CTMRM/D

Modular Microcontroller Family

CTM

CONFIGURABLE TIMER MODULE

REFERENCE
MANUAL

Modular Microcontroller Family
CTM REFERENCE MANUAL
MOTOROLA

MOTOROLA

CTM

Configurable Timer Module
Reference Manual

GJL-ekb

Conventions

Where abbreviations are used in the text, an explanation can be found in the glossary, at the back of this document. Register and bit mnemonics are defined in the paragraphs describing them.

A horizontal bar over a signal name indicates that the signal is active-low, e.g. $\overline{\text{RESET}}$.

Unless stated otherwise, shaded cells in a register diagram indicate that the bits are either unimplemented bits or reserved, and always read as zero.

In register diagrams, ‘u’ indicates that the state on reset is undefined.

When a bit is ‘set’, it has the value 1 (one).

When a bit is ‘clear’, it has the value 0 (zero).

When a bit is ‘reset’, it has its default value, which may be 1 or 0.

Reference documents

CPU16 Central Processor Unit Reference Manual (CPU16RM/D)

CPU32 Central Processor Unit Reference Manual (CPU32RM/AD)

GPT General Purpose Timer Reference Manual (GPTRM/AD)

An introduction to the HC16 for HC11 users (AN461/D)

CUSTOMER FEEDBACK QUESTIONNAIRE (CTMRM)

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this card (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organization	☐	☐	☐	☐	Tables	☐	☐	☐	☐
Readability	☐	☐	☐	☐	Table of contents	☐	☐	☐	☐
Understandability	☐	☐	☐	☐	Index	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐	Page size/binding	☐	☐	☐	☐
Illustrations	☐	☐	☐	☐	Overall impression	☐	☐	☐	☐

Comments: _____

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application ☐
System design ☐
Training purposes ☐
Other ☐ Please specify: _____

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely ☐ ☐ ☐ ☐ Not at all

Comments: _____

4. How easy is it to find the information you are looking for?

Easy ☐ ☐ ☐ ☐ Difficult

Comments: _____

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

		Too little detail				Too much detail
Table of Contents/List of Figures/List of Tables		☐	☐	☐	☐	☐
SECTION 1	FUNCTIONAL OVERVIEW	☐	☐	☐	☐	☐
SECTION 2	BUS INTERFACE UNIT SUBMODULE (BIUSM)	☐	☐	☐	☐	☐
SECTION 3	INTERRUPTS	☐	☐	☐	☐	☐
SECTION 4	COUNTER PRESCALER SUBMODULE (CPSM)	☐	☐	☐	☐	☐
SECTION 5	FREE-RUNNING COUNTER SUBMODULE (FCSM)	☐	☐	☐	☐	☐
SECTION 6	MODULUS COUNTER SUBMODULE (MCSM)	☐	☐	☐	☐	☐
SECTION 7	SINGLE ACTION SUBMODULE (SASM)	☐	☐	☐	☐	☐
SECTION 8	DOUBLE ACTION SUBMODULE (DASM)	☐	☐	☐	☐	☐
SECTION 9	PULSE WIDTH MODULATION SUBMODULE (PWMSM)	☐	☐	☐	☐	☐
SECTION 10	ELECTRICAL SPECIFICATIONS	☐	☐	☐	☐	☐
APPENDIX A	REGISTER SUMMARY	☐	☐	☐	☐	☐
APPENDIX B	CTM EXAMPLE – CTM2	☐	☐	☐	☐	☐
APPENDIX C	GLOSSARY	☐	☐	☐	☐	☐
APPENDIX D	INDEX	☐	☐	☐	☐	☐

Comments: _____

6. Have you found any errors? If so, please comment: _____

7. From your point of view, is anything missing from the document? If so, please say what: _____

– *Cut along this line to remove* –

8. How could we improve this document?

9. How would you rate Motorola's documentation?

	Excellent			Poor
– In general	☐	☐	☐	☐
– Against other semiconductor suppliers	☐	☐	☐	☐

10. Which semiconductor manufacturer provides the best technical documentation?
11. Which company (in any field) provides the best technical documentation?
12. How many years have you worked with microprocessors?

Less than 1 year ☐ 1–3 years ☐ 3–5 years ☐ More than 5 years ☐

– Second fold back along this line –

By air mail
Par avion

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.,
Colvilles Road,
Kelvin Industrial Estate,
EAST KILBRIDE,
G75 8BR.
GREAT BRITAIN.

F.A.O. Technical Publications Manager
(re: CTMRM/D)

SECTION

MOTOROLA LTD.
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

13. Currently there is some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment.

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: Phone No:
Position: FAX No:
Department:
Company:
Address:

Thank you for helping us improve our documentation,
Graham Livey, Technical Publications Manager, Motorola Ltd., Scotland.

– Last tuck this edge into opposite flap –

TABLE OF CONTENTS

3
INTERRUPTS

4
COUNTER PRESCALER SUBMODULE (CPSM)

5
FREE-RUNNING COUNTER SUBMODULE (FCSM)

6
MODULUS COUNTER SUBMODULE (MCSM)

7
SINGLE ACTION SUBMODULE (SASM)

8
DOUBLE ACTION SUBMODULE (DASM)

9
PULSE WIDTH MODULATION SUBMODULE (PWMSM)

10
ELECTRICAL SPECIFICATIONS

APPENDIX A
REGISTER AND BIT SUMMARY

APPENDIX B
CTM EXAMPLE – CTM2

APPENDIX C
GLOSSARY

APPENDIX D
INDEX

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

1
FUNCTIONAL OVERVIEW

The configurable timer module (CTM) is an integral module of Motorola's family of modular microcontrollers. Members of this family are normally composed of several modules, interconnected by means of the intermodule bus (IMB). The CTM is unusual in the sense that it is, in itself, modular and is composed of submodules, making it easily configurable for different kinds of applications.

1.1 CTM features

- Modular architecture
- Counter submodules:
  - Clock prescaler
  - 16-bit free-running counter
  - 16-bit modulus counter
- Action submodules:
  - Single action input capture/output compare channels
  - Double action input capture/output compare channels, with PWM (pulse width modulation) mode
  - PWM channels
- I/O pin for each input capture/output compare
- Output-only pin for each PWM channel
- External clock input capability
- Interrupt capability on all capture/compare/PWM channels and on counter overflow conditions
- Two, three or four time base buses, allowing great flexibility in CTM configuration

1.2 CTM description

The highly modular architecture of the CTM is illustrated in Figure 1-1. Submodules are located on either side of the CTM's internal submodule bus (SMB). All data and control signals within the CTM are passed over this bus. The SMB is connected to the outside world via a special CTM submodule, known as the bus interface unit submodule (BIUSM), which is connected to the intermodule bus (IMB) and hence to the main CPU. This configuration allows the CPU to access the data and control registers in each CTM submodule on the SMB.

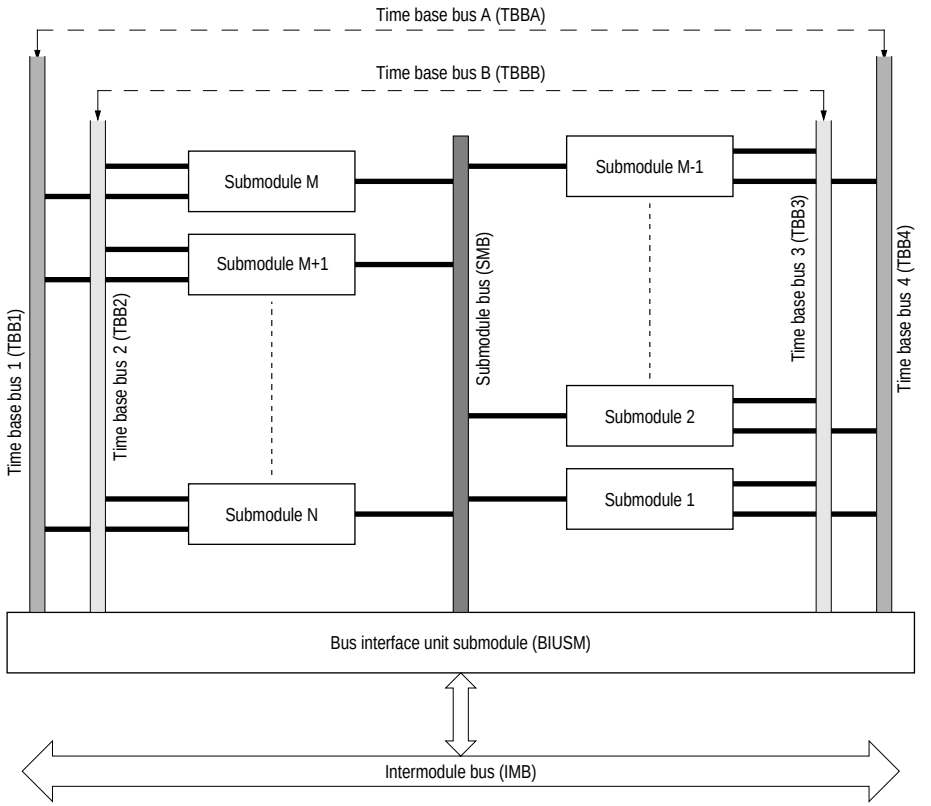


Figure 1-1 CTM architecture block diagram

Four local time base buses (TBB1 – TBB4) are arranged in such a way that each CTM submodule can be connected to two of them. As can be seen in Figure 1-1, CTM submodules numbered 1 to M-1 can be connected to TBB3 and TBB4 and CTM submodules M to N can be connected to TBB1 and TBB2. Control bits within each CTM submodule allow the software to connect the submodule to the desired time base bus(es). During the design of the CTM module, the four local time base

buses can be connected together (as shown by the dotted lines in Figure 1-1) to form two time base buses TBBA (TBB1/TBB4) and TBBB (TBB2/TBB3). The time base buses are each 16-bits wide and are used to transfer timing information from counters to action submodules. Each CTM submodule can either be a clock source module (and drive one or two of the time base buses) or an action submodule (and read and react to the timing information on the time base buses).

