operation of the SASM channel, as shown in the following table. MODE1 and MODE0 are cleared by reset. 00 = Input capture (IC). 01 = Output port (OP). 10 = Output compare (OC 11 = Output compare and toggle (OCT)

13.4.4.2 SDATA — SASM Data Register A

SDATA is the 16-bit read-write register associated with channel A. In IC mode, SDATA contains the last captured value. In the OC, OCT and OP modes, it is loaded with the value of the next output compare. SDATA is not affected by reset.

S14DATA — SASM Data Register A **0xYF F272,**
S16DATA **0xYF F282**
S18DATA **0xYF F292**
S20DATA **0xYF F2A2**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

13.4.4.3 SICB — SASM Status/Interrupt Control Register B

This register contains the control and status bits for SASM channel B. The bits it contains are identical to those in SICA, with the exception of the IL[2:0], IARB3 and IEN which apply to both channels simultaneously and which are included only in SICA. For descriptions of the bits, please refer to **13.4.4.1 SICA — SASM Status/Interrupt Control Register A**).

S14ICB — SASM Status/Interrupt Control Register B **0xYF F274**
S16ICB **0xYF F284**
S18ICB **0xYF F294**
S20ICB **0xYF F2A4**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
FLAG	0	0	0	0	0	0	BSL	IN	0	FORCE	EDOUT	0	0	MODE1	MODE0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.4.4 SDATB — SASM Data Register B

SDATB is the 16-bit read-write register associated with channel A. In the IC mode, SDATB contains the last captured value. In the OC, OCT and OP modes, it is loaded with the value of the next output compare. SDATB is not affected by reset.

S14DATB — SASM Data Register B **0xYF F276**
S16DATB **0xYF F286**
S18DATB **0xYF F296**
S20DATB **0xYF F2A6**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

13.5 Double-Action Submodule (DASM)

The double-action submodule (DASM) provides two 16-bit input captures or two 16-bit output compare functions that can occur automatically without software intervention. The input edge detector is programmable to cause the capture function to occur on the desired edges. The output flip-flop is set by one of the output compares and is reset by the other one. In either the input capture modes or the output compare modes, an optional interrupt is available to the software. Software selection is provided for which of two incoming time-base buses is used for input captures or output compares.

The DASM can work in six different modes: disable mode, pulse length measurement, period measurement, input capture mode, single pulse generation, and continuous pulse width generation.

The DASM has three data registers that are accessible to the software from the various modes. For some of the modes, two of the registers are cascaded together to provide double buffering. The value in one register is transferred to another register automatically at the correct time so that the minimum pulse (measurement or generation) is just one time-base bus count. See **Figure 13-6** for a DASM block diagram.

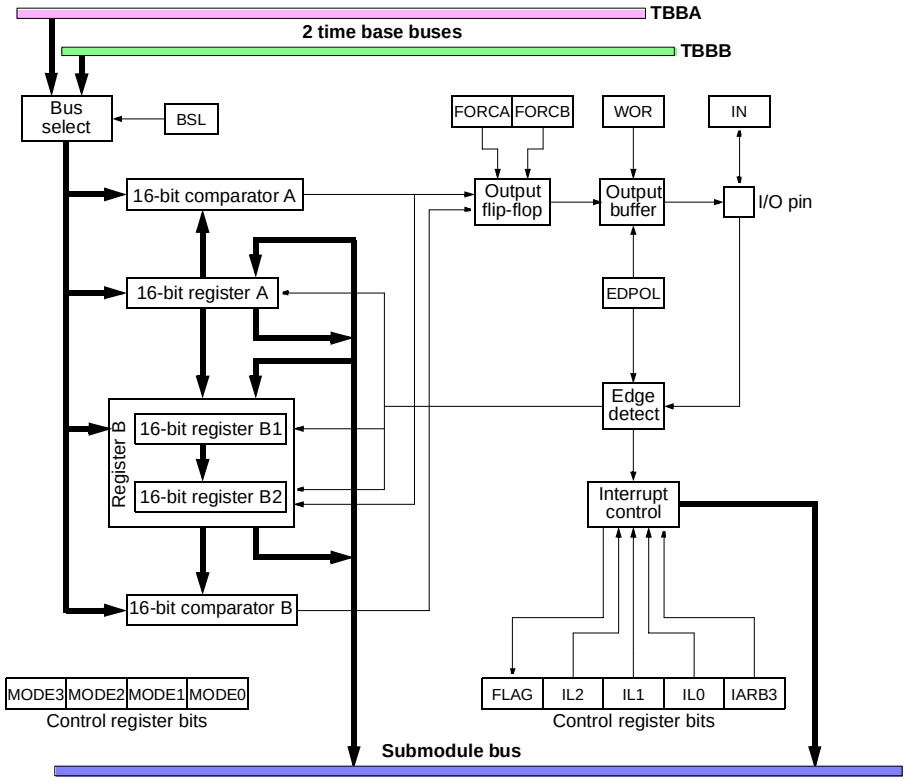


Figure 13-6 DASM Block Diagram

Channel A comprises one 16-bit data register and one 16-bit comparator. Channel B also appears to the user to consist of one 16-bit data register and one 16-bit comparator, however, internally, channel B has two data registers B1 and B2, and the operating mode determines which register is accessed by the software:

- In the input capture modes (IPWM, IPM and IC), registers A and B2 are used to hold the captured values; in these modes, the B1 register is used as a temporary latch for channel B.
- In the output compare modes (OCA and OCAB), registers A and B2 are used to define the output pulse; register B1 is not used in these modes.
- In the output pulse width modulation mode (OPWM), registers A and B1 are used as primary registers and hidden register B2 is used as a double buffer for channel B.

Register contents are always transferred automatically at the correct time so that the minimum pulse (measurement or generation) is just one time base bus count. The A and B data registers are always read/write registers, accessible via the CTM's submodule bus.

In the input capture modes, the edge detect circuitry triggers a capture whenever a rising or falling edge (as defined by the EDPOL bit) is applied to the input pin. The signal on the input pin is Schmitt triggered and synchronized with the system clock (f_{SYS}).

In the disabled mode (DIS) and in the input modes, the IN bit reflects the state present on the input pin (after being Schmitt triggered and synchronized). In the output modes the IN bit reflects the value present at the output of the output flip-flop. The output flip-flop is used in output modes to hold the logic level applied to the output pin.

The time base bus selector is common to all input and output functions; it connects the DASM to time base bus A or B and is controlled in software by the bus select bit BSL in the DASMSIC register.

13.5.1 32-Bit Coherent Access

In the IPWM and IPM modes, 32-bit coherent access of the data registers is supported. A 32-bit coherent access consists of doing a long word aligned access of data register A. In this case, register A is accessed first, immediately followed (on the next cycle) by a register B access. During this time, any flag setting or data transfer from the hidden B register is deferred until coherent access has ended. When the 32-bit access has ended, the DASM finishes any pending B action and resumes normal operation.

13.5.2 DASM Modes of Operation

The mode of operation of the DASM is determined by the mode select bits MODE[3:0] in the DASMSIC register (see **Table 13-8**).

Table 13-8 DASM Modes of Operation

MODE[3:0]	Mode	Description of mode
0000	DIS	Disabled — Input pin is high impedance; IN gives state of the input pin.
0001	IPWM	Input pulse width measurement — Capture on the leading edge and the trailing edge of an input pulse.
0010	IPM	Input period measurement — Capture two consecutive rising/falling edges.
0011	IC	Input capture — Capture when the designated edge is detected.
0100	OCB	Output compare, flag set on B compare — Generate leading and trailing edges of an output pulse and set the flag.
0101	OCAB	Output compare, flag on A and B compare — Generate leading and trailing edges of an output pulse and set the flag.
1xxx	OPWM	Output pulse width modulation — Generate continuous PWM output with 7, 9, 11, 12, 13, 14, 15 or 16 bits of resolution.

WARNING

To avoid spurious interrupts, and to make sure that the FLAG bit is set according to the newly selected mode, the following sequence of operations should be adopted when changing mode:

1. Disable DASM interrupts
2. Change mode
3. Reset the corresponding FLAG bit

4. Re-enable DASM interrupts (if desired)

NOTE

When changing between output modes (OP, OC or OCT), it is not necessary to follow this procedure, as in these modes the FLAG bit merely indicates to the software that the compare value can be updated.

13.5.2.1 Disable (DIS) mode

DIS mode is selected by making MODE[3:0] = 0000.

In this mode, all input capture and output compare functions of the DASM are disabled and the FLAG bit is maintained in its reset state, but the input port pin function remains available. The associated pin becomes a high impedance input and the input level on this pin is reflected by the state of the IN bit in the DASMSIC register. All control and interrupt bits remain accessible, allowing the software to prepare for future mode selection. Data registers A and B are accessible at consecutive addresses. Writing to data register B stores the same value in registers B1 and B2.

WARNING

When changing modes, it is imperative to go through the DIS mode in order to reset the DASM's internal functions properly. Failure to do this could lead to invalid and unexpected output compare or input capture results, and to flags being set incorrectly.

13.5.2.2 Input Pulse Width Measurement (IPWM) Mode

IPWM mode is selected by making MODE[3:0] = 0001.

This mode allows the width of a positive or negative pulse to be determined by capturing the leading edge of the pulse on channel B and the trailing edge of the pulse on channel A; successive captures are done on consecutive edges of opposite polarity. The edge sensitivity is selected by the EDPOL bit in the DASMSIC register.

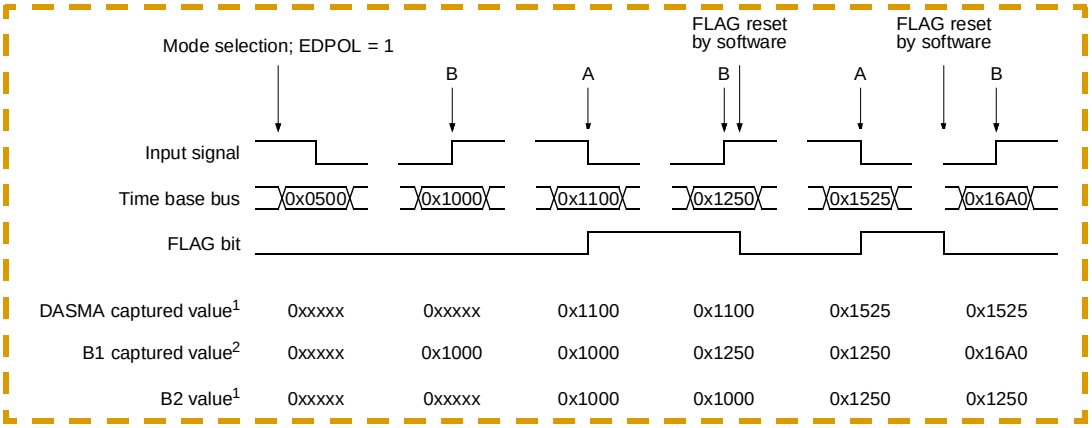
This mode also allows the software to determine the logic level on the input pin at any time by reading the IN bit in the DASMSIC register.

The channel A input capture function remains disabled until the first rising edge triggers the first input capture on channel B. When this rising edge is detected, the count value of the time base bus selected by the BSL bit is latched in the 16-bit data register B1; the FLAG bit is not affected. When the next falling edge is detected, the count value of the time base bus is latched into the 16-bit data register A and, at the same time, the FLAG bit is set and the contents of register B1 are transferred to register B2. Reading data register B returns the value in register B2. If subsequent input capture events occur while the FLAG bit is set, data registers A and B will be updated with the latest captured values and the FLAG bit will remain set.

If a 32-bit coherent operation is in progress when the falling edge is detected, the transfer from B1 to B2 is deferred until the coherent operation is completed. Operation of the DASM then continues on channels B and A as previously described.

The input pulse width is calculated by subtracting the value in data register B from the value in data register A.

Figure 13-7 provides an example of how the DASM can be used for input pulse width measurement.



Notes: 1. These values are accessible to the software.
2. These values are internal and are not accessible.

Figure 13-7 Input Pulse Width Measurement Example

13.5.2.3 Input Period Measurement (IPM) Mode

IPM mode is selected by making MODE[3:0] = 0010.

This mode allows the period of an input signal to be determined by capturing two consecutive rising edges or two consecutive falling edges; successive input captures are done on consecutive edges of the same polarity. The edge polarity is defined by the EDPOL bit in the DASMSIC register.

This mode also allows the software to determine the logic level on the input pin at any time by reading the IN bit in the DASMSIC register.