Every CTM module implementation must include at least a BIUSM and some form of clock submodule. All other submodules are optional and would be selected from a library of CTM submodules at the design stage, as required by the user to meet the needs of his application.

1.3 Byte/word/long word accesses

All CTM registers and data buses are 16 bits wide. Consequently, 16-bit (word) accesses are the normal case. 8-bit and 32-bit accesses are also permitted; however, as there is no pipelining in the CTM, 8-bit coherency is not supported.

1.3.1 8-bit (byte) accesses

8-bit accesses are illustrated in Figure 1-2 for even addresses and Figure 1-3 for odd addresses.

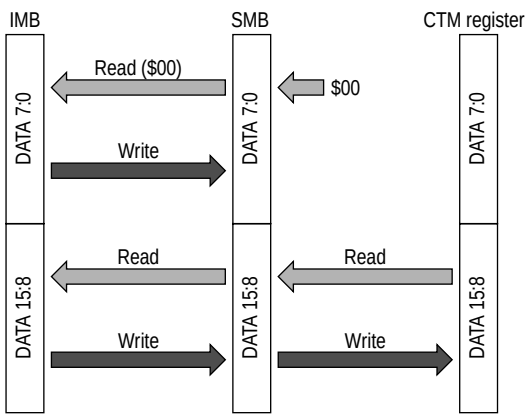


Figure 1-2 8-bit (byte) access (even addresses)

1.3.2 16-bit (word) aligned accesses

16-bit aligned access is the normal case and such accesses of counter or action submodule registers is coherent. This is illustrated in Figure 1-4.

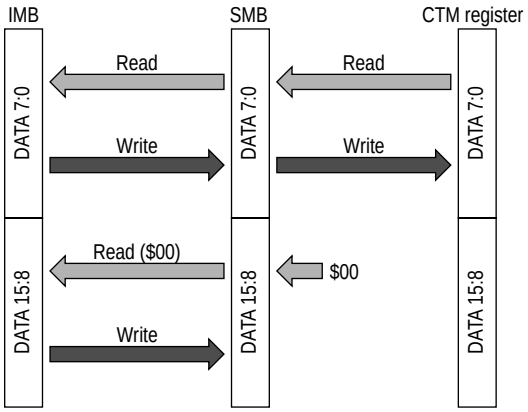


Figure 1-3 8-bit (byte) access (odd addresses)

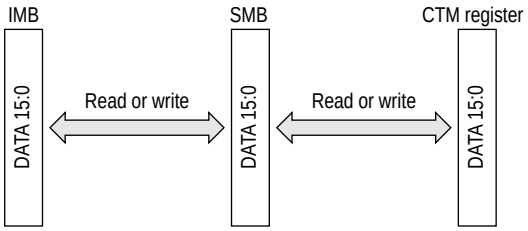


Figure 1-4 16-bit (word) aligned access

1.3.3 16-bit (word) misaligned accesses

A 16-bit misaligned access consists of two 8-bit accesses, the first to an odd address (see Figure 1-3), the second to the following even address (see Figure 1-2). A 16-bit misaligned access is treated by the BIU as an 8-bit odd address access. It is then the responsibility of the bus master to access the following byte. As in the case of the 8-bit access, since there is no pipelining, coherency is not supported.

Note: Neither CPU16 nor CPU32 (see section on reference documents at the beginning ot this document) support 16-bit misaligned accesses.

1.3.4 32-bit (long word) aligned accesses

A long word aligned access consists of two 16-bit aligned accesses (see Figure 1-4). When a long word access is attempted, a signal line is activated on the SMB during the access of the high order byte. This allows the CTM architecture to be compatible with submodules supporting long word coherency.

1.3.5 32-bit (long word) misaligned accesses

A long word misaligned access consists of three accesses: first a byte access to an odd address (see Figure 1-3), followed by a 16-bit aligned access to the following even address (see Figure 1-4), followed by a byte access to the remaining even address (see Figure 1-2). Note that the latter two accesses (16-bit aligned access followed by byte access to an even address) represent what is called a 3-byte access. As there is no pipelining, coherency is not supported.

Note: Neither CPU16 nor CPU32 support 32-bit misaligned accesses.

1.3.6 3-byte accesses

A 3-byte access is normally part of a long word misaligned access. It consists of a 16-bit aligned access (see Figure 1-4), followed by a byte access to the remaining even address (see Figure 1-2). A 3-byte access is treated by the BIUSM as a 16-bit aligned access. It is then the responsibility of the bus master to access the following byte. As there is no pipelining, coherency is not supported.

1.4 The time base bus system

The time base bus system is composed of four 16-bit buses: TBB1, TBB2, TBB3 and TBB4 (see Figure 1-1). Typically, TBB2 and TBB3 are tied together to form a global bus (TBBB) while TBB1 and TBB4 remain as partial buses (collectively called TBBA).

How the submodules are connected to these time base buses is different for each CTM configuration. This is shown generically in Figure 1-1, where all counter and action submodules in the right half of the diagram (numbered from 1 to M-1) can be connected to TBB3 and/or TBB4, and all submodules in the left half of the diagram (numbered from M to N) can be connected to TBB1 and/or TBB2. An example of how the time base buses are configured and how the submodules are connected to them in a practical CTM module (CTM2) is provided in Appendix B.

The time base buses are precharge/discharge type buses with wired-OR capability, so that no hardware damage occurs when several counters are driving the same bus at the same time.

Depending on software options, counter and action submodules located in the left half of Figure 1-1 (submodules M to N) can be connected to buses TBB1 or TBB2, while counter and action submodules located in the right half of Figure 1-1 (submodules 1 to M-1) can be connected to buses TBB3 and TBB4.

1.5 Pin descriptions

Input/output requirements are specific to each CTM submodule; pin allocation and functionality is described in the relevant sections of this document.

1.6 Input capture (IC) concepts

A typical 16-bit input capture function is shown in Figure 1-5. It has three basic parts: edge select logic, a 16-bit input capture latch and a 16-bit free-running counter. The edge select logic determines the input signal transition (rising or falling) that triggers the input capture circuitry. When the selected transition occurs, the contents of the counter are latched into the input capture latch. This action sets a status flag indicating that an input capture has occurred. An interrupt is generated if enabled. The value of the count latched or 'captured' is the time of the event. Because this value is stored in the input capture register when the actual event occurs, user software can respond to this event at a later time and determine the actual time of the event. However, this must be done prior to another input capture on the same pin; otherwise, the previous time value will be lost.

By recording the times for successive edges on an incoming signal, software can determine the period and/or pulse width of the signal. To measure a period, two successive edges of the same polarity are captured. To measure a pulse width, consecutive edges of opposing polarity are captured. For example, to measure the high time of a pulse, the input transition time is captured on the rising edge and subtracted from the time captured on the subsequent falling edge. When the period or pulse width is less than a full 16-bit counter overflow period, the measurement is very straightforward. In practice, however, software usually must keep track of the number of overflows of the 16-bit counter in order to extend the range.

Another typical use of the input capture function is to establish a time reference. In this case it may be used in conjunction with an output compare function in the same timer. For example, consider the case where it is required to generate an output signal transition a specific number of clock cycles after detecting an event (edge). The input capture function can be used to record the time at which the event occurred. A number corresponding to the desired delay can then be added to this captured value and stored in an output compare register. Because input capture and output compare functions are referenced to the same 16-bit counter, the delay can be controlled to the resolution of the free-running counter, independent of software latencies.

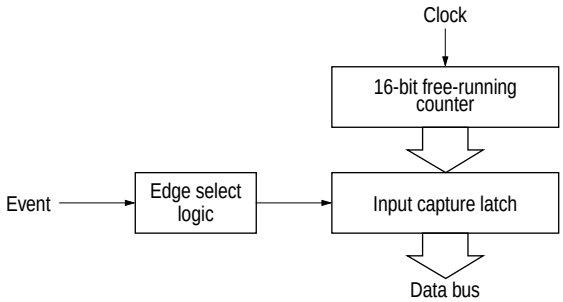


Figure 1-5 Simplified block diagram of 16-bit input capture

1.7 Output compare (OC) concepts

Output compare functions are used to cause events to occur at specific times, i.e. to cause signal transitions to occur on an output pin. A typical 16-bit output compare function is shown in Figure 1-6; it comprises a 16-bit compare register, a 16-bit comparator and a 16-bit free-running counter. When the value stored in the compare register matches the value of the free-running counter, the comparator sets an output compare flag.

Other events can occur when the output compare flag is set: an interrupt may be generated (if interrupts are enabled) and the logic levels on pins associated with the output compare function may change.

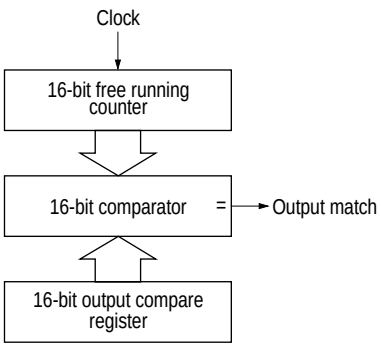


Figure 1-6 Simplified block diagram of 16-bit output compare

The output compare function can generate an output of a specific duration and polarity. A 16-bit value corresponding to the time a when a pin state change will occur is written to the output

compare register. The output compare function is configured to generate a high or low output automatically on the pin, or to toggle the state on the pin, when the match occurs. The output compare register is loaded with a new value after the compare occurs. Typically, more than one output compare function is associated with each pin; because pin state changes occur automatically at specific values of the free-running counter, the pulse width can be controlled to the resolution of the free-running counter independent of software latencies. A periodic pulse of a specific frequency and duty cycle can be generated by repeating the above steps.

1.8 Pulse accumulator (PA) concepts

Pulse accumulator systems are usually based on 8 or 16 bits. A typical 16-bit pulse accumulator is shown in Figure 1-7; it consists of a 16-bit counter and edge select logic, and has two modes of operation: event counting mode and gated mode. In event counting mode, the counter is incremented each time an event occurs. In gated mode, an internal clock source increments the counter while a selected level is present on the input pin (the gate). When the signal on the input pin is negated, the counter is stopped. Two status flags are available: one to indicate the occurrence of an event, and the other to indicate counter overflow. Either of these flags can cause the processor to be interrupted.

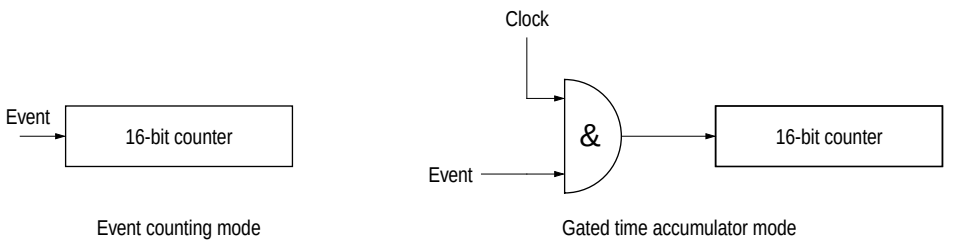


Figure 1-7 Simplified block diagram of a typical pulse accumulator

The pulse accumulator can be used, for example, to count the number of items going by on a conveyor belt or the number of teeth that have gone by on a crankshaft timing gear. As each item or tooth is detected, the counter is incremented (event counting mode). The counter therefore contains the number of items (or teeth). The flag indicates the occurrence of an event (an item or tooth went by). If interrupts are enabled, an interrupt is generated. Software can read the counter at this time.

The gated mode of operation can be used to measure the pulse width or period of an input signal. When the input to the pulse accumulator is active, the counter begins counting the input clock. When the signal is negated it stops counting. If the counter is set to zero before the pulse starts, the count value multiplied by the clock period gives the width of the input pulse to the nearest clock period. This could be used to determine how long a stimulus is present.

In an 8-bit pulse accumulator only 255 events can be counted before the counter overflows; the overflow flag can be used to extend the counter range beyond this value if required.

1.9 Pulse width modulation (PWM) concepts

A PWM waveform is created when the mark-to-space ratio of a periodic rectangular signal can be varied. If the waveform can be incrementally changed by 1/65536 of its period, it has 16 bits of resolution (see Figure 1-8).

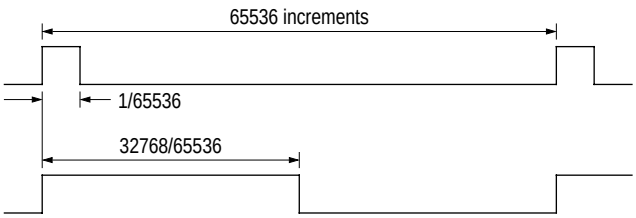


Figure 1-8 PWM example waveforms

A typical 16-bit PWM system (Motorola's General Purpose Timer, or GPT) is shown in Figure 1-9. Each time the counter overflows from $FFFF to $0000, the zero detector sets the output latch (output pin in high state). The zero detector is used as the reference to start the high time. As the counter is incremented, the counter value is compared with the contents of the pulse width register. When the comparator detects a match, the latch is reset. By changing the value in the pulse width register, the duty cycle is continuously variable in 1/65536 increments.

If the pulse width register contains $0000, the output latch will always be in the reset condition (output pin in low state). If the pulse width register is loaded with $0001, the output latch will be set for one count before being reset for the remainder of the period. If the register contains $8000 (32768 in decimal), the latch will be set for 32768 counts of the timer before being reset, resulting in a duty cycle of 50%. Provision is usually made to allow a 100% duty cycle (output latch always set; output pin always high) to be generated.

Varying the input clock frequency to the PWM counter also varies the period of the PWM signal.

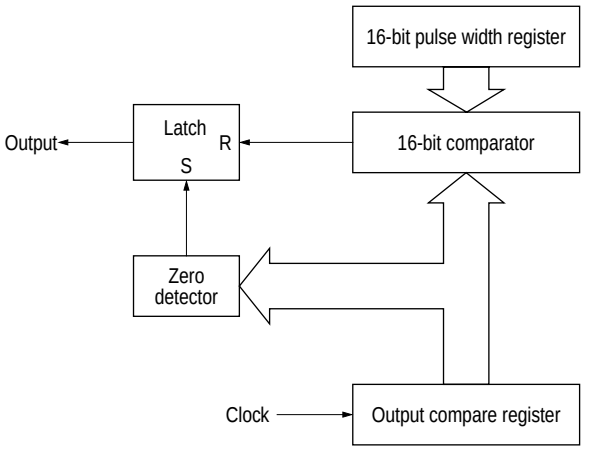


Figure 1-9 Simplified block diagram of a typical 16-bit PWM system

Typically, PWM systems are provided with increased flexibility via additional features such as output polarity selection, variable resolution and variable pulse periods.

1.10 Using and clearing flag bits

To clear any flag bit in the CTM, the software must first read the register containing the flag in question (usually the SIC register), then write a zero to the flag bit. These two steps do not have to be done on consecutive instructions. Writing a one to the flag bit has no effect.

Note: 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.

2
BUS INTERFACE UNIT SUBMODULE (BIUSM)

2.1 BIUSM description

The BIUSM connects the CTM's SMB to the IMB and allows the CTM submodules to communicate with the bus master (usually a CPU). The BIUSM also communicates interrupt requests, from the CTM submodules to the IMB, and transfers the interrupt level, arbitration bit and vector number to the CPU during the interrupt acknowledge cycle. The BIUSM contains a module configuration register, a time base bus register and a test register (for factory testing only).

2.2 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.

2.3 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.

2.4 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 Section 2.5.1).

2.5 BIUSM registers

The BIUSM register map comprises four 16-bit register locations. As shown in Table 2-1, 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.

Note: All BIUSM register addresses in this section are specified as offsets from the base address of the CTM.

Table 2-1 BIUSM register map

Address (1)	15 — 8	7 — 0
$00	BIUSM module configuration register (BIUMCR)	
$02	BIUSM test register (BIUTEST)	
$04	BIUSM time base register (BIUTBR)	
$06		

(1) Offset from the base address of the CTM.

2.5.1 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).

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIUMCR $00 (1)		STOP	FRZ		VECT7	VECT6	ARB2	ARB1	ARB0			TBRS1					TBRS0
	Reset:	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the CTM.

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).

The STOP bit is cleared by reset.

1 (set) – Stops operation of the CTM.

0 (clear) – Allows operation of the CTM.

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.

The FRZ bit is cleared by reset.

1 (set) – Halts the CTM sub module when the FREEZE signal appears on the IMB.

0 (clear) – Ignores the FREEZE signal on the IMB.

VECT7, VECT6 — Interrupt vector base number bits

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. This places the CTM vectors in one of four possible positions in the interrupt vector table, as follows.

VECT7	VECT6	Resulting vector base number
0	0	$00
0	1	$40
1	0	$80
1	1	$C0

Note: The reader should refer also to Section 3.4 and to the relevant CPU reference manuals for more detailed information on interrupt vector tables.