When the first edge having the selected polarity is detected, the time base bus value is latched into the 16-bit data register A, the data in register B1 is transferred to data register B2 and finally the data in register A is transferred to register B1. On this first capture the FLAG bit is not set. On the second and subsequent captures, the FLAG bit is set immediately before the data in register A is transferred to register B1.

When the second edge of the same polarity is detected, the time base bus value is latched into data register A, the data in register B1 is transferred to data register B2, the FLAG bit is set to signify that the beginning and end points of a complete period have been captured, and finally data register A is transferred to register B1. This

sequence of events is repeated for each subsequent capture. Reading data register B returns the value in register B2.

If a 32-bit coherent operation is in progress when an edge is detected, the transfer of data from B1 to B2 is deferred until the coherent operation is completed. At any time, the input level present on the input pin can be read on the IN bit.

The input pulse period is calculated by subtracting the value in data register B from the value in data register A.

Figure 13-8 provides an example of how the DASM can be used for input period measurement.

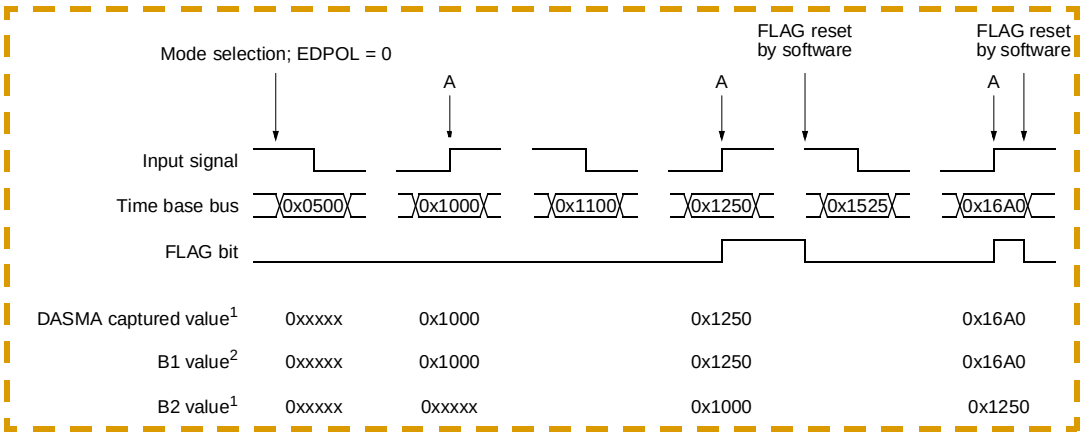


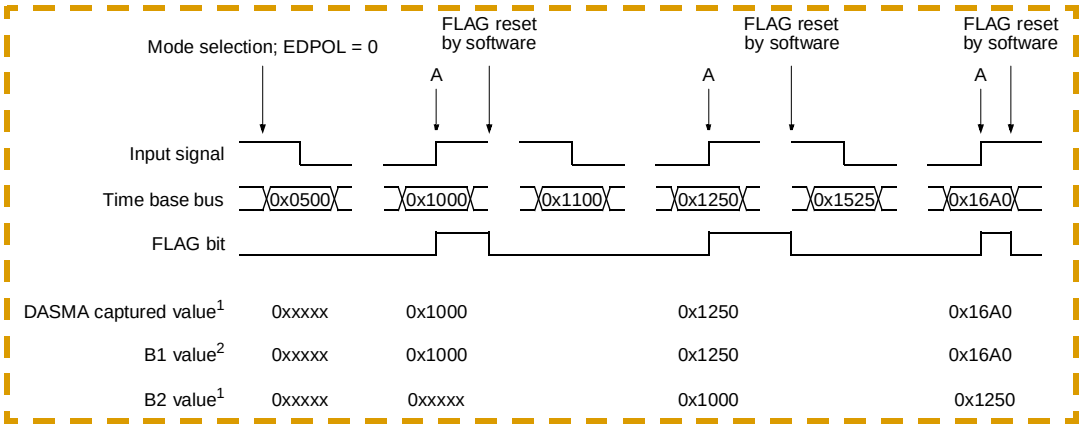
Figure 13-8 Input Period Measurement Example

13.5.2.4 Input Capture (IC) Mode

IC mode is selected by making MODE[3:0] = 0011.

This mode is identical to the input period measurement mode (IPM) described above, with the exception that the FLAG bit is also set at the occurrence of the first detected edge of the selected polarity. In this mode the DASM functions as a standard input capture function in a similar way to the M68HC11 family timers. In this case the value latched in channel B can be ignored.

Figure 13-9 provides an example of how the DASM can be used for input capture.



Notes: 1. These values are accessible to the software.
2. These values are internal and are not accessible.

Figure 13-9 DASM Input Capture Example

13.5.2.5 Output Compare (OCB and OCAB) Modes

OC mode is selected by making MODE[3:0] = 010x. The MODE0 bit controls the setting criteria for the FLAG bit, i.e. when a compare occurs only on channel B or when a compare occurs on either channel (see **13.5.5.1 DASMSIC — DASM Status/Interrupt Control Register**).

This mode allows the DASM to perform four different output functions:

- Single-shot output pulse (two edges), with FLAG set on the second edge.
- Single-shot output pulse (two edges), with FLAG set on both edges.
- Single-shot output transition (one edge).
- Output port pin, with output compare function disabled.

In this mode the leading and trailing edges of variable width output pulses are generated by calculated output compare events occurring on channels A and B, respectively. OC mode may also be used to perform a single output compare function, similar to the M68HC11 timer, or may be used as an output port bit.

In this mode, channel B is accessed via register B2. Register B1 is not used and is not accessible to the user. Both channels work together to generate one ‘single shot’ output pulse signal. Channel A defines the leading edge of the output pulse, while channel B defines the trailing edge of the pulse. FLAG setting can be done when a compare occurs on channel B only or when a compare occurs on either channel (as defined by the MODE0 bit in the DASMSIC register).

When this mode is first selected, both comparators are disabled. Each comparator is enabled by writing to its data register; it remains enabled until the next successful comparison is made on that channel, whereupon it is disabled. The values stored in registers A and B are compared with the count value on the selected time base bus when their corresponding comparators are enabled.

The output flip-flop is set when a match occurs on channel A. The output flip-flop is reset when a match occurs on channel B. The polarity of the output signal is selected by the EDPOL bit. The output flip-flop level can be obtained at any time by reading the IN bit.

If subsequent enabled output compares occur on channels A and B, the output pulses continue to be output, regardless of the state of the FLAG bit.

At any time, the FORCA and FORCB bits allow the software to force the output flip-flop to the level corresponding to a comparison on channel A or B, respectively. Note that the FLAG bit is not affected by these 'force' operations.

Totem pole or open-drain output circuit configurations can be selected using the WOR bit in the DASMSIC register.

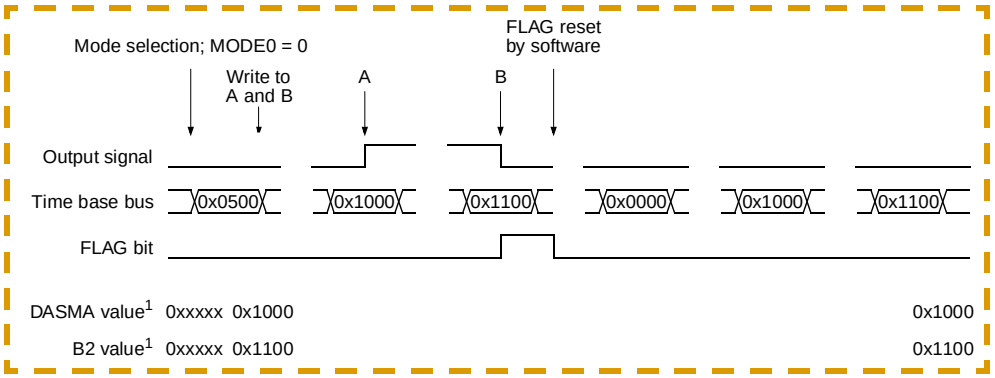
WARNING

There is no hardware protection to disable comparator B while comparator A is enabled. It is the user's responsibility to load data registers A and B with the values needed to produce the desired output pulse.

NOTE

If both channels are loaded with the same value they will try to force different levels on the output flip-flop. Hardware protection circuitry ensures that no contention occurs and the output flip-flop provides a logic zero level output.

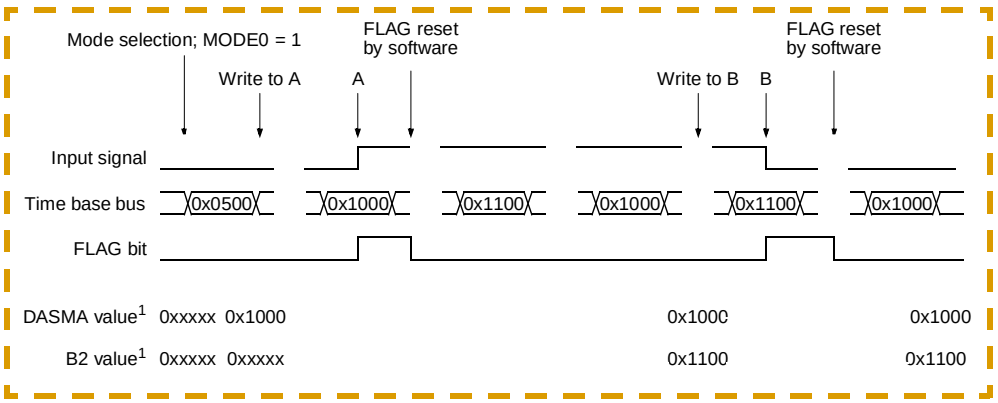
Single Shot Output Pulse Operation — The single shot output pulse operation is selected by writing the leading edge value of the desired pulse to data register A and the trailing edge value to data register B. A single pulse will be output at the desired time, thereby disabling the comparators until new values are written to the data registers. In this mode, registers A and B2 are accessible to the user software (at consecutive addresses). **Figure 13-10** provides an example of how the DASM can be used to generate a single output pulse.



Note: 1. These values are accessible to the software.

Figure 13-10 Single Shot Output Pulse Example

Single Output Compare Operation — The single output compare operation is selected by writing to only one of the two data registers (A or B), thus enabling only one of the comparators. Following the first successful match on the enabled channel, the output level is fixed and remains at the same level indefinitely with no further software intervention being required. In this mode, registers A and B2 are accessible to the user software (at consecutive addresses). **Figure 13-11** provides an example of how the DASM can be used to perform a single output compare.



Note: 1. These values are accessible to the software.

Figure 13-11 Single Shot Output Transition Example

Output Port Bit Operation — The output port bit operation is selected by leaving both channels disabled, i.e. by writing to neither register A nor B. The EDPOL bit alone controls the output value. The same result can be achieved by keeping EDPOL at zero and using the FORCA and FORCB bits to obtain the desired output level.

13.5.2.6 Output Pulse Width Modulation (OPWM) Mode

OPWM mode is selected by making MODE[3:0] = 1xxx. The MODE[2:0] bits allow some of the comparator bits to be masked.

This mode allows pulse width modulated output waveforms to be generated, with eight selectable frequencies (for a given time base). Both channels (A and B) are used to generate one PWM output signal on the DASM pin.

Channel B is accessed via register B1. Register B2 is not accessible to the user. Channels A and B define the leading and trailing edges, respectively, of the PWM output pulse. The value in register B1 is continuously transferred to register B2 in the time between each trailing edge and the following leading edge.

The value loaded in register A is continuously compared with the value on the time base bus. When a match on A occurs, the FLAG bit is set and the output flip-flop is set. The value loaded in register B2 is continually compared with the value on the time base bus. When a match occurs on B, the output flip-flop is reset.

The polarity of the PWM output signal is selected by the EDPOL bit. The output flip-flop level can be obtained at any time by reading the IN bit.

If subsequent compares occur on channels A and B, the PWM pulses continue to be output, regardless of the state of the FLAG bit.

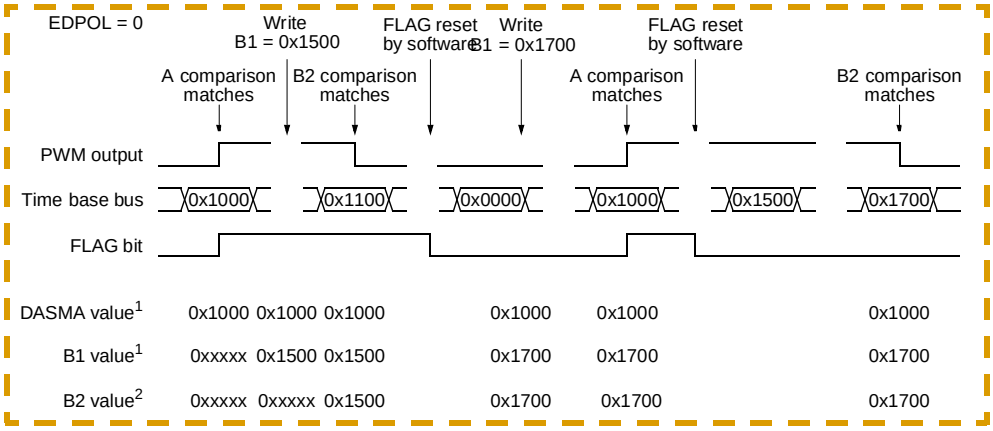
At any time, the FORCA and FORCB bits allow the software to force the output flip-flop to the level corresponding to comparison on A or B respectively. Note that the FLAG bit is not affected by the FORCA and FORCB operations.

WARNING

There is no hardware protection to disable comparator B while comparator A is enabled. It is the user's responsibility to load data registers A and B with the values needed to produce the desired PWM output pulse.

If both channels are loaded with the same value they will try to force different levels on the output flip-flop. Hardware protection circuitry ensures that no contention occurs and the output flip-flop provides a logic zero level output.

Figure 13-12 provides an example of how the DASM can be used for pulse width modulation.



Notes: 1. These values are accessible to the software.
2. These values are internal and are not accessible.

Figure 13-12 DASM Output Pulse Width Modulation Example

To generate PWM output pulses of different frequencies, the 16-bit comparator can have some of its bits masked. This is controlled by bits MODE2, MODE1 and MODE0. The frequency of the PWM output (f_{PWM}) is given by Equation 1 (assuming the DASM is connected to a free running counter):

$$f_{PWM} = \frac{f_{SYS}}{N_{CPSM} \bullet N_{DASM}}$$

where N_{CPSM} is the overall CPSM clock divide ratio (÷2 to ÷512 or ÷3 to ÷768) and N_{DASM} is the DASM divide ratio.

A few examples of frequencies and resolutions that can be obtained are shown in **Table 13-9**.

Table 13-9 DASM PWM Example Output Frequencies/Resolutions at f_{SYS} = 16 MHz

N_{CPSM}	N_{DASM}[1]	PWM output frequency (Hz)	Resolution (bits)
512	65536	0.48	16
2	65536	122.07	16
512	32768	0.95	15
2	32768	244.14	15
512	16384	1.91	14
2	16384	488.28	14
512	8192	3.81	13
2	8192	976.56	13
512	4096	7.63	12
2	4096	1953.13	12
512	2048	15.26	11
2	2048	3906.25	11
512	512	31.04	9
2	512	15625.00	9
512	128	244.14	7
2	128	62500.00	7

NOTES:
1. This table is valid only if the DASM is connected to a free-running counter.

When using 16 bits of resolution on the comparator (MODE[2:0] = 000), the output can vary from a 0% duty cycle up to a duty cycle of 65535/65536. In this case it is not possible to have a 100% duty cycle. In cases where 16-bit resolution is not needed, it is possible to have a duty cycle ranging from 0% to 100%. Setting bit 15 of the value stored in register B to ‘1’ results in the output being ‘always set’. Clearing bit 15 (to ‘0’) allows normal comparisons to occur and the normal output waveform is obtained. Changes to and from the 100% duty cycle are done synchronously, as are all other width changes.

In the OPWM mode, the WOR bit selects whether the output is totem pole driven or open-drain.

13.5.3 DASM interrupts

When the FLAG bit is set, an interrupt request is generated on one of eight levels as defined by the interrupt level bits (IL[2:0]) in the DASMSIC register. If the interrupt level is set to zero, interrupts are disabled.

13.5.4 Freeze Action on the DASM

When the IMB FREEZE signal is recognized, the DASM capture and compare functions are halted. As soon as the FREEZE signal is negated, DASM actions resume as if nothing had happened. During freeze, the IN bit of the DASMSIC register is readable

and returns the level present at the input pin if an input mode is selected, or the output value if an output mode is in operation. When one of the output modes is in operation, the force output function remains available, allowing the software to output the desired level and simplifying debugging. All DASM registers are accessible during freeze.

13.5.5 DASM Registers

The DASM register map comprises four 16-bit register locations. As shown in **Table 13-10**, the register block contains three DASM registers and one reserved register. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no meaning or effect. All register addresses in this section are specified as offsets from the base address of the DASM. In CTM implementations featuring multiple DASMs, each DASM has its own set of registers.

Table 13-10 DASM Register Map

Address	15 8	7 0
0xYF F218	DASM3 status/interrupt/control register (DASM3SIC)	
0xYF F21A	DASM3 register A (DASM3A)	
0xYF F21C	DASM3 register B (DASM3B)	
0xYF F220	DASM4 status/interrupt/control register (DASM4SIC)	
0xYF F222	DASM4 register A (DASM4A)	
0xYF F224	DASM4 register B (DASM4B)	
0xYF F248	DASM9 status/interrupt/control register (DASM9SIC)	
0xYF F24A	DASM9 register A (DASM9A)	
0xYF F24C	DASM9 register B (DASM9)	
0xYF F250	DASM10 status/interrupt/control register (DASM10SIC)	
0xYF F252	DASM10 register A (DASM10A)	
0xYF F254	DASM10 register B (DASM10)	

13.5.5.1 DASMSIC — DASM Status/Interrupt Control Register

DASM3SIC — DASM Status/Interrupt Control Register **0xYF F218**
DASM4SIC **0xYF F220**
DASM9SIC **0xYF F248**
DASM10SIC **0xYF F250**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
FLAG	IL2	IL1	IL0	IARB3	0	WOR	BSL	IN	FORCA	FORCB	ED-POL	MODE3	MODE2	MODE1	MODE0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13-11 DASMSIC Bit Settings

Bit(s)	Name	Description
15	FLAG	Flag status. This status bit indicates whether or not an input capture or output compare event has occurred. If the IL field is non-zero, an interrupt request is generated when the FLAG bit is set. The flag clearing mechanism will work only if no flag setting event occurs between the read and write operations; if a FLAG setting event occurs between the read and write operations, the FLAG bit will not be cleared. – In the DIS mode, the FLAG bit is cleared. – In the IPWM mode, the FLAG bit is set each time there is a capture on channel A. – In the IPM mode, the FLAG bit is set each time there is a capture on channel A, except for the first time. – In the IC mode, the FLAG bit is set each time there is a capture on channel A. – In the OCB mode (i.e. when MODE0 = 0), the FLAG bit is only set each time there is a successful comparison on channel B. In the OCAB mode (i.e. when MODE0 = 1), the FLAG bit is set each time there is a successful comparison on either channel A or B. – In the OPWM mode, the FLAG bit is set whenever there is a successful comparison on channel A. 0 = An input capture or output compare event has not occurred. 1 = An input capture or output compare event has occurred.
14:12	IL[2:0]	Interrupt level. The three interrupt level bits are read/write control bits that select the priority level of interrupt requests made by the DASM. These bits can be read or written at any time and are cleared by reset. 000 = Interrupt disabled 001 = Interrupt level 1 (lowest) 010 = Interrupt level 2 011 = Interrupt level 3 100 = Interrupt level 4 101 = Interrupt level 5 110 = Interrupt level 6 111 = Interrupt level 7 (highest)
11	IARB3	Interrupt arbitration bit 3. The read/write IARB3 bit works in conjunction with the IARB[2:0] field in the BIUSM module configuration register. Each module that generates interrupt requests on the IMB must have a unique value in the arbitration field (IARB). This interrupt arbitration identification number is used to arbitrate for the IMB when modules generate simultaneous interrupts of the same priority.
10	—	Reserved
9	WOR	Wired-OR. In the DIS, IPWM, IPM and IC modes, the WOR bit is not used; reading this bit returns the value that was previously written. In the OCB, OCAB and OPWM modes, the WOR bit selects whether the output buffer is configured for open-drain or totem pole operation. 0 = Output buffer is totem pole. 1 = Output buffer is open-drain.
8	BSL	Bus select. This control bit selects the time base bus to be connected to the DASM. 0 = The DASM is connected to time base bus A. 1 = The DASM is connected to time base bus B.
7	IN	Input pin status. In the DIS, IPWM, IPM and IC modes, this read-only status bit reflects the logic level on the input pin. In the OCB, OCAB and OPWM modes, reading this bit returns the value latched on the output flip-flop, after EDPOL polarity selection. Writing to this bit has no effect.
6:5	FORCA, FORCB	Force A, B. In the OCB, OCAB and OPWM modes, the FORCA, B bit allows the software to force the output flip-flop to behave as if a successful comparison had occurred on channel A, B (except that the FLAG bit is not set). Writing a one to FORCA, B sets the output flip-flop; writing a zero to it has no effect. In the DIS, IPWM, IPM and IC modes, FORCA and FORCB are not used and writing to them has no effect. Writing a one to both FORCA and FORCB simultaneously resets the output flip-flop

Table 13-11 DASMSIC Bit Settings (Continued)

Bit(s)	Name	Description
4	EDPOL	Edge polarity. In the DIS mode, this bit is not used; reading it returns the last value written. In the IPWM mode, this bit is used to select the capture edge sensitivity of channels A and B. 0 = Channel A captures on a rising edge. Channel B captures on a falling edge. 1 = Channel A captures on a falling edge. Channel B captures on a rising edge. In the IPM and IC modes, the EDPOL bit is used to select the input capture edge sensitivity of channel A. 0 = Channel A captures on a rising edge. 1 = Channel A captures on a rising edge. In the OCB, OCAB and OPWM modes, the EDPOL bit is used to select the voltage level on the output pin. 0 = The output flip-flop logic level appears on the output pin: a compare on channel A sets the output pin, a compare on channel B resets the output pin. 1 = The complement of the output flip-flop logic level appears on the output pin: a compare on channel A resets the output pin; a compare on channel B sets the output pin.
3:0	MODE[3:0]	Mode select. The four mode select bits select the mode of operation of the DASM. To avoid spurious interrupts, it is recommended that DASM interrupts are disabled before changing the operating mode.

13.5.5.2 DASMA — DASM Data Register A

DASM3A — DASM Data Register A **0xYF F21A**
DASM4A **0xYF F222**
DASM9A **0xYF F24A**
DASM10A **0xYF F252**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DASMA is the data register associated with channel A; its use varies with the different modes of operation:

- In the DIS mode, DASMA can be accessed to prepare a value for a subsequent mode selection.
- In the IPWM mode, DASMA contains the captured value corresponding to the trailing edge of the measured pulse.
- In the IPM and IC modes, DASMA contains the captured value corresponding to the most recently detected dedicated edge (rising or falling edge).
- In the OCB and OCAB modes, DASMA is loaded with the value corresponding to the leading edge of the pulse to be generated. Writing to DASMA in the OCB and OCAB modes also enables the corresponding channel A comparator until the next successful comparison.
- In the OPWM mode, DASMA is loaded with the value corresponding to the leading edge of the PWM pulse to be generated.

13.5.5.3 DASMB — DASM Data Register B

DASM3B — DASM Data Register B **0xYF F21C**
DASM4B **0xYF F224**
DASM9B **0xYF F24C**
DASM10B **0xYF F254**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DASMB is the data register associated with channel B; its use varies with the different modes of operation. Depending on the mode selected, software access is to register B1 or register B2.

In the DIS mode, DASMB can be accessed to prepare a value for a subsequent mode selection. In this mode, register B1 is accessed in order to prepare a value for the OPWM mode. Unused register B2 is hidden and cannot be read, but is written with the same value when register B1 is written.

In the IPWM mode, DASMB contains the captured value corresponding to the leading edge of the measured pulse. In this mode, register B2 is accessed; buffer register B1 is hidden and cannot be accessed.

In the IPM and IC modes, DASMB contains the captured value corresponding to the most recently detected period edge (rising or falling edge). In this mode, register B2 is accessed; buffer register B1 is hidden and cannot be accessed.

In the OCB and OCAB modes, DASMB is loaded with the value corresponding to the trailing edge of the pulse to be generated. Writing to DASMB in the OCB and OCAB modes also enables the corresponding channel B comparator until the next successful comparison. In this mode, register B2 is accessed; buffer register B1 is hidden and cannot be accessed.