IARB[2:0] — Interrupt arbitration identification bits

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.

IARB[2:0] are all cleared by reset.

TBRS1, TBRS0 — Time base register bus select bits

These bits specify which time base bus is accessed when the time base register (BIUTBR) is read.

TBRS1	TBRS0	Time base bus
0	0	TBB1
0	1	TBB2
1	0	TBB3
1	1	TBB4

2.5.2 BIUTEST — BIUSM test configuration register

The BIUTEST register is located at CTM base address offsets $02 and $03 and is reserved for factory testing of the CTM.

2.5.3 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.

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIUTBR $04(1)		MSB								LSB							
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the CTM.

THIS PAGE INTENTIONALLY LEFT BLANK

3
INTERRUPTS

This section describes the interrupt functions of the CTM and its submodules and how these interrupts are passed to the CPU via the IMB. Interrupt requests from the CTM are treated as exceptions by the CPU and are dealt with by the CPU's exception processing routines. For a more detailed description of exception processing in IMB based microprocessors, please refer to the following Motorola publications:

- *CPU16 Central Processor Unit Reference Manual (CPU16RM/D)*
- *CPU32 Central Processor Unit Reference Manual (CPU32RM/AD)*

3.1 Interrupt levels on the IMB

The CTM and its submodules are capable of generating interrupts on eight different levels on the intermodule bus (IMB). Interrupt levels, arbitration and a hardwired daisy-chain priority system of the submodules in the CTM allow each of the many interrupt sources on the IMB to be uniquely identified and to have a unique vector address.

Each CTM submodule contains an interrupt control register that sets the interrupt priority for the submodule to one of eight levels (IL[2:0]). Level 7 is the highest priority level and level 0 disables interrupts. (Note that the CPSM and the BIUSM do not have the capability to generate interrupts and do not have interrupt vectors associated with them.)

When an interrupt is requested and is at a higher level than the current interrupt level set by the interrupt or exception level mask in the CPU's status register, the CPU starts an interrupt acknowledge (IACK) cycle. The CTM compares the interrupt level it requested with the interrupt level acknowledged by the CPU during the IACK cycle. If the levels match, arbitration with other modules requesting service on the same interrupt level begins.

An interrupt of the same level as the CPU's current interrupt or exception level mask cannot be executed until the mask level reduces below that level, except for level 7 interrupts. Level 7 is non maskable and exception processing on this level will be interrupted by other level 7 interrupts.

A higher level exception will interrupt a lower level exception routine, which must then wait until the exception mask returns to its original level before continuing.

3.2 Arbitration

The interrupt and exception processing system on IMB family devices is very similar to that used in the M68000 microprocessor family architecture, and is designed to support a very large number of interrupt sources. Within each of the eight interrupt request levels, defined by IL[2:0], there are sixteen different arbitration priority levels, defined by IARB[3:0], available to each IMB module. Level 15 is the highest arbitration priority level and level 1 is the lowest. Level 0 is a special case and is treated by the CPU as a spurious interrupt. Interrupting modules present their arbitration ID (IARB[3:0]) on the IMB and the module with the highest ID wins.

Note: Simultaneous interrupts on the same interrupt level are arbitrated on the basis of the four arbitration bits IARB[3:0]. Consequently, no two IMB modules may have the same IARB field value.

In the CTM, IARB[2:0] are contained within the BIUSM module configuration register (BIUMCR) and are common to all the CTM submodules, and each CTM submodule contains its own IL[2:0] and IARB3 bits. This allows each CTM submodule to request interrupts with one of two arbitration levels on any one of the 8 available interrupt levels. For example, if the IARB[2:0] bits in the BIUMCR are set to 101, each submodule can have an arbitration priority level of 0101 (5) or 1101 (13), depending on the state of the IARB3 bit.

3.3 CTM daisy-chain priority

To allow resolution between CTM submodule interrupts on the same interrupt level (IL[2:0]) and with the same arbitration priority (IARB3), the six hardwired vector bits (VECT[5:0]) in each submodule provide a hardware system of priority, or daisy-chain, within the CTM. The submodules are daisy-chained in descending order of their vector numbers, i.e. submodule 0 has the highest position in the daisy-chain and will win over all other submodules generating simultaneous interrupts on the same level with the same arbitration. The position of each submodule in the daisy-chain is specific to each different CTM variant. This is shown generically in Table 3-1. For an example of the daisy-chain structure of a specific CTM implementation, CTM2, see Appendix B.

3.4 Interrupt vector number and vector address

If the CTM wins an arbitration sequence, it generates a uniquely coded 8-bit interrupt vector number that indicates which timer submodule is requesting service. The two highest order bits (VECT[7:6]) of the interrupt vector number come from the BIUMCR and establish the vector base number of the CTM at $00, $40, $80 or $C0. The remaining bits of the interrupt vector number (VECT[5:0]) are hardwired into each CTM submodule and are unique for each interrupt source. The vector address is obtained by multiplying the vector base number by two. (See Table 3-1.)

Note: Some CTM submodules, e.g. the SASM, have more than one interrupt source and therefore have a corresponding number of vectors and uniquely coded vector base numbers.

Table 3-1 CTM submodule and interrupt vector number convention

Submodule number	Interrupt vector number (1) (VBN + vect[5:0])	Daisy-chain priority
0	VBN	Highest
1	VBN + 1	▲
2	VBN + 2	|
3	VBN + 3	|
4	VBN + 4	|
•	•	|
•	•	|
•	•	|
•	•	|
61	VBN + $3D	|
62	VBN + $3E	▼
63	VBN + $3F	Lowest

(1) VBN = $00, $3F, $7F or $BF depending on the state of the VECT[7:6] bits in the BIUSM.

THIS PAGE INTENTIONALLY LEFT BLANK

4
COUNTER PRESCALER SUBMODULE (CPSM)

4.1 CPSM description

The counter prescaler submodule (CPSM) is a programmable divider system that provides the CTM counters with a choice of six clock signals (PCLKx) derived from the main MCU system clock (f_{SYS}). The first counter prescaler stage generates PCLK1 by dividing f_{SYS} by 2 or by 3. The output of this first counter is then applied to an 8-bit prescaler which divides the clock signal further (by 2, 4, 8 and 16) to produce PCLK2, PCLK3, PCLK4 and PCLK5 (respectively). The division ratio for PCLK6 is software selectable (using the PSEL[1:0] control bits in the counter prescaler control register) from divide by 32, 64, 128 and 256. A block diagram of the CPSM is given in Figure 4-1. The clock division ratios available on PCLKx are also shown in the table in Section 4.3.1. These clock signals are provided on the SMB and may be used by any or all CTM submodules.

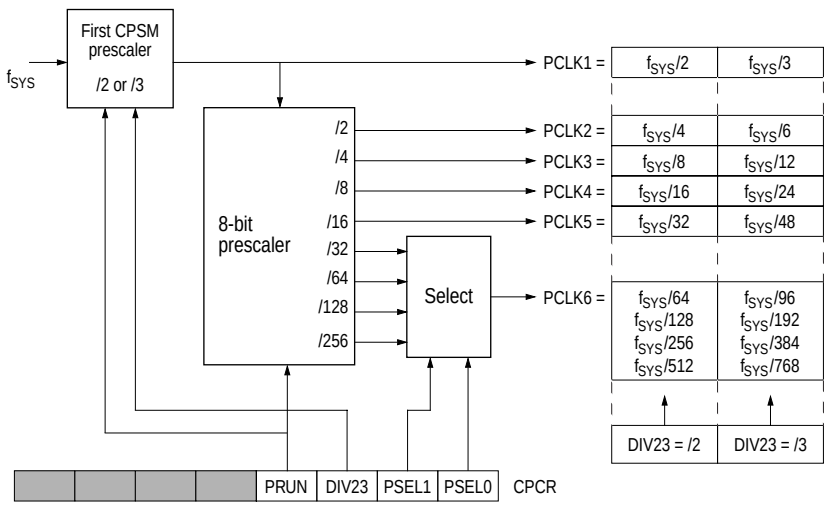


Figure 4-1 CPSM block diagram

4.2 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.

4

4.3 CPSM registers

The CPSM register map comprises four 16-bit register locations. As shown in Table 4-1, 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.

Note: All CPSM register addresses in this section are specified as offsets from the base address of the CTM.

Table 4-1 CPSM register map

Address (1)	15 — 8	7 — 0
$08	CPSM control register (CPCR)	
$0A	CPSM test register (CPTR)	
$0C		
$0E		

(1) Offset from the base address of the CTM.

4.3.1 CPCR — CPSM control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPCR $08(1)														PRUN	DIV23	PSEL1	PSEL0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the CTM.

PRUN — Prescaler running bit

The PRUN bit is a read/write control bit that allows the software to switch the prescaler counter on and off.

1 (set) – Prescaler is running.

0 (clear) – Prescaler divider is held in reset and is not running.

This bit allows the counters in various CTM submodules to be synchronized. It is cleared by reset.

DIV23 — Divide by 2 or divide by 3 bit

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.

1 (set) – First prescaler stage divides by 3.

0 (clear) – First prescaler stage divides by 2.

PSEL1, PSEL0 — Prescaler division ratio select bits

These control bits select the division ratio of the programmable prescaler output signal, PCLK6.

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

4.3.2 CPTR — CPSM test register

This test register is located at CTM address offsets $0A and $0B and is reserved for factory testing of the CPSM.

THIS PAGE INTENTIONALLY LEFT BLANK

5
FREE-RUNNING COUNTER SUBMODULE (FCSM)

5.1 FCSM description

The free-running counter submodule (FCSM) provides a multipurpose 'fixed' time base for use in a wide range of applications, such as input capture, output compare and PWM signal generation. The FCSM can also be configured to operate as an event counter; in this case, a flag is set after a predefined number of events (internal clocks or external events).

A block diagram of the FCSM is shown in Figure 5-1. 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.

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.

5.2 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 $FFFF to $0000. An overflow condition causes the COF flag bit in the FCSMSIC register to be set.

Note: Reset presets the counter register to $0000. Writing $0000 to the counter register while the counter's value is $FFFF does not set the COF flag and does not generate an interrupt request.

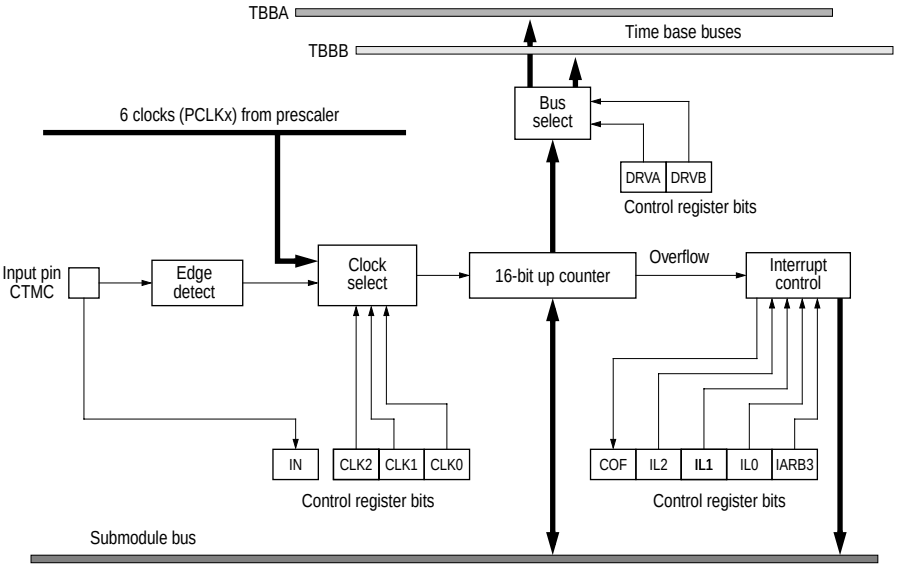


Figure 5-1 FCSM block diagram

5.3 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 Section 5.7.1). 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}).

5.3.1 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 $f_{SYS}/4$.

5.4 The FCSM time base bus driver

The DRVA and DRVB bits in the FCSMSIC register select the time base buses to be driven (see Section 5.7.1). Which of the time base buses is driven depends on where the FCSM is physically placed in any particular CTM implementation. See Section 1.4 for more information on the structure of the time base buses. For examples of FCSM waveforms and timings, please refer to Section 10.1.

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

5.5 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.

5.6 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 Section 5.7.1).

5.7 FCSM registers

The FCSM register map comprises four 16-bit register locations. As shown in Table 5-1, 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.

5

Table 5-1 FCSM register map

Address (1)	15 8	7 0
$00	Status, interrupt and control register (FCSMSIC)	
$02	Counter register(FCSMCNT)	
$04		
$06		

(1) Offset from the base address of the FCSM submodule.

5.7.1 FCSMSIC — FCSM status/interrupt/control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCSMSIC $00 (1)		COF	IL2	IL1	IL0	IARB3		DRVA	DRVB	IN					CLK2	CLK1	CLK0
	Reset:	0	0	0	0	0	0	0	0	u	0	0	0	0	0	0	0

(1) Offset from the base address of the FCSM submodule.

COF — Counter overflow flag bit

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 $FFFF to $0000. If the IL field is non-zero, an interrupt request is generated when the COF bit is set.

1 (set) – Counter overflow has occurred.

0 (clear) – Counter overflow has not occurred.

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 ‘one’) then write a ‘zero’ to the bit.

Note: 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.

IL[2:0] — Interrupt level bits

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.

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

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 (see Section 3). The IARB3 bit is cleared by reset.

DRVA, DRVB — Drive time base bus bits

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. (See Section 1.4 for information on the time base buses.)

DRVA	DRVB	Bus selected
0	0	Neither time base bus A nor time base bus B is driven
0	1	Time base bus B is driven
1	0	Time base bus A is driven
1	1	Both time base bus A and time base bus B are driven

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

IN — Input pin status bit

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.

CLK[2:0] — Counter clock select bits

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 $f_{SYS}/4$.

CLK2	CLK1	CLK0	Free running counter clock source
0	0	0	Prescaler output 1 (/2 or /3)
0	0	1	Prescaler output 2 (/4 or /6)
0	1	0	Prescaler output 3 (/8 or /12)
0	1	1	Prescaler output 4 (/16 or /24)
1	0	0	Prescaler output 5 (/32 or /48)
1	0	1	Prescaler output 6 (/64 to /512 or /96 to /768)
1	1	0	CTMC pin input, negative edge
1	1	1	CTMC pin input, positive edge

5.7.2 FCSMCNT — FCSM counter register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCSMCNT $02(1)		MSB								LSB							
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the FCSM submodule.

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

6
MODULUS COUNTER SUBMODULE (MCSM)

6.1 MCSM description

The MCSM is a versatile timer submodule capable of performing complex counting and timing functions, including modulus counting, in a wide range of applications. The MCSM may also be configured as an event counter, allowing the overflow flag to be set after a predefined number of events (internal clocks or external events), or as a variable time source for PWM generation. Note that the MCSM can also operate as a free running counter; in this case it behaves exactly like an FCSM. A block diagram of the MCSM is shown in Figure 6-1.

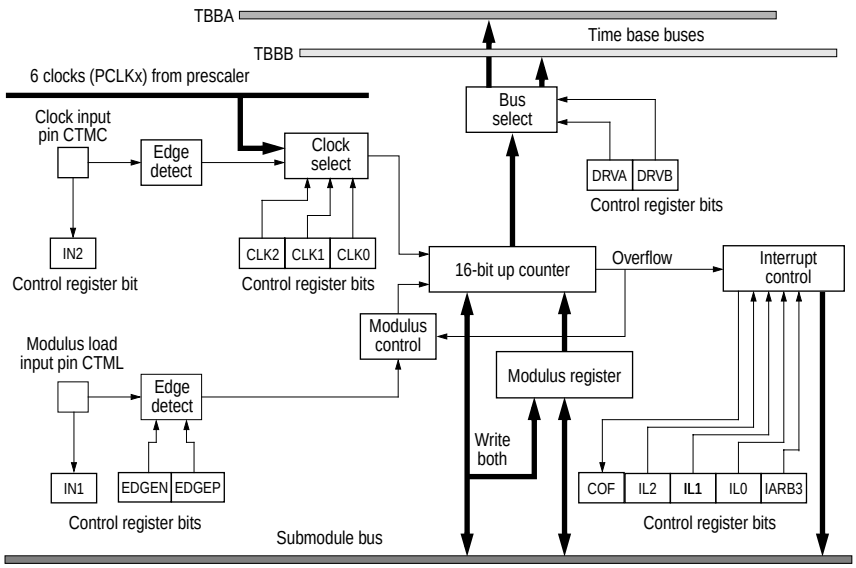


Figure 6-1 MCSM block diagram

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.

6.2 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 $0000 by reset.

6.3 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 Section 6.4).

Note: Reset presets the counter register to $0000. Writing $0000 to the counter register while its value is $FFFF does not set the COF flag and does not generate an interrupt.

6.3.1 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 $FFFF, the overflow flag is set on every counter clock pulse.

Loading of the counter register from the modulus register can also be triggered 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.

6.3.1.1 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 $0000.

6.4 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 Section 6.8.1). 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.

6.4.1 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 $f_{SYS}/4$.

6.5 The MCSM time base bus driver

The DRVA and DRVB bits in the MCSMSIC register select the time base buses to be driven (see Section 6.8.1). Which of the time base buses is driven depends on where the MCSM is physically placed in any particular CTM implementation. See Section 1.4 for information on the structure of the time base buses. For examples of MCSM waveforms and timings, please refer to Section 10.1.

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

6.6 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.

6.7 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 Section 6.8.1).

6.8 MCSM registers

The MCSM register map comprises four 16-bit register locations. As shown in Table 6-1, 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.

Note: All register addresses in this section are specified as offsets from the base address of the MCSM.