In the OPWM mode, DASMB is loaded with the value corresponding to the trailing edge of the PWM pulse to be generated. In this mode, register B1 is accessed; buffer register B2 is hidden and cannot be accessed.

13.6 Pulse Width Modulation Submodule (PWMSM)

The purpose of the pulse width modulation submodule (PWMSM) is to create a variable pulse width output signal at a wide range of frequencies, independent of other CTM9 output signals. The PWMSM includes its own counter, and thus does not use the CTM9 time-base buses. The PWMSM pulse width can vary from 0.0 percent to 100.0 percent, with up to 16 bits of resolution. The finest output resolution is the MCU system clock time divided by two (for a system clock of 16.78 MHz, the finest output pulse width resolution is 119 nanoseconds). With the full 16 bits of resolution and the first stage prescaler divide-by-2 clock selection, the period of the PWM output can

range from 7.8 milliseconds to 2.0 seconds (assuming a 16.78 MHz MCU clock). By reducing the counting value, the output signal period can be reduced. The period can be as fast as 488 microseconds (2.048 KHz) with 12 bits of resolution, as fast as 30.5 microseconds (32.768 KHz) with 8 bits of resolution, and as fast as 7.6 microseconds (131.072 KHz) with 6 bits of resolution (still assuming a 16.78 MHz system clock and a first stage prescaler divide-by-2 clock selection). A block diagram of the PWMSM is shown in **Figure 13-13**.

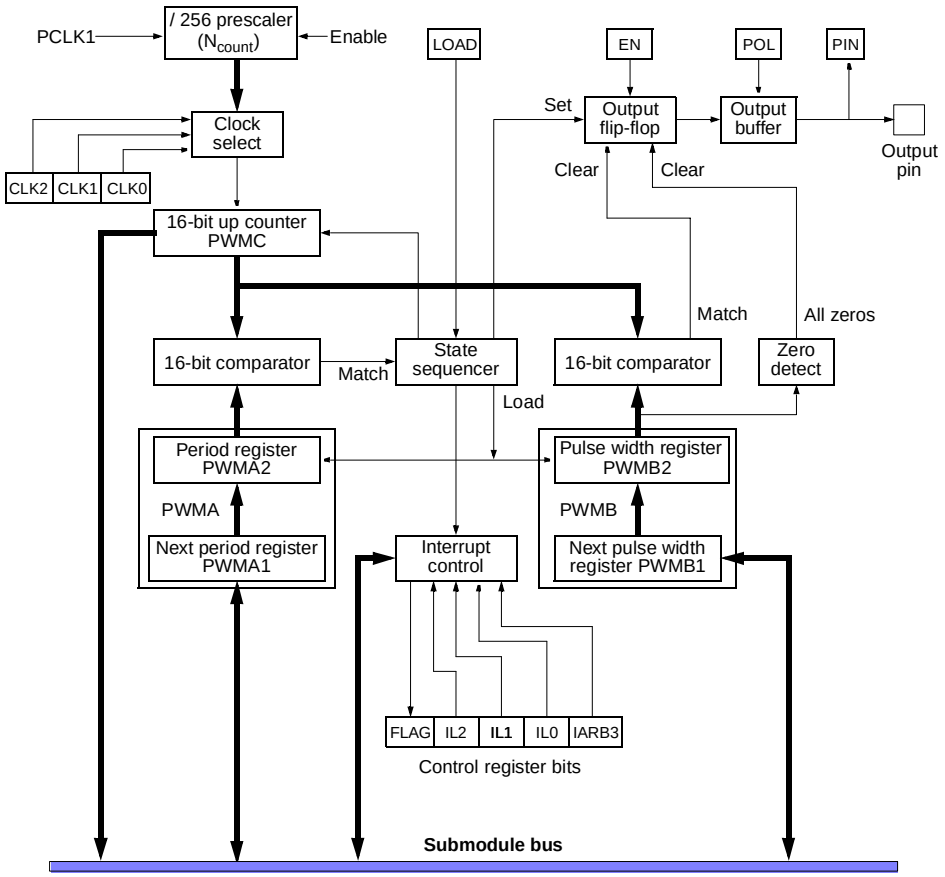


Figure 13-13 Pulse Width Modulation Submodule Block Diagram

13.6.1 Output Flip-Flop and Pin

The output flip-flop is the basic output mechanism of the PWMSM. Except when the required pulse width is 0% or 100%, the output flip-flop is set at the beginning of each period and is cleared at the end of the designated pulse width. The polarity of the output pulse can be selected in software. The output of the PWMSM is connected to an external, output-only pin. When the PWMSM is not required, and is disabled by clearing the EN bit in the PWMSIC register, this pin serves as a digital output-only port pin. When the PWMSM is disabled, the POL bit in the SIC register serves as an output port bit.

13.7 Time Base Bus System

The time base bus (TBB) system makes it possible to freely configure connections between counter submodules and action submodules. However the PWMSM submodules are independent of the time base bus system. The CTM9 configuration is shown in **Figure 13-1**.

13.7.1 Clock Selection

The PWMSM contains an 8-bit prescaler that is clocked by the PCLK1 signal from the CPSM (i.e. the MCU system clock divided by 2 or by 3). A 3-bit field (CLK[2:0]) in the PWMSM status, interrupt and control register (PWMSIC) allows the software to select which of the 8 prescaler outputs drives the PWMSM counter. The prescaler outputs are the main MCU clock divided by: 2, 4, 8, 16, 32, 64, 128 and 512 (or 3, 6, 12, 24, 48, 96, 192 and 768, if the divide-by-3 option is used in the CPSM to generate PCLK1).

13.7.2 The PWMSM Counter (PWMC)

The 16-bit up-counter in the PWMSM provides the time base for the PWM output signal. The counter is held in the 0x0001 state on reset or when the PWMSM is disabled. When the PWMSM is enabled, the counter begins counting at the rate defined by the clock selection. Each time the counter matches the contents of the period register, the counter is preset to 0x0001 and starts to count from that value. The counter can be read at any time without affecting its value. Writing to the counter has no effect.

13.7.3 PWMSM Period Registers and Comparator

The period section of the PWMSM consists of two 16-bit period registers (PWMA1 and PWMA2) and one 16-bit comparator. PWMA2 holds the current PWM period value and PWMA1 holds the next PWM period value. The software establishes the next period of the output PWM signal by writing a value into PWMA1. PWMA2 acts as a double buffer of PWMA1, allowing the contents of PWMA1 to be changed at any time without affecting the current period of the output signal; it cannot be accessed directly by the software. PWMA1 can be read or written at any time. The new value in the PWMA1 register is transferred to PWMA2 on the next full cycle of the output or when a ‘1’ is written to the LOAD bit in the PWMSIC register.

The comparator continuously compares the contents of the PWMA2 register with the value in the PWMSM counter. When a match occurs, the state sequencer sets the output flip-flop and resets the counter to 0x0001.

Period values 0x0000 and 0x0001 are special cases. When PWMA2 contains 0x0000, an output period of 65536 PWM clock periods is generated.

When PWMA2 contains 0x0001, a period match occurs on every PWM clock period: the counter never increments beyond 0x0001 and the output level never changes.

NOTE

A value of 0x0002 in the period register and a value of 0x0001 in the pulse register are the conditions necessary to obtain the maximum possible output frequency for a given PWM clock period.

13.7.4 PWMSM Pulse Width Registers and Comparator

The pulse width section of the PWMSM consists of two 16-bit pulse width registers (PWMB1 and PWMB2) and one 16-bit comparator. PWMB2 holds the current PWM pulse width value and PWMB1 holds the next PWM pulse width value. The software establishes the next pulse width of the output PWM signal by writing a value into PWMB1. Software may write a new pulse width value into PWMB1 at any time and this new value will take effect at the start of the next PWM period (or when the LOAD bit in the PWMSIC register is written to a '1'). The PWMSM hardware does not modify the contents of PWMB1 at any time.

PWMB2 acts as a double buffer of PWMB1, allowing the contents of PWMB1 to be changed at any time without affecting the current pulse width of the output signal; it cannot be accessed directly by the software. PWMB1 can be read or written at any time. The new value in the PWMB1 register is transferred to PWMB2 on the next full cycle of the output or when a '1' is written to the LOAD bit in the PWMSIC register.

The pulse width comparator is a 16-bit 'ones-equality' comparator that compares the contents of the PWMB2 register with the 16-bit PWM counter. When the counter reaches the value in PWMB2, a match occurs and the output flip-flop is cleared. This pulse width match completes the pulse width; it does not affect the counter. Since a 'ones-equality' comparator is used, subsequent comparisons can occur, but will have no effect on the output signal as the output flip-flop has already been cleared.

The PWM output pulse may be as short as one PWM clock period (PWMB2 = 0x0001). It may be as long as one PWM clock period less than the PWM period; for example, the pulse width equal to 65535 PWM clock periods can be obtained by setting PWMB2 = 0xFFFF and PWMA2 = 0x0000.

13.7.5 0% and 100% 'Pulses'

The 0% and 100% 'pulses' are special limiting cases (zero width and infinite width) that are defined by the 'always clear' and 'always set' states of the output flip-flop.

The 0% pulse is generated by making the pulse width value in PWMB2 equal to 0x0000. The output is a true steady state signal with no glitches.

The 100% pulse is created by making the pulse width value in PWMB2 equal to or greater than the period value in PWMA2. The output is a true steady state signal with no glitches.

It is not possible to have a 100% duty cycle when the output period is selected to be 65536 PWM clock periods (by setting PWMB2 = 0x0000); in this case the maximum duty cycle is 99.998% (100 x 65535/65536).

When using the PWM output signal to generate analog levels, the 0% and 100% pulses provide the full scale values.

Even when 0% or 100% pulses are being generated, the 16-bit PWM counter continues to count and output changes to or from these limit values are done synchronously with the selected period.

13.7.6 PWMSM Coherency

Byte access of registers is discussed in **13.5.1 32-Bit Coherent Access**, however, it should be noted that byte writes to the double buffered registers PWMA1 and PWMB1 are not recommended as the transfer from the primary registers to the secondary registers is done on a word basis.

For most PWMSM operations, 16-bit accesses are sufficient and long word accesses are treated as two word accesses, with one exception — a long word write to the period/pulse width registers. In this case, if the long word write is done within the PWM period, there is no visible effect on the output signal and the new values are stored in PWMA1 and PWMB1 ready to be loaded into the buffer registers at the start of the next period. If, however, the long word write coincides with the end of the period, then the transfer of values from the primary registers to the secondary registers is suppressed until the end of the next PWM period; during this period, the current values in the secondary registers are used for the period and the pulse width.

13.7.7 PWMSM Interrupts

The FLAG bit in the PWMSIC register is set when a new period begins and indicates that the period and pulse width registers (PWMA1 and PWMB1) may be updated with values for the next output period and pulse width. When the FLAG bit is set, an interrupt request is generated on one of eight levels as defined by the interrupt level bits (IL[2:0]) in the PWMSIC register. If the interrupt level is set to zero, interrupts are disabled.

13.7.8 Freeze Action on the PWMSM

When the IMB FREEZE signal is recognized, the PWMSM counter stops incrementing and remains set at its last value. When the FREEZE signal is negated, the counter starts incrementing from its last value, as if nothing had happened.

13.7.9 PWM frequency, Pulse Width and Resolution

Table 13-12 and **Table 13-13** shows the pulse widths and frequencies that can be achieved using the /2 and /3 options and a clock frequency of 16.78 MHz.

Table 13-12 PWM Pulse and Frequency Ranges (in Hz) Using /2 Option (16.78 MHz)

Minimum Pulse Width	Bits of Resolution															
	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0.119µs/2	128	256	512	1024	2048	4096	8192	16384	32768	65.5k	131k	262k	524k	1049k	2097k	4195k
0.238µs/4	64	128	256	512	1024	2048	4096	8192	16384	32768	65.5k	131k	262k	524k	1049k	2097k
0.477µs/8	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65.5k	131k	262k	524k	1049k
0.954µs/16	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65.5k	131k	262k	524k
1.91µs/32	8.0	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65.5k	131k	262k
3.81µs/64	4.0	8.0	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65.5k	131k
7.63µs/128	2.0	4.0	8.0	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65.5k
30.5µs/512	0.5	1.0	2.0	4.0	8.0	16	32	64	128	256	512	1024	2048	4096	8192	16384

Table 13-13 PWM Pulse and FrequencyRanges (in Hz) Using /3 Option (16.78 MHz)

Minimum Pulse Width	Bits of Resolution															
	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0.179µs/3	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k	87.38k	174.8k	349.5k	699.1k	1398k	2796k
0.358µs/6	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k	87.38k	174.8k	349.5k	699.1k	1398k
0.715µs/12	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k	87.38k	174.8k	349.5k	699.1k
1.431µs/24	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k	87.38k	174.8k	349.5k
2.861µs/48	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k	87.38k	174.8k
5.722µs/96	2.667	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k	87.38k
11.44µs/192	1.333	2.667	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69k
45.78µs/768	0.333	0.667	1.333	2.667	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923

13.7.10 PWM Frequency

The relationship between the PWM output frequency (f_{PWMO}) and the MCU system clock frequency (f_{SYS}) is given by Equation .

$$f_{PWMO} = \frac{f_{SYS}}{N_{CLOCK} \bullet N_{COUNTER}}$$

where N_{CLOCK} is the CPSM clock divide ratio (2 or 3) and $N_{COUNTER}$ is the PWMSM counter divide ratio.

13.7.11 PWM Pulse Width

The minimum output pulse width (t_{PWMIN}) and the MCU system clock frequency (f_{SYS}) is given by Equation .

$$t_{PWMIN} = \frac{N_{CLOCK}}{f_{SYS}}$$

13.7.12 PWM Period and Pulse Width Register Values

The value to be loaded into the PWM period register (PWMA1) to obtain a given period is given by Equation .

$$PWMA1 = \frac{f_{SYS}}{N_{CLOCK} \bullet f_{PWMO}}$$

The value to be loaded into the PWM pulse width register (PWMB1) to obtain a given period is given by Equation .

$$PWMB1 = \frac{t_{PWMO}}{t_{PWMIN}} = \frac{\text{Duty cycle \%}}{100} \cdot PWMA1$$

where $t_{(PWMO)}$ is the actual output pulse width.

13.7.13 PWMSM Register Map and Registers

The PWMSM register map comprises four 16-bit registers as shown in **Table 13-14**. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no meaning nor effect. All register addresses in this section are specified as offsets from the base address of the PWMSM.

Table 13-14 PWMSM Register Map

Address	15 — 8	7 — 0
0xYF F228	PWM5 Status, interrupt and control register (PWM5SIC)	
0xYF F22A	PWM5 period register(PWM5A)	
0xYF F22C	PWM5 pulse width register (PWM5B)	
0xYF F22E,	PWM5 counter register (PWM5C)	
0xYF F230	PWM6 Status, interrupt and control register (PWM6SIC)	
0xYF F232	PWM6 period register(PWM6A)	
0xYF F234	PWM6 pulse width register (PWM6B)	
0xYF F236	PWM6 counter register (PWM6C)	
0xYF F238	PWM7 Status, interrupt and control register (PWM7SIC)	
0xYF F23A	PWM7 period register(PWM7A)	
0xYF F23C	PWM7 pulse width register (PWM7B)	
0xYF F23E	PWM7 counter register (PWM7C)	
0xYF F240	PWM8 Status, interrupt and control register (PWM8SIC)	
0xYF F242	PWM8 period register(PWM8A)	
0xYF F244	PWM8 pulse width register (PWM8B)	
0xYF F246	PWM8 counter register (PWM8C)	

13.7.13.1 PWMSIC — Status, Interrupt and Control Register

The PWMSIC register contains status, interrupt enable and control bits for the PWMSM. It also contains interrupt level and arbitration bits.

PWM5SIC — PWM Status/Interrupt Control Register **0xYF F228**
PWM6SIC **0xYF F230**
PWM7SIC **0xYF F238**
PWM8SIC **0xYF F240**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
FLAG	IL2	IL1	IL0	IARB3	0	0	0	PIN	0	LOAD	POL	EN	CLK2	CLK1	CLK0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13-15 PWMSIC Bit Settings

Bit(s)	Name	Description
15	FLAG	Period completion status. The FLAG bit is a status bit that indicates when the PWM output period has been completed. The FLAG bit is set by the hardware each time a PWM period is completed. Whenever the PWM is enabled, the FLAG bit is set immediately to indicate that the contents of the buffer registers PWMA2 and PWMB2 have been updated, and that the period using these new values has started. It also indicates that the user accessible period and pulse width registers PWMA1 and PWMB1 can be loaded with values for the next PWM period. Once set, the FLAG bit will remain set and will not be affected by any subsequent period completions, until it is cleared by the software. The FLAG bit can only be cleared by software. To clear the flag, the software must first read the bit (as 'one') then write a 'zero' to the bit. Writing a one to the FLAG bit has no effect. When the PWM is disabled the FLAG bit remains in the cleared state. The flag clearing mechanism will work only if no flag setting event occurs between the read and write operations; if a FLAG setting event occurs between the read and write operations, the FLAG bit will not be cleared. When the interrupt level set by the interrupt level bit IL[2:0] is not equal to zero, an interrupt request is generated when the FLAG bit is set. Before returning from the interrupt service routine, the FLAG bit should be cleared by software to prevent the PWMSM from immediately generating another interrupt request on the IMB. 0 = PWM period not completed. 1 = PWM period completed.
14:12	IL[2:0]	Interrupt level. The three interrupt level bits select the interrupt level of requests made by the PWMSMt. 000 = Interrupt disabled 001 = Interrupt level 1 (lowest) 010 = Interrupt level 2 011 = Interrupt level 3 100 = Interrupt level 4 101 = Interrupt level 5 110 = Interrupt level 6 111 = Interrupt level 7 (highest)
11	IARB3	Interrupt arbitration. The read/write IARB3 bit works in conjunction with the IARB[2:0] field in the BIUSM module configuration register. Each module that generates interrupt requests on the IMB must have a unique value in the arbitration field (IARB). This interrupt arbitration identification number is used to arbitrate for the IMB when modules generate simultaneous interrupts of the same priority. The IARB3 bit is cleared by reset.
10:8	—	Reserved
7	PIN	Output pin status. The PIN bit is a status bit that indicates the logic state present on the output pin. The software can thus monitor the waveform being created on the output pin. PIN is a read-only bit; writing to it has no effect. 0 = Logic zero state present on the output pin. 1 = Logic one state present on the output pin.
6	—	Reserved
5	LOAD	Load control. The LOAD bit is a control bit that allows the software to reinitialize the PWMSM and start a new PWM period without causing a glitch on the PWM output signal. 0 = No action. 1 = Load period and pulse width registers. This bit is always read as a zero. Writing a one to this bit results in the following immediate actions: – The contents of PWMA1 (period value) are transferred to PWMA2, – The contents of PWMB1 (pulse width value) are transferred to PWMB2, – The counter register (PWMC) is initialized to 0x0001, – The control logic and state sequencer are reset, – The FLAG bit is set, and – The output flip-flop is set if the new value in PWMB2 is different from 0x0000.

Table 13-15 PWMSIC Bit Settings (Continued)

Bit(s)	Name	Description
4	POL	Output pin polarity control. The POL bit is a control bit that allows the software to set the polarity of the PWM output signal. It works in conjunction with the EN bit and controls whether the PWMSM drives the output pin with the true or inverted value of the output flip-flop, see **Table 13-16**.
3	EN	Enable control. The EN bit is a control bit that allows the software to enable and disable the PWMSM as required. 0 = Disable the PWMSM and stop generation of PWM output pulses. 1 = Enable the PWMSM and start generation of PWM output pulses. While the PWMSM is disabled (EN = 0): – The output flip-flop is held reset and the level on the output pin is set to one or zero according to the state of the POL bit, – The PWMSM's divide-by-256 prescaler is held in reset, – The counter stops incrementing and is held equal to 0x0001, – The comparators are disabled, – And the PWMA1 and PWMB1 registers permanently transfer their contents to the buffer registers (PWMA2 and PWMB2, respectively). When the EN bit is changed from zero to one: – The output flip-flop is set to start the first pulse, – The PWMSM's divide-by-256 prescaler is released, – The counter is released and starts to increment from 0x0001, – And the FLAG bit is set (to indicate that PWMA1 and PWMB1 can be updated with new values of period and pulse width. While EN is set, the PWMSM generates continuously a pulse width modulated output signal based on the data in PWMA2 and PWMB2 (which are updated via PWMA1 and PWMB2 each time a period is completed). To prevent unwanted glitches on the output waveform when disabling the PWMSM, the EN bit should not be cleared by the software until one period has been output as a 0% pulse (PWMB2 = 0x0000)
2:0	CLK[2:0]	Clock rate selection. The CLK bits are control bits that allow the software to select one of the eight counter clock sources coming from the PWMSM prescaler. These bits can be changed by the software at any time. **Table 13-17** shows the counter clock sources and rates in detail.

Table 13-16 PWMSM Output Pin Polarity Selection

Control Bits		Output Pin State	Periodic Edge	Variable Edge	Optional Interrupt On
POL	EN				
0	0	Always low	—	—	—
1	0	Always high	—	—	—
0	1	High pulse	Rising edge	Falling edge	Rising edge
1	1	Low pulse	Falling edge	Rising edge	Falling edge

Table 13-17 PWMSM Clock Rate Selection

PWMSM CLK Bits			CPSM Bit DIV23	PWMSM Clock	Clock Source
CLK2	CLK1	CLK0			
0	0	0	0	f_{SYS} / 2	PCLK1
0	0	1	0	f_{SYS} / 4	Prescaler (/2)
0	1	0	0	f_{SYS} / 8	Prescaler (/4)
0	1	1	0	f_{SYS} / 16	Prescaler (/8)
1	0	0	0	f_{SYS} / 32	Prescaler (/16)
1	0	1	0	f_{SYS} / 64	Prescaler (/32)
1	1	0	0	f_{SYS} / 128	Prescaler (/64)
1	1	1	0	f_{SYS} / 512	Prescaler (/256)
0	0	0	1	f_{SYS} / 3	PCLK1
0	0	1	1	f_{SYS} / 6	Prescaler (/2)
0	1	0	1	f_{SYS} / 12	Prescaler (/4)
0	1	1	1	f_{SYS} / 24	Prescaler (/8)
1	0	0	1	f_{SYS} / 48	Prescaler (/16)
1	0	1	1	f_{SYS} / 96	Prescaler (/32)
1	1	0	1	f_{SYS} / 192	Prescaler (/64)
1	1	1	1	f_{SYS} / 768	Prescaler (/256)

13.7.13.2 PWMA — PWM Period Register

The PWMA register contains the period value for the next cycle of the PWM output waveform. In normal usage, with the PWMSM enabled, the software writes a period value into PWMA1 and this value is then loaded into the PWMA2 register at the end of the current period. If the PWMSM is disabled, a period value written to PWMA1 is loaded into PWMA2 on the next tic (of the MCU system clock). PWMA2 is a temporary register that is used for smoothly updating the PWM period value; it cannot be read or written directly by software.

Software may write a new period value into PWMA1 at any time and this new value will take effect at the start of the next PWM period (or when the LOAD bit in the PWMSIC register is written to a ‘1’). The PWMSM hardware does not modify the contents of PWMA1 at any time.

PWM5A — PWM Period Register **0xYF F22A**
PWM6A **0xYF F232**
PWM7A **0xYF F23A**
PWM8A **0xYF F242**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

13.7.13.3 PWMB — PWM Pulse Width Register

The PWMB register contains the pulse width value for the next cycle of the PWM output waveform. In normal usage, with the PWMSM enabled, the software writes a pulse width value into PWMB1 and this value is then loaded into the PWMB2 register at the end of the current period. If the PWMSM is disabled, a pulse width value written to PWMB1 is loaded into PWMB2 on the next tic (of the MCU system clock). PWMB2 is a temporary register that is used for smoothly updating the PWM pulse width value; it cannot be read or written directly by software.

Software may write a new pulse width value into PWMB at any time and this new value will take effect at the start of the next PWM period (or when the LOAD bit in the PWMSIC register is written to a ‘1’). The PWMSM hardware does not modify the contents of PWMB1 at any time.

PWM5B — PWM Pulse Width Register **0xYF F22C**
PWM6B **0xYF F234**
PWM7B **0xYF F23C**
PWM8B **0xYF F244**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

13.7.13.4 PWMC — PWM Counter Register

The counter (register PWMC) is read-only: software may read the counter register at any time; writing to it has no effect. PWMC is loaded with the value 0x0001 on reset and is set to that value and held whenever the PWMSM is disabled (EN = 0).

PWM5C — PWM Counter Register **0xYF F22E**
PWM6C **0xYF F236**
PWM7C **0xYF F23E**
PWM8C **0xYF F246**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

13.8 Bus Interface Unit Submodule (BIUSM)

The bus interface unit submodule (BIUSM) allows all the CTM9 submodules to communicate to the IMB3 via the SMB (sub module bus).