Table 6-1 MCSM register map

Address (1)	15 8	7 0
$00	MCSM status/interrupt/control register (MCSMSIC)	
$02	MCSM counter (MCSMCNT)	
$04	MCSM modulus latch (MCSMML)	
$06		

(1) Offset from the base address of the MCSM submodule.

6.8.1 MCSMSIC — MCSM status/interrupt/control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSMSIC $00 (1)		COF	IL2	IL1	IL0	IARB3		DRVA	DRVB	IN2	IN1	EDGEN	EDGEP		CLK2	CLK1	CLK0
	Reset:	0	0	0	0	0	0	0	0	u	u	0	0	0	0	0	0

(1) Offset from the base address of the MCSM submodule.

COF — Counter overflow flag bit

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 $FFFF to $xxxx, where $xxxx 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.

1 (set) – Counter overflow has occurred.

0 (clear) – Counter overflow has not occurred.

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 ‘one’) then write a ‘zero’ to the bit.

Note: 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.

IL[2:0] — Interrupt level bits

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.

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

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 (see Section 3). The IARB3 bit is cleared by reset.

DRVA, DRVB — Drive time base bus bits

DRVA and DRVB are read/write bits that control the connection of the MCSM to the time base buses A and B. These bits are cleared by reset. (See Section 1.4 for information on the time base buses.)

DRVA	DRVB	Bus selected
0	0	Neither time base bus A nor time base bus B is driven
0	1	Time base bus B is driven
1	0	Time base bus A is driven
1	1	Both time base bus A and time base bus B are driven

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

IN2 — Clock input pin status bit

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.

IN1 — Modulus load input pin status bit

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.

EDGEN, EDGEP — Modulus load edge sensitivity bits

These read/write bits select the sensitivity of the edge detection circuitry on the modulus load pin CTML.

EDGEN	EDGEP	IN1 edge detector sensitivity
0	0	None
0	1	Positive edge only
1	0	Negative edge only
1	1	Positive and negative edge

CLK[2:0] — Counter clock select bits

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 $f_{SYS}/4$.

CLK2	CLK1	CLK0	Free running counter clock source
0	0	0	Prescaler output 1 (/2 or /3)
0	0	1	Prescaler output 2 (/4 or /6)
0	1	0	Prescaler output 3 (/8 or /12)
0	1	1	Prescaler output 4 (/16 or /24)
1	0	0	Prescaler output 5 (/32 or /48)
1	0	1	Prescaler output 6 (/64 to /768)
1	1	0	CTMC pin input, negative edge
1	1	1	CTMC pin input, positive edge

6.8.2 MCSMCNT — MCSM counter register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSMCNT $02(1)		MSB								LSB							
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the MCSM submodule.

The MCSM counter register is a read/write register.

6.8.3 MCSMML — MCSM modulus latch register

(1) Offset from the base address of the MCSM submodule.

The MCSM modulus latch register is a read/write register.

7
SINGLE ACTION SUBMODULE (SASM)

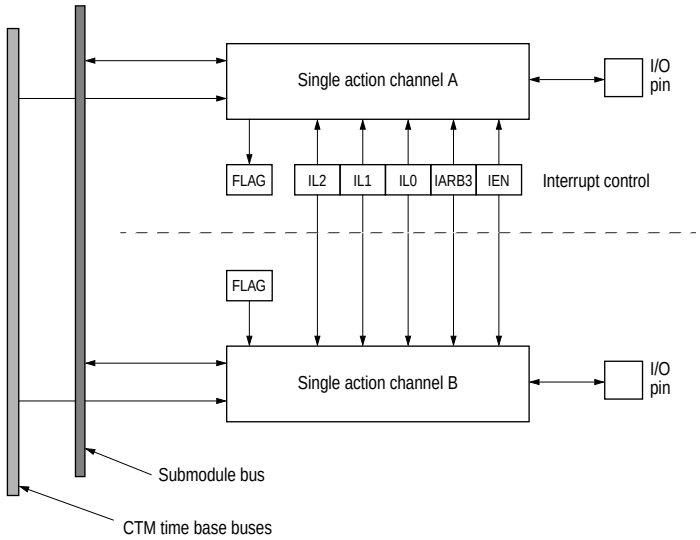


Figure 7-1 SASM block diagram

7.1 SASM description

The dual-channel SASM provides two identical single-action channels, each having its own input/output pin but sharing the same interrupt circuitry (see Figure 7-1). Each channel can be configured independently by the software to perform either input capture or output compare. The single action submodule is so called because each SASM channel can perform a single timing action (input capture or output compare) before some software intervention is required. Each channel can also work as a simple I/O pin.

A more detailed block diagram of a SASM channel is shown in Figure 7-2. Each 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 'zero').

7.2 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.

7.2.1 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.

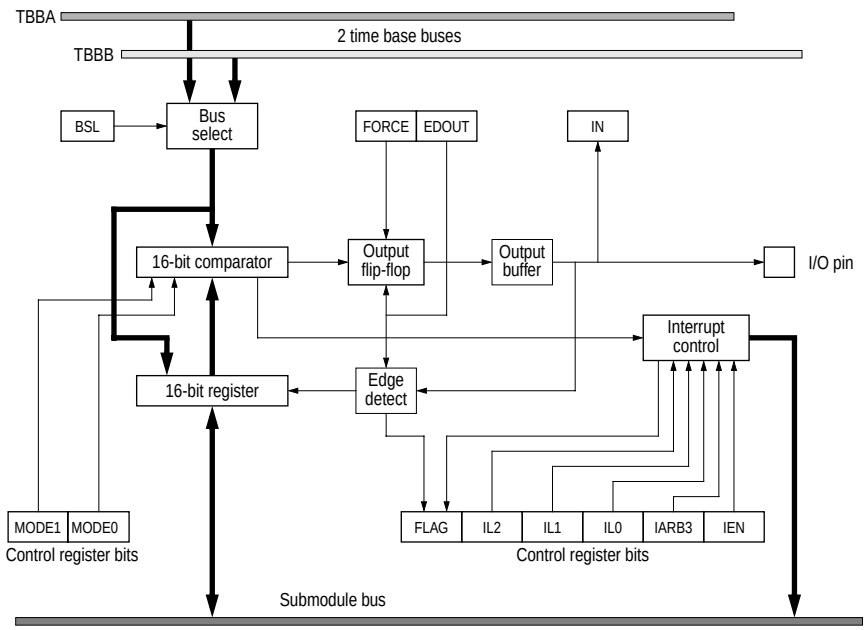


Figure 7-2 SASM block diagram (channel A)

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.

7.2.2 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.

Note: 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 Section 7.2.1). 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.

Note: 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.

7.2.3 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 Section 7.2.1). 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.

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

7.2.4 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 Section 7.2.1). 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.

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

7.2.5 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 Section 7.2.1). 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.

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

7.3 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.

7.4 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 Section 7.5.1 and Section 7.5.3). 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.

7.5 SASM registers

The SASM register map comprises eight 16-bit register locations. As shown in Table 7-1, 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 7-1 SASM register map

Address (1)	15 8	7 0
$00	SASM status/interrupt/control register A (SICA)	
$02	SASM data register A (SDATA)	
$04	SASM status/interrupt/control register B (SICB)	
$06	SASM data register A (SDATB)	
$08		
$0A		
$0C		
$0E		

(1) Offset from the base address of the SASM submodule.

7.5.1 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).

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SICA $00(1)		FLAG	IL2	IL1	IL0	IARB3	IEN		BSL	IN		FORCE	EDOUT			MODE1	MODE0
	Reset:	0	0	0	0	0	0	0	0	u	0	0	0	0	0	0	0

(1) Offset from the base address of the SASM submodule.

FLAG — Event flag bit

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.

1 (set) – An input capture or output compare event has occurred.

0 (clear) – An input capture or output compare event has not occurred.

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 ‘one’) then write a ‘zero’ to the bit.

Note: 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.

IL[2:0] — Interrupt level bits

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.

Note: These bits affect both SASM channels, not just channel A.

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

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 (see Section 3). The IARB3 bit is cleared by reset.

Note: This bit affects both SASM channels, not just channel A.

IEN — Interrupt enable bit

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.

1 (set) – Interrupts enabled.

0 (clear) – Interrupts disabled.

BSL — Time base bus select bit

This control bit selects the time base bus to be connected to SASM channel A. This bit is cleared by reset.

1 (set) – Time base bus B selected.

0 (clear) – Time base bus A selected.

IN — Input pin status bit

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.

FORCE — Force compare control bit

In the IC and OP modes, the FORCE bit is not used and writing to it has no effect.

In the OC and OCT modes, the FORCE bit is used by the software to cause the output flip-flop (and the output pin) to behave as though an output compare had occurred. In OC mode, setting the FORCE bit causes the value of EDOUT to be transferred to the output of the output flip-flop; in OCT mode, setting the FORCE bit causes the output flip-flop to toggle. 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.

1 (set) – Force output flip-flop to behave as if an output compare has just occurred.

0 (clear) – No action.

The FORCE bit is cleared by reset and always reads as zero.

Note: The FLAG bit is not affected by the use of the FORCE bit.