13.8.1 Freeze Action on the BIUSM

When the IMB freeze condition is detected, the FRZ bit in the BIUSM module configuration register determines whether or not the freeze condition is passed on to the other CTM submodules. If FRZ = 0, the freeze condition is ignored; if FRZ = 1, the BIUSM passes the FREEZE signal from the IMB through to the CTM submodules. Each CTM submodule then reacts to the FREEZE signal as defined by its own internal circuitry and control bits.

13.8.2 LPSTOP Action on the BIUSM

When the CPU is stopped by an LPSTOP instruction (from CPU32 or CPU16), the system clock (f_{SYS}) is stopped, thereby shutting down all dependent modules, including the CTM, until the low-power STOP mode is exited.

13.8.3 STOP and WAIT Action on the BIUSM

When the STOP instruction on CPU32 or the WAIT instruction on CPU16 is executed, only the CPU is stopped; the CTM continues to operate as normal. (To stop the CTM operation selectively, refer to the description of the STOP bit in **13.8.4.1 BIUMCR — BIUSM Module Configuration Register**).

13.8.4 BIUSM Registers

The BIUSM register map comprises four 16-bit register locations. As shown in **Table 13-18**, the register block contains the three BIUSM registers and one reserved register. The BIUSM register block always occupies the first four register locations in the CTM register space and cannot be relocated within the CTM structure. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no effect.

Table 13-18 BIUSM Register Map

Address	15 — 8	7 — 0
0xYF F200	BIUSM module configuration register (BIUMCR)	
0xYF F202	BIUSM test register (BIUTST)	
0xYF F204	BIUSM time base register (BIUTBR)	

13.8.4.1 BIUMCR — BIUSM Module Configuration Register

The BIUMCR register contains nine defined bits that allow the software to control five functions of the CTM: enabling/disabling of the module, response to FREEZE, vector base address, interrupt arbitration number and access to the time base buses (via the time base register).

BIUMCR — BIUSM Module Configuration Register **0xYF F200**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
STOP	FRZ	0	VECT7	VECT6	IARB2	IARB1	IARB0	0	0	TBRS1	0	0	0	0	TBRS0
RESET:															
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Table 13-19 BIUMCR Bit Settings

Bit(s)	Name	Description
15	STOP	Stop enable. The STOP bit, while asserted, activates the FREEZE signal on the SMB regardless of the state of the FREEZE signal on the IMB. This completely stops the operation of the CTM. Note that some submodules may validate this signal with internal enable bits. The BIUSM continues to operate to allow the CPU access to the submodule's registers. The SMB FREEZE signal remains active until reset or until the STOP bit is negated by the CPU (via the IMB). 0 = Allows operation of the CTM. 1 = Stops operation of the CTM.
14	FRZ	Freeze enable. The FRZ bit, while asserted, activates the FREEZE signal on the SMB when the FREEZE signal on the IMB is active. This completely stops the operation of the CTM. Note that some submodules may validate this signal with internal enable bits. The BIUSM continues to operate to allow the CPU access to the submodule's registers. The SMB FREEZE signal remains active until the FRZ bit is cleared or the IMB FREEZE signal is negated. 0 = Ignores the FREEZE signal on the IMB. 1 = Halts the CTM sub module when the FREEZE signal appears on the IMB.
13	—	Reserved
12:11	VECT[7:8]	Interrupt vector base number. The interrupt vector base number bits select the interrupt vector base number for the CTM. Of the 8 bits necessary for vector number definition, the six least significant bits are programmed by hardware on a submodule basis, while the two remaining bits are provided by VECT7 and VECT6. 00 = Vector base number 0x00. 01 = Vector base number 0x40. 10 = Vector base number 0x80. 11 = Vector base number 0xC0.

Table 13-19 BIUMCR Bit Settings (Continued)

Bit(s)	Name	Description
10:8	IARG[2:0]	Interrupt arbitration identification. The interrupt arbitration bit field (IARB), composed of IARB[2:0] in the BIUMCR and the IARB3 bit within each submodule, provides fifteen different arbitration identification numbers that can be used to arbitrate between interrupt requests occurring on the IMB with the same interrupt priority level. The IARB field defaults to zero on reset, thus preventing the module from arbitrating during an interrupt arbitration acknowledge cycle (IACK). If no IMB arbitration takes place during the IACK cycle the spurious interrupt vector is generated by the SIM (system integration module). This tells the system that the interrupt arbitration number has not been initialized. The seven levels of interrupt are the primary means by which interrupt priority is established. The 4-bit interrupt arbitration number is the secondary priority, allowing up to 15 requests at each primary level. During the IACK cycle the request with the highest arbitration number gets serviced (binary 1111 is the highest priority and binary 0001 is the lowest). Many IMB modules have one software assignable arbitration number for the whole module. The CTM allows two different arbitration numbers to be used by providing each submodule with its own IARB3 bit (which can be set or cleared in software). Once IARB[2:0] are assigned in the BIUSM, they apply to all CTM interrupt requests. Therefore, CTM submodule interrupts can be interleaved in priority with requests from other modules at the same interrupt level.
7:6	—	Reserved
5,0	TBRS1, TBRS0	Time base register bus select. These bits specify which time base bus is accessed when the time base register (BIUTBR) is read. 00 = Time base bus TBB1 01 = Time base bus TBB2 10 = Time base bus TBB3 11 = Time base bus TBB4
4:1	—	Reserved

13.8.4.2 BIUTBR — BIUSM Time Base Register

In normal operation, the BIUTBR is a read-only register used to read the value present on one of the time base buses. The time base bus being accessed is determined by TBRS1 and TBRS0 in the BIUMCR. Writing to the BIUTBR has no effect, except in certain test modes.

BIUTBR — BIUSM Time Base Register **0xYF F204**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSB								LSB							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.9 Counter Prescaler Submodule (CPSM)

The counter prescaler submodule (CPSM) generates six different clock frequencies which can be used by any counter submodule. Five of these frequencies are derived from a fixed divider. The divide ratio of the last clock frequency is software selectable from a choice of four divide ratios. Note that this submodule is contained within the BIUSM. A block diagram of the CPSM is given in **Figure 13-14**. The clock division ratios available on PCLKx are also shown in the table in **13.9.2.1 CPCR — CPSM Control Register**. These clock signals are provided on the SMB and may be used by any or all CTM submodules.

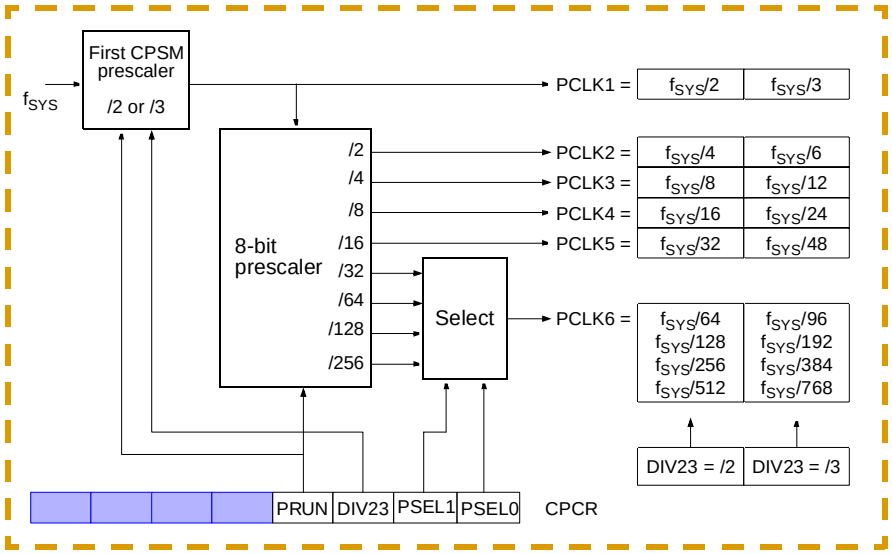


Figure 13-14 CPSM Block Diagram

13.9.1 Freeze Action on the CPSM

When the IMB FREEZE signal is recognized, the CPSM counters stop counting and remain set at their current values. When the FREEZE signal is negated, the counters start incrementing from their current values, as if nothing had happened. All registers are accessible during freeze.

13.9.2 CPSM Registers

The CPSM register map comprises four 16-bit register locations. As shown in **Table 13-20**, the register block contains two CPSM registers and two reserved registers. The CPSM register block always immediately follows the BIUSM register block in the CPSM register map. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no effect.

Table 13-20 CPSM Register Map

Address	15 8	7 0
0xYF F208	CPSM control register (CPCR)	
0xYF F20A	CPSM test register (CPTR)	

13.9.2.1 CPCR — CPSM Control Register

CPCR — CPSM Control Register **0xYF F208**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
0	0	0	0	0	0	0	0	0	0	0	0	PRUN	DIV23	PSEL 1	PSEL 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13-21 CPCR Bit Settings

Bit(s)	Name	Description
15:4	—	Reserved
3	PRUN	Prescaler running. The PRUN bit is a read/write control bit that allows the software to switch the prescaler counter on and off. This bit allows the counters in various CTM submodules to be synchronized. 0 = Prescaler divider is held in reset and is not running. 1 = Prescaler is running.
2	DIV23	Divide by 2 or divide by 3 . The DIV23 bit is a read/write control bit that selects the division ratio of the first prescaler counter. It may be changed by the software at any time and is cleared on reset. 0 = First prescaler stage divides by 2. 1 = First prescaler stage divides by 3.
1:0	PSEL[1:0]	Prescaler division ratio select. These control bits select the division ratio of the programmable prescaler output signal, PCLK6, See **Table 13-22**.

Table 13-22 Prescaler Division Ratio Select

Prescaler Control Register Bits				Prescaler Division Ratio					
PRUN	DIV23	PSEL1	PSEL0	PCLK1	PCLK2	PCLK3	PCLK4	PCLK5	PCLK6
0	X	X	X	0	0	0	0	0	0
1	0	0	0	2	4	8	16	32	64
1	0	0	1	2	4	8	16	32	128
1	0	1	0	2	4	8	16	32	256
1	0	1	1	2	4	8	16	32	512
1	1	0	0	3	6	12	24	48	96
1	1	0	1	3	6	12	24	48	192
1	1	1	0	3	6	12	24	48	384
1	1	1	1	3	6	12	24	48	768

13.9.3 Clock Sources for the Counter Submodules

The software chooses one of seven clock sources for each counter. Six of them are prescaler taps derived from the on-chip oscillator. The highest frequency available to the counter is the MCU system clock divided by 2. Four of the other five taps are binary divisible from the system clock cycle — divide by 4, 8, 16, and 32. Another input clock to the counter is a software defined divide by 64, 128, 256, or 512 from the MCU clock. There is an alternate prescaler option where the MCU clock is divided by 3, 6, 12, 24,

48, 96, 192, 384, and 768. The seventh selectable clock source is an external pin, which may trigger on the rising or falling edge of the input signal. The external input allows the counter to use a frequency not based on the microcontroller oscillator. An alternate use for the external clock source is for event or pulse counting.

13.10 CTM9 Interrupts

The CTM9 is able to generate a diverse set of interrupts on the IMB3. Each interrupting submodule is capable of requesting an interrupt on any of seven levels. A 3-bit level number and a 1-bit arbitration number included in each submodule are initialized by the software. The 3-bit level number selects which of the seven interrupt signals on the IMB are driven by that submodule to create an interrupt request. Of the four priority bits provided on the IMB3 during arbitration among the modules, one of them comes from the interrupting submodule and the CTM9 BIUSM provides the other three. Thus, the CTM9 may respond to two of the possible fifteen arbitration numbers.

During the IMB3 arbitration process, the CTM9 BIUSM manages the separate arbitration among the CTM9 submodules to determine which submodule will respond. Of the submodules which have an interrupt request pending at the level being arbitrated on the IMB, the submodule which has the lowest address is given the highest priority to respond.

Following the interrupt arbitration process, the CTM9 provides an 8-bit vector number. Six of the eight bits are provided by the interrupting submodule. Of the submodules produced to date, a submodule can identify up to two separate interrupt causes, each with unique interrupt vectors. The high-order two bits of the 8-bit vector are provided by the CTM9 BIUSM. The low order six vector bits identify the highest priority interrupt request pending in the CTM9 at the beginning of the arbitration cycle.