EDOUT — Edge detect and output level bit

In IC mode, the EDOUT bit is used to select the edge that will trigger the input capture circuitry.

1 (set) – Input capture on rising edge.

0 (clear) – Input capture on falling edge.

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.

The EDOUT bit is cleared by reset.

MODE1, MODE0 — SASM operating mode select bits

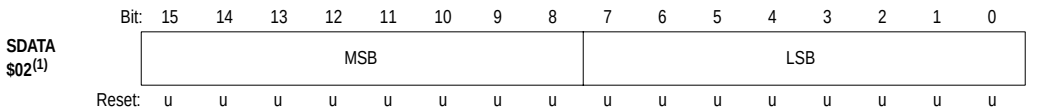
These control bits select the mode of operation of the SASM channel, as shown in the following table.

MODE1 and MODE0 are cleared by reset.

MODE1	MODE0	SASM channel operating mode
0	0	Input capture (IC)
0	1	Output port (OP)
1	0	Output compare (OC)
1	1	Output compare and toggle (OCT)

7.5.2 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.



(1) Offset from the base address of the SASM submodule.

7.5.3 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 Section 7.5.1).

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SICB $04(1)		FLAG							BSL	IN		FORCE	EDOUT			MODE1	MODE0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the SASM submodule.

7.5.4 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.

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDATB $06(1)		MSB								LSB							
	Reset:	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

(1) Offset from the base address of the SASM submodule.

THIS PAGE INTENTIONALLY LEFT BLANK

8
DOUBLE ACTION SUBMODULE (DASM)

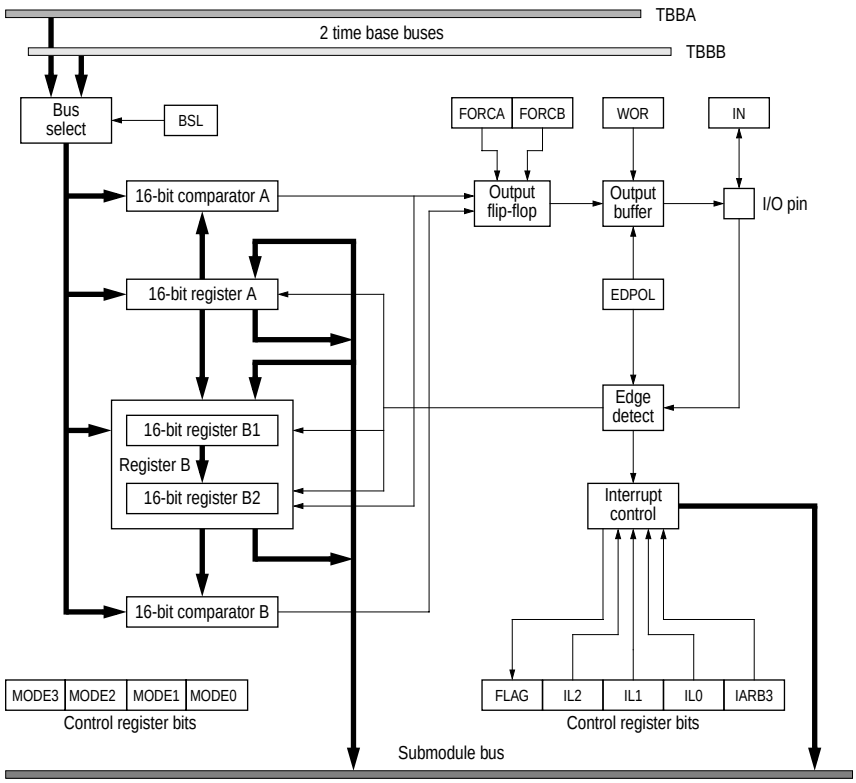


Figure 8-1 DASM block diagram

8.1 DASM description

The DASM is a timer submodule designed specifically to be integrated into CTM systems used in Motorola's M68300 and M68HC16 family MCUs. It contains two timing channels A and B associated with the same input/output pin. The dual action submodule is so called because its timing channel configuration allows two events (input capture or output compare) to occur before some software intervention is required.

Six operating modes allow the software to use the DASM's input capture and output compare functions to perform pulse width measurement, period measurement, single pulse generation and continuous pulse width generation, as well as standard input capture and output compare. The DASM can also work as a single I/O pin (see Table 8-1).

The DASM is composed of two timing channels (A and B), an output flip-flop, an input edge detector, some control logic and an interrupt section (see Figure 8-1). All control and status bits are contained in the DASMSIC register.

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.

8.2 32-bit coherent access

In the IPWM and IPM modes, 32-bit coherent access of the data registers is supported (see Section 1.3.4). 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.

8.3 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 8-1).

Table 8-1 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.

8.3.1 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.

8.3.2 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.

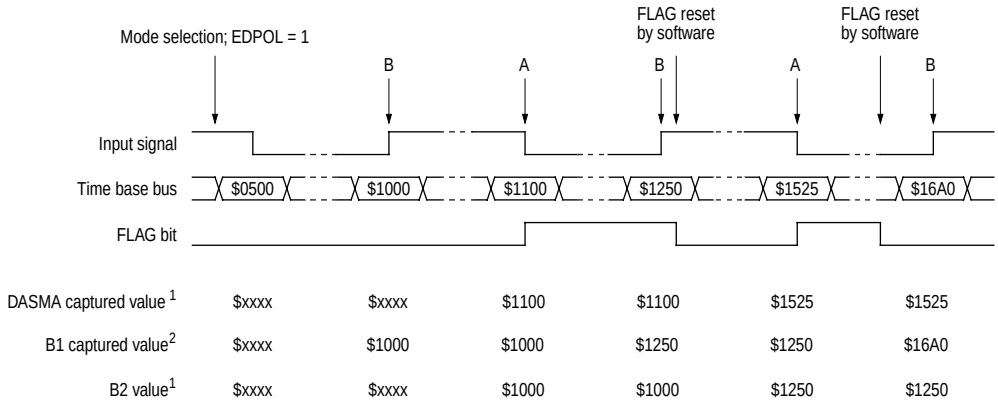
This mode also allows the software to determine the logic level on the input pin at any time by reading the IN bit in the DASMSIC register.

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 8-2 provides an example of how the DASM can be used for input pulse width measurement.



DASMA captured value [1]	$xxxx	$xxxx	$1100	$1100	$1525	$1525
B1 captured value[2]	$xxxx	$1000	$1000	$1250	$1250	$16A0
B2 value[1]	$xxxx	$xxxx	$1000	$1000	$1250	$1250

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

Figure 8-2 Input pulse width measurement example

8.3.3 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.

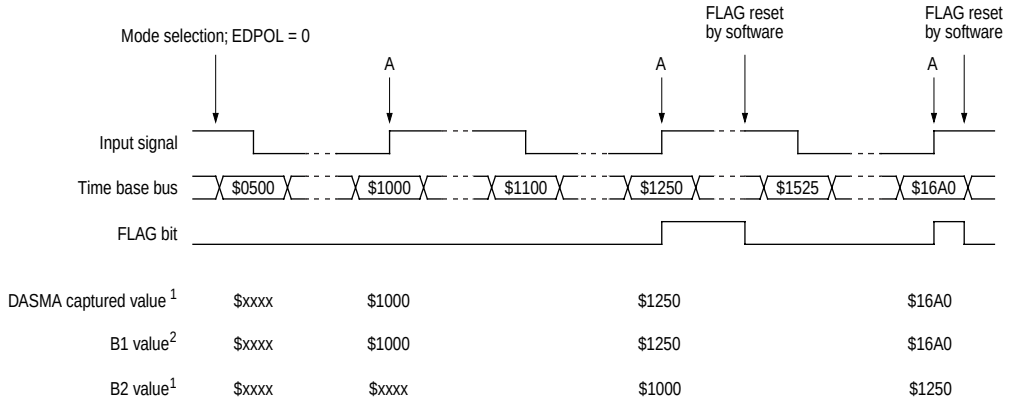
When the first edge having the selected polarity is detected, the time base bus value is latched into the 16-bit data register A, the data in register B1 is transferred to data register B2 and finally the data in register A is transferred to register B1. On this first capture the FLAG bit is not set. 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 8-3 provides an example of how the DASM can be used for input period measurement.