13.11 CTM9 Function Examples

The versatility of the CTM9 timer architecture is based on multiple counters and capture/compare channel units interconnected on time-base buses. Rather than present block diagrams of each submodule, this section includes some typical application examples — to show how the submodules can be interconnected to form timing functions. The diagrams used to illustrate these examples show only the blocks utilized for that function.

To illustrate the timing range of the CTM9 in different applications, many of the following paragraphs include time intervals quoted in microseconds and seconds. The assumptions used are that the microcontroller system clock is at 16.78 MHz with minimum prescaling (0.119 microsecond cycle) and with the maximum prescaling (48.0 microsecond cycle). For other system clock cycle rates and prescaler choices, the times mentioned in these paragraphs scale appropriately.

13.11.1 CTM9 Single Input Capture

The CTM9 single-action submodule (SASM) has an input capture register to latch the current state of a time-base bus when an external input edge is detected. The SASM is software programmable to latch on the rising or falling edge of the input signal. The

software also selects one of two time-base buses, each originating at a counter submodule. The software can also enable an interrupt to occur when the input edge is detected to notify the software that new edge capture information is available.

Figure 13-15 shows an example of an MCSM a counter submodule for an SASM configured for input capturing. To measure the period of an incoming signal, the software reads and saves the latched value in register A for one edge, then when the next edge arrives, the software subtracts the new captured value in register A from the previously saved value to obtain the period interval. The maximum period that can be measured is the worst case software response time to a newly captured value.

The software measures the width of a pulse in a very similar way, the only difference is that after each edge, the edge detector is reprogrammed to trigger on the next opposite edge.

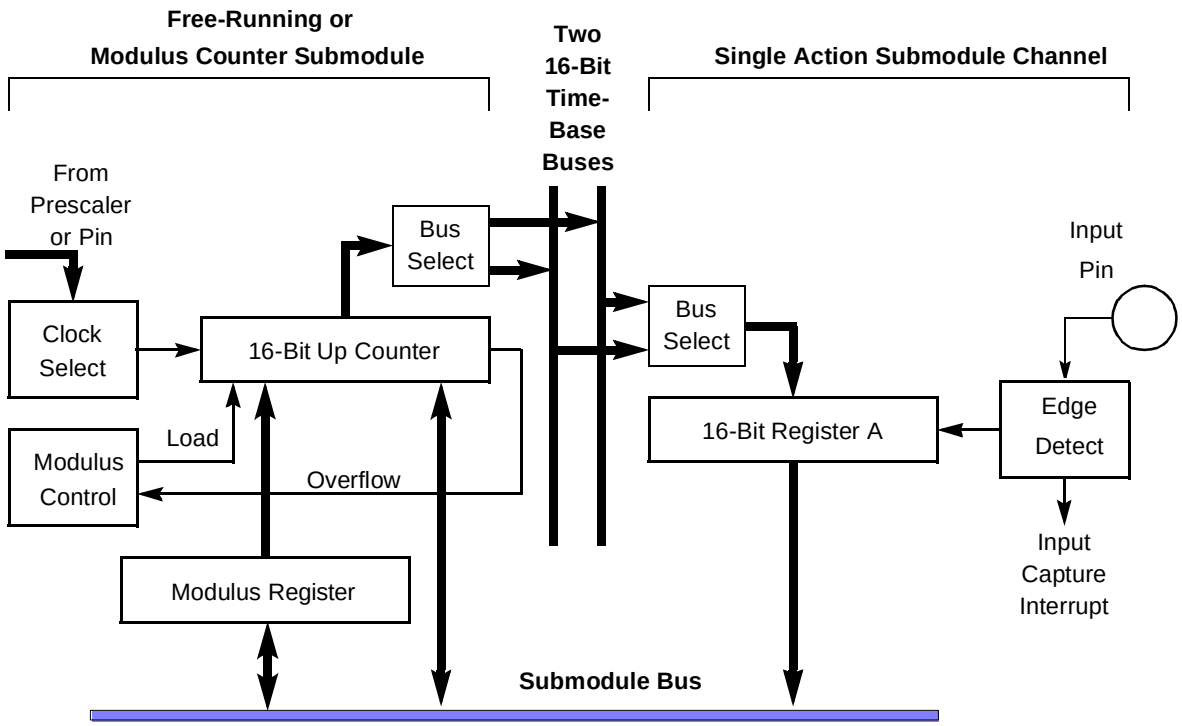


Figure 13-15 CTM9 Example — Single Edge Input Capture

13.11.2 CTM9 Input Double Edge Pulse Width Measurement

To measure the width of an input pulse, the CTM9 double-action submodule (DASM) has two capture registers so that only one interrupt is needed after the second edge. The software can read both edge samples and subtract them to get the pulse width. The leading edge sample is double latched so that the software has the time of one full period of the input signal to read the samples to be sure that nothing is lost. Depending on the prescaler divide ratio, pulses width from 0.119 microseconds to 3

seconds can be measured. Note that a software option is provided to also generate an interrupt after the first edge.

In the example shown in **Figure 13-16**, a counter submodule is used as the time-base for a DASM configured in the input pulse width measurement mode. When the leading edge (programmed for either rising or falling edge) of the input signal occurs, the state of the time-base bus is saved in register B1. When the trailing edge occurs, the time-base bus is latched into register A, and the content of register B1 is transferred to register B2. This operation leaves register B1 free for the next leading edge to occur, as soon as on the next clock cycle. When enabled, an interrupt is provided after the trailing edge, to notify the software that pulse width measurement data is available for a new pulse. After the trailing edge, the software has one cycle time of the input signal to obtain the values for each edge. When software attention is not needed for every pulse, the interrupt can be disabled. The software can at any time read registers A and B2 coherently (using a 32-bit read instruction) to get the latest edge measurements. Since the measurement resolution is 16 bits, signals with pulse duty cycles from 0.0015% to 99.9985% can be measured. The software work is less than half that needed with a timer that requires the software to read one edge and save the value, and then wait for the second edge.

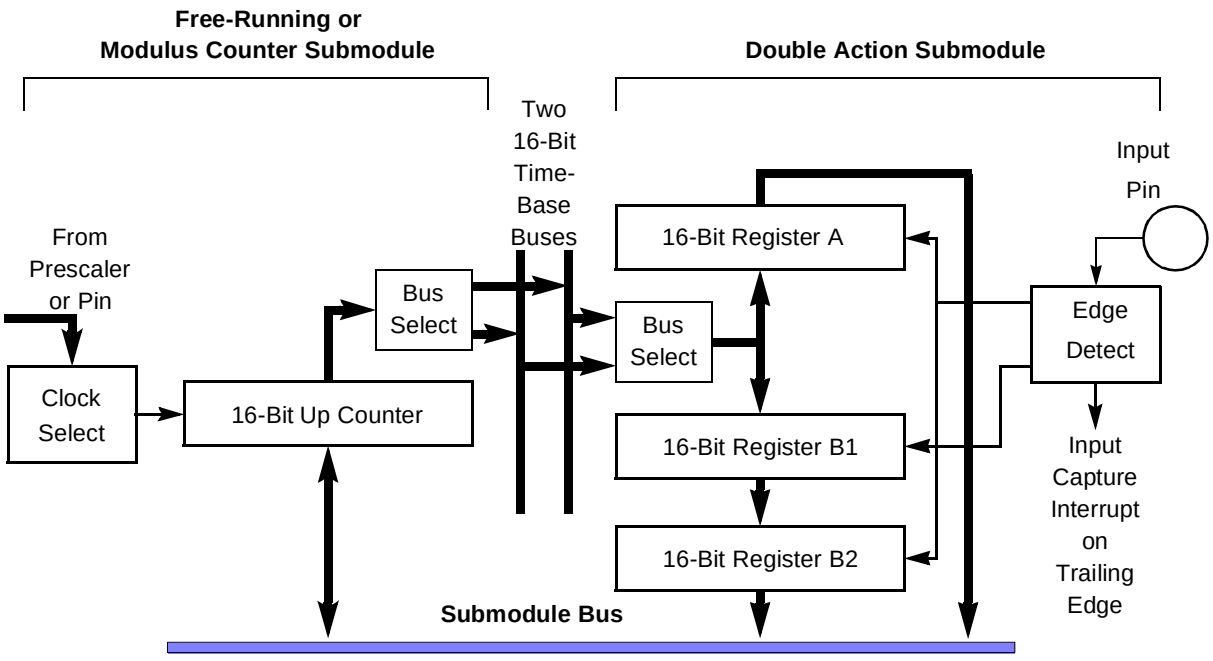


Figure 13-16 CTM9 Example — Double Capture Pulse Width Measurement

13.11.3 CTM9 Input Double Edge Period Measurement

Two samples are also available to the software from a double-action submodule for period measurement. The software can read the previous and the current edge samples and subtract them. As with pulse width measurement, the software can be sure

of not missing samples by insuring that the interrupt response time is faster than the fastest input period. Alternately, when the software is just interested in the latest period measurement, one 32-bit coherent read instruction can get both the current and the previous samples. Depending on the prescaler divide ratio, period times can be measured from 0.119 microseconds to 3 seconds.

Figure 13-17 shows a counter submodule and a DASM combination as an example of period measurement. The software designates whether the rising or falling edge of the input signal is to be used for the measurements. When the edge is detected, the state of the time-base bus is stored in register A, and the content of register B1 is transferred into register B2. After register B2 is safely latched, the content of register A is transferred to register B1. This procedure gives the software coherent current and previous samples in registers A and B2 at all times. An interrupt is available for the cases where the software needs to be aware of each new sample. Note that a software option is provided to also generate an interrupt after the first edge.

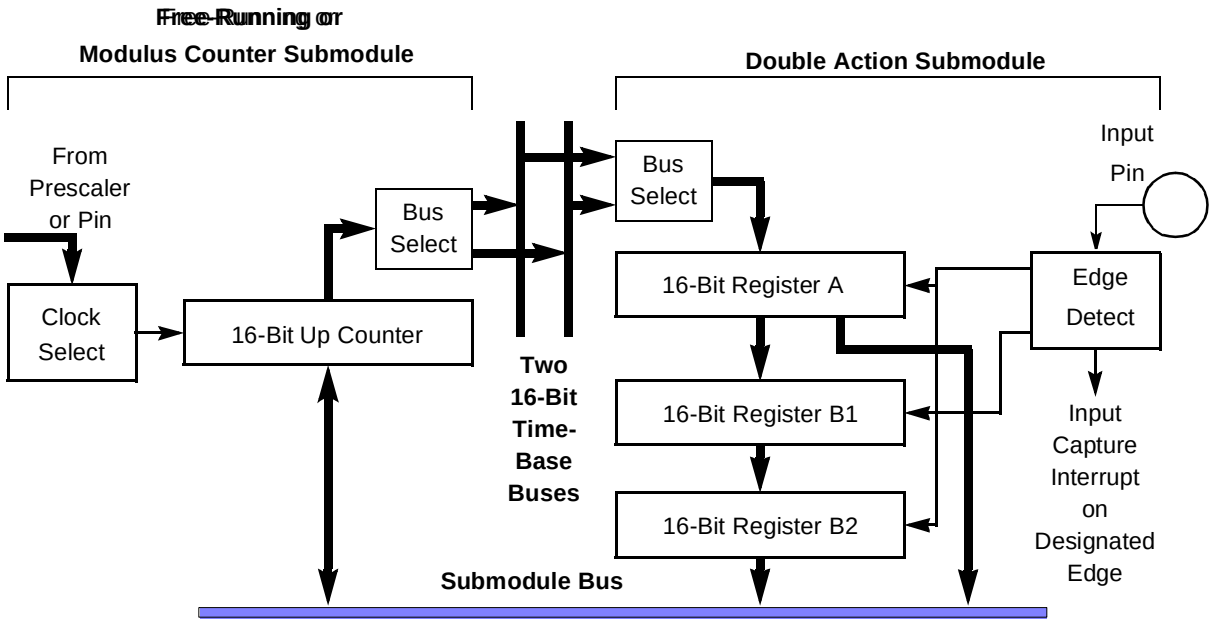


Figure 13-17 CTM9 Example — Double Capture Period Measurement

13.11.4 CTM9 Single Output Compare

To create one output edge, the software can use an SASM channel. The software provides a compare value in a register and the SASM compares that value to the incrementing value seen on one of the time-base buses. When a comparison is detected, the state of the output pin is changed.

The example shown in **Figure 13-18** uses a counter submodule with one channel of an SASM to create an output signal. The software can read the current state of the counter submodule. That, or some other criteria, is used to determine the time-base

value when the output is to change. The software thus writes the compare value into register A. In the SASM control register, the software establishes whether the output flip-flop is to toggle to the opposite state, or is to go to a high or a low level. The output compare interrupt is typically used to notify the software that the previous compare is complete and the SASM is available for a new compare value in register A.

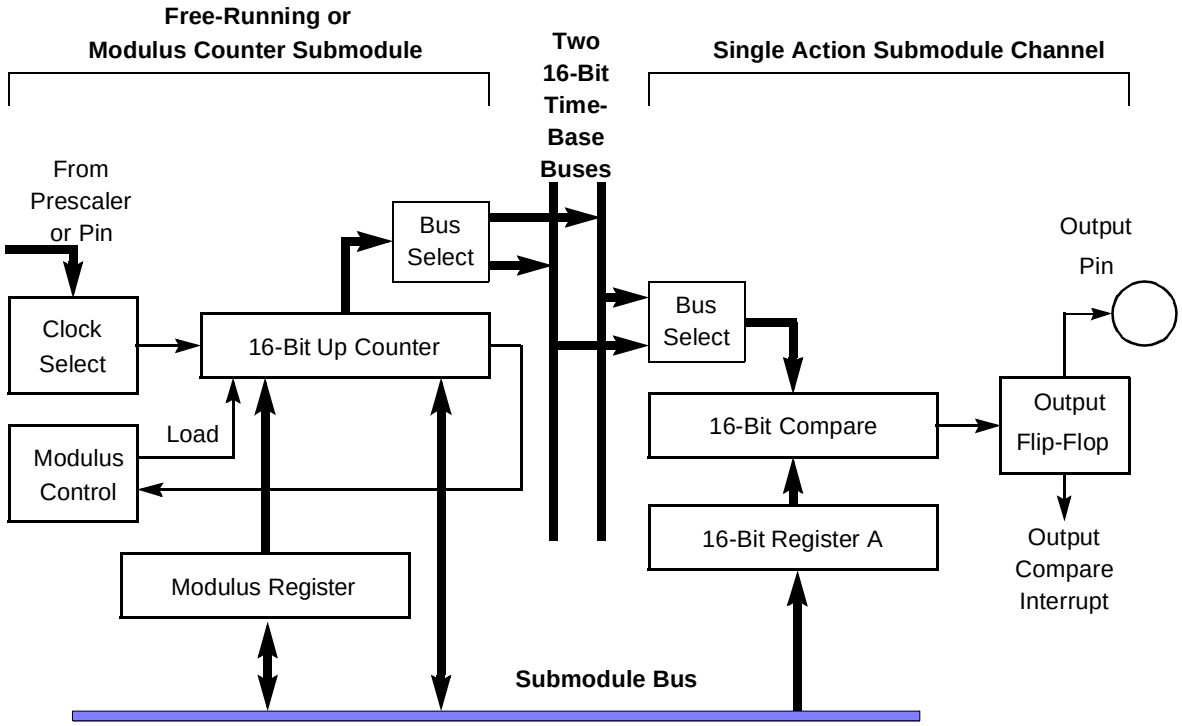


Figure 13-18 CTM9 Example — Single Edge Output Compare

13.11.5 CTM9 Double Edge Single Output Pulse Generation

Software can initialize the CTM9 to generate both the rising and the falling edge of an output pulse. With a DASM, pulses as narrow as 0.119 microseconds can be generated since software action is not needed between the edges. Pulses as long as 3 seconds can be generated. When an interrupt is desired, it can be selected to occur on every edge or only after the second edge.

Figure 13-19 shows how a counter submodule and a DASM can be used to generate both edges of a single output pulse. The software puts the compare value for one edge in register A and the other one in register B2. The DASM automatically creates both edges, and the pulse can be selected by software to be a high-going or a low-going. After the trailing edge, the DASM stops to await further commands from the software. Note that a single edge output can be generated by writing to only one register.

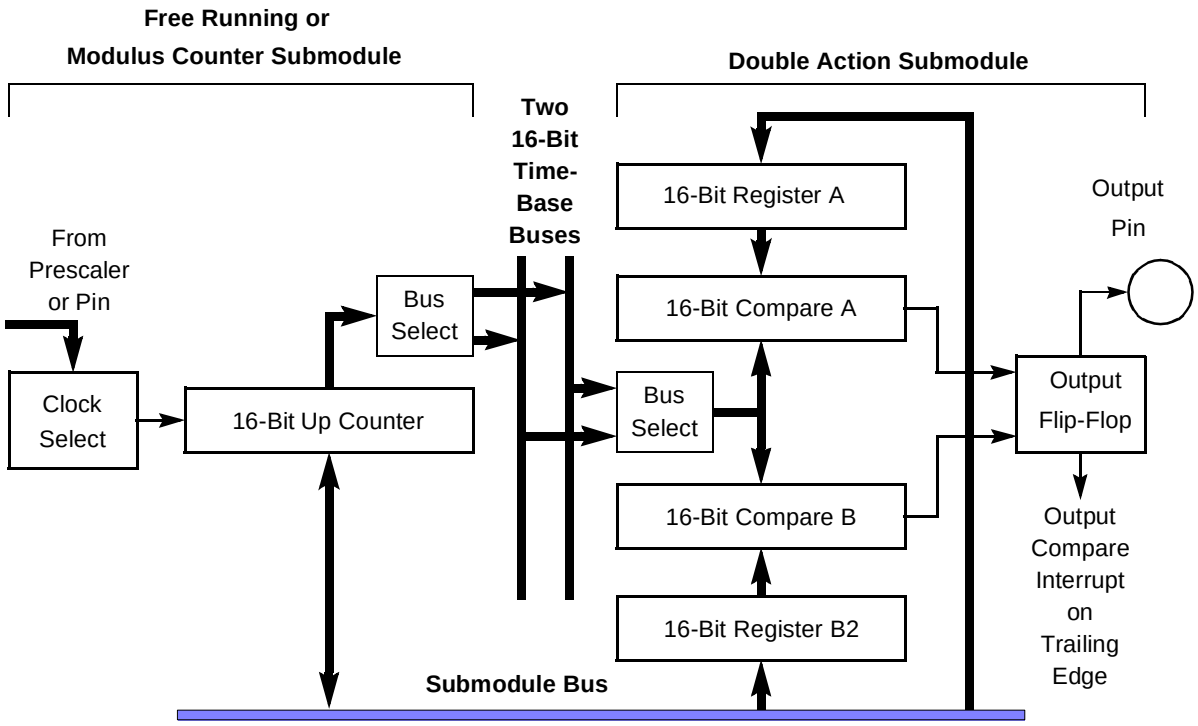


Figure 13-19 CTM9 Example — Double Edge Output Compare

13.11.6 CTM9 Output Pulse Width Modulation With DASM

Output waveforms can be generated with any duty cycle without software involvement. The software sets up a DASM with the compare times for the rising and falling edges, and they are automatically repeated. The software does not need to respond to interrupts to generate continuous pulses. The period may be selected as the period of a free-running counter submodule time-base, times a binary multiplier selected in the DASM. Multiple PWM outputs can be created from multiple DASMs and share one counter submodule, provided that the periods of all of the output signals are a binary multiple of the time-base, and that the counter submodule is operating in a free-running mode. Each DASM has a software selectable “don’t care” on high-order bits of the time-base comparison so that the period of one output can be a binary multiple of another signal. Masking the time-base serves to multiply the period of the time-base by a binary number to form the period of the output waveform. The duty cycle can vary from one cycle to 64K cycles. The frequency can range from 0.3 Hz to 62.5 KHz, though the resolution decreases at the higher frequencies to as low as 7 bits. The generation of output square wave signals is of course the special case where the high and low time are equal.

When an MCSM is used to drive the time-base, the modulus value is the period of the output PWM signal. **Figure 13-20** shows such an example. The polarity of the leading edge of an output waveform is programmable for a rising or a falling edge. The software selects the period of the output signal by programming the MCSM with a modulus

value. The leading edge compare value is written into register A by software, and the trailing edge time is written into register B1. When the leading edge value is reached, the content of register B1 is transferred to register B2, to form the next trailing edge value. Subsequent changes to the output pulse width are made by writing a new time into register B1. Updates to the pulse width are always synchronized to the leading edge of the waveform.

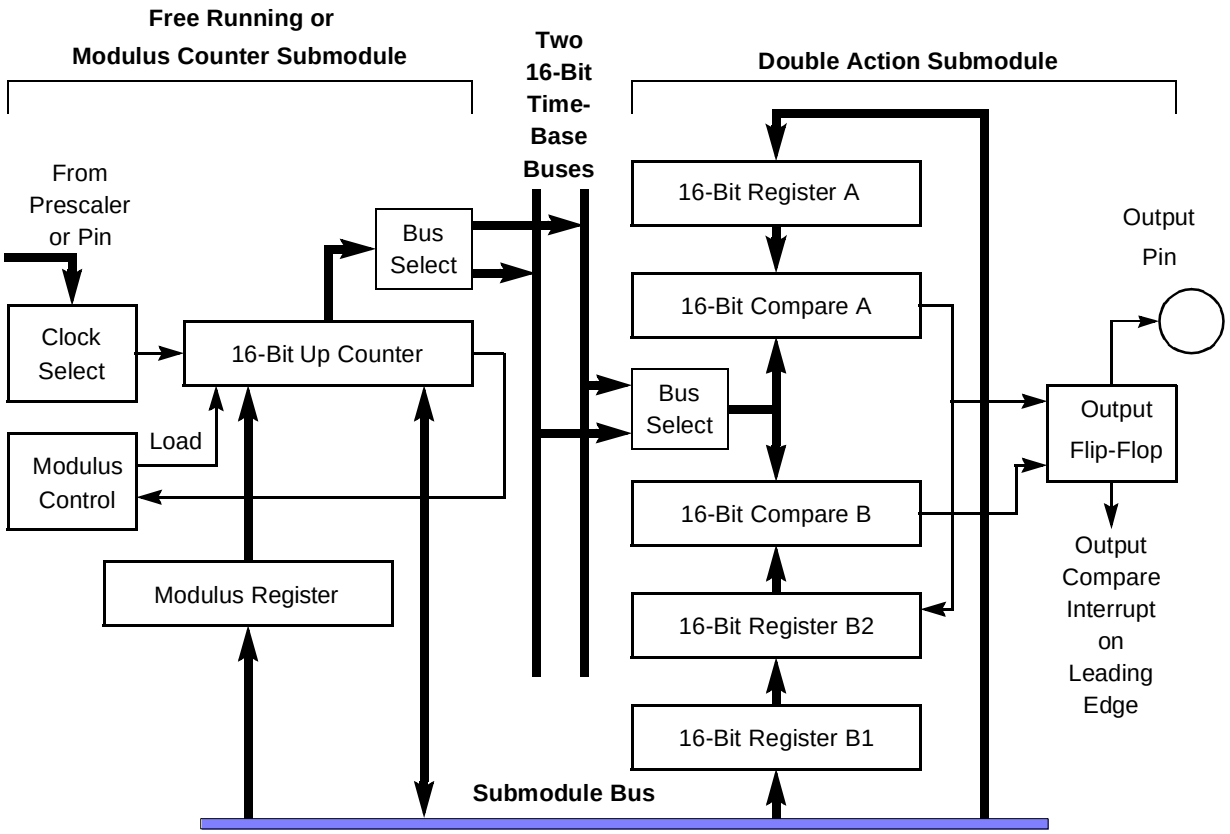


Figure 13-20 CTM9 Example — Pulse Width Modulation Output

It is typical to use the pulse width modulation mode of the DASM without interrupts, though an interrupt can be enabled to occur on the leading edge. When the output is an unchanging repetitive waveform, the DASM continuously generates the signal without any software intervention. When the software needs to change the pulse width, a new trailing edge time is written to the DASM. The output is changed on the next full pulse. When the software needs to change the output at a regular rate, such as an acceleration curve, the leading edge interrupt gives the software one period time to update the new trailing edge time.

13.11.7 CTM9 Input Pulse Accumulation

Counting the number of pulses on an input signal is another capability of the CTM9. Pulse accumulation uses either an FCSM or an MCSM. Since the counters in the

counter submodules are software readable, pulse accumulation does not require the use of an action submodule. The pulse accumulation can operate continuously, interrupting only on binary overflow of the 16-bit counter. When an MCSM is used, an interrupt can instead be created when the pulse accumulation reaches a preprogrammed value. To do that, the two's complement of the value is put in the modulus register and the interrupt occurs when the counter overflows. A similar function can be accomplished with the free-running counter submodule by writing a value into the counter.