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

8

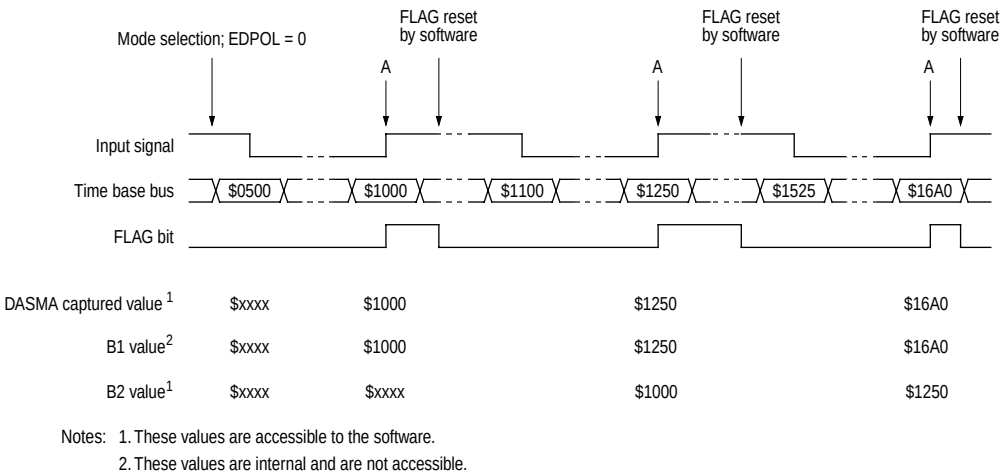
Figure 8-3 Input period measurement example

8.3.4 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 8-4 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 8-4 DASM input capture example

8.3.5 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 Section 8.6.1).

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.

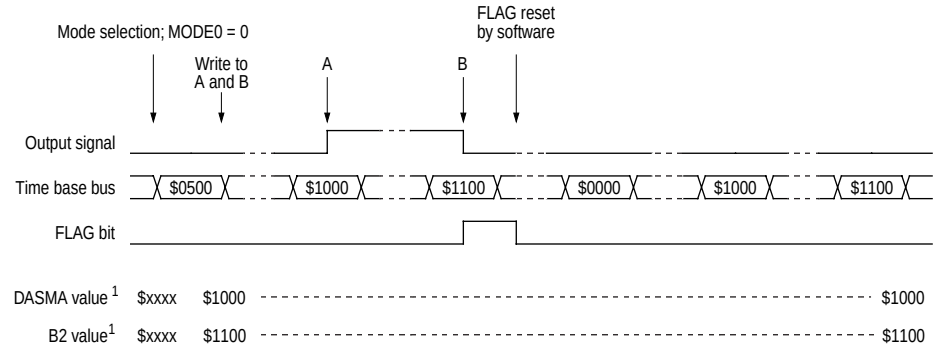
Note: If both channels are loaded with the same value they will try to force different levels on the output flip-flop. Hardware protection circuitry ensures that no contention occurs and the output flip-flop provides a logic zero level output.

8.3.5.1 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.

Note: In this mode, registers A and B2 are accessible to the user software (at consecutive addresses).

Figure 8-5 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 8-5 Single-shot output pulse example

8.3.5.2 Single output compare operation

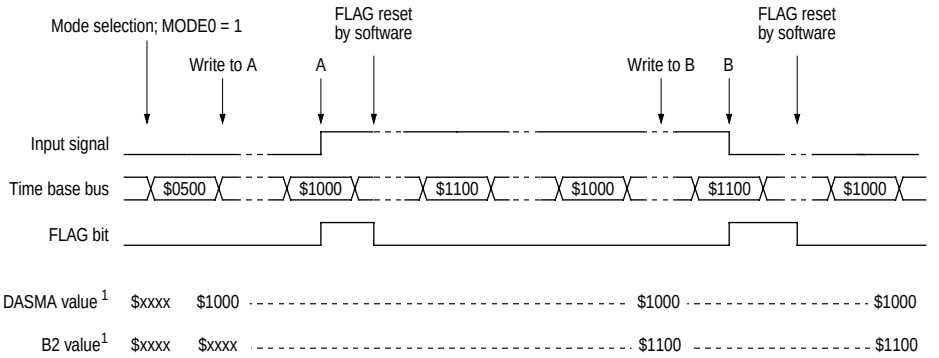
The single output compare operation is selected by writing to only one of the two data registers (A or B), thus enabling only one of the comparators. Following the first successful match on the enabled channel, the output level is fixed and remains at the same level indefinitely with no further software intervention being required.

Note: In this mode, registers A and B2 are accessible to the user software (at consecutive addresses).

Figure 8-6 provides an example of how the DASM can be used to perform a single output compare.

8.3.5.3 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.



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

Figure 8-6 Single shot output transition example

8.3.6 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.

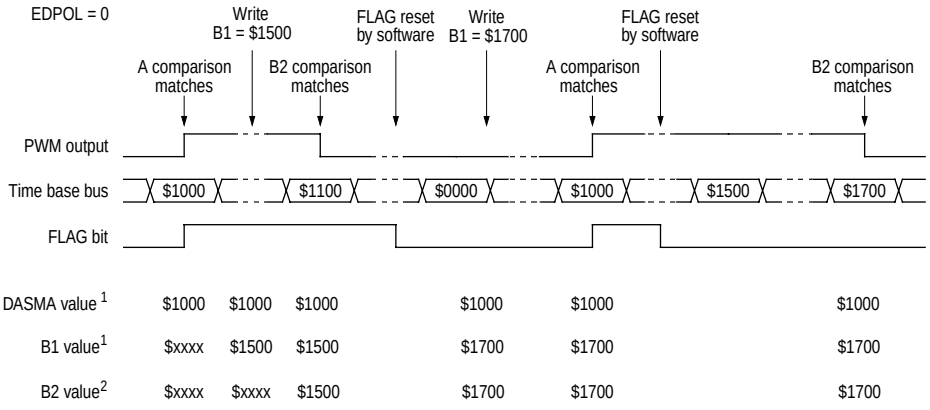
If subsequent compares occur on channels A and B, the PWM pulses continue to be output, regardless of the state of the FLAG bit.

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.

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.

Figure 8-7 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 8-7 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 the following equation (assuming the DASM is connected to a free running counter):

$$f_{PWM} = \frac{f_{SYS}}{N_{CPSM} \bullet N_{DASM}} \quad [1]$$

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 8-2.

Table 8-2 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

(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 one results in the output being ‘always set’. Clearing bit 15 (to zero) 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.

8.4 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.

8.5 Freeze action on the DASM

When the IMB FREEZE signal is recognized, the DASM captures and compares 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.

8.6 DASM registers

The DASM register map comprises four 16-bit register locations. As shown in Table 8-3, 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 nor 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 8-3 DASM register map

Address (1)	15 8	7 0
$00	DASM status/interrupt/control register (DASMSIC)	
$02	DASM register A (DASMA)	
$04	DASM register B (DASMB)	
$06		

(1) Offset from the base address of the DASM submodule.

8.6.1 DASMSIC — DASM status/interrupt/control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DASMSIC $00(1)		FLAG	IL2	IL1	IL0	IARB3		WOR	BSL	IN	FORCA	FORCB	EDPOL	MODE3	MODE2	MODE1	MODE0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the DASM submodule.

FLAG — Flag status bit

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.

1 (set) – An input capture or output compare event has occurred.

0 (clear) – An input capture or output compare event has not occurred.

- 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.

This flag bit is set only by the hardware and is cleared only by the software or by a system reset. The software can clear the FLAG bit either by writing a zero to it, having first read the bit as a one, or by selecting the DIS mode.

To clear the flag, the software must first read the bit (as ‘one’) then write a ‘zero’ to the bit.

Note: 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.

IL[2:0] — Interrupt level bits

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.

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

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 (see Section 3). The IARB3 bit is cleared by reset.

WOR — Wired-OR bit

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.

1 (set) – Output buffer is open-drain.

0 (clear) – Output buffer is totem pole.

The WOR bit is cleared by reset.

BSL — Bus select bit

This control bit selects the time base bus to be connected to the DASM.

1 (set) – The DASM is connected to time base bus B.

0 (clear) – The DASM is connected to time base bus A.

Note: The time base bus configurations (A and B) are specific to each CTM implementation (eg. CTM2). Please refer to the appropriate appendix for details.

IN — Input pin status bit

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.

FORCA — Force A bit

In the OCB, OCAB and OPWM modes, the FORCA bit allows the software to force the output flip-flop to behave as if a successful comparison had occurred on channel A (except that the FLAG bit is not set). Writing a one to FORCA sets the output flip-flop; writing a zero to it has no effect.

In the DIS, IPWM, IPM and IC modes, the FORCA bit is not used and writing to it has no effect.

FORCA is cleared by reset and always reads as zero.

Note: Writing a one to both FORCA and FORCB simultaneously resets the output flip-flop.

FORCB — Force B bit

In the OCB, OCAB and OPWM modes, the FORCB bit allows the software to force the output flip-flop to behave as if a successful comparison had occurred on channel B (except that the FLAG bit is not set). Writing a one to FORCB resets the output flip-flop; writing a zero to it has no effect.

In the DIS, IPWM, IPM and IC modes, the FORCB bit is not used and writing to it has no effect.

FORCB is cleared by reset and always reads as zero.

Note: Writing a one to both FORCA and FORCB simultaneously resets the output flip-flop.

EDPOL — Edge polarity bit

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.

1 (set) – Channel A captures on a falling edge.
Channel B captures on a rising edge.

0 (clear) – Channel A captures on a rising edge.
Channel B captures on a falling edge.

In the IPM and IC modes, the EDPOL bit is used to select the input capture edge sensitivity of channel A.

1 (set) – Channel A captures on a falling edge.

0 (clear) – 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.

1 (set) – 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.

0 (clear) – 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.

The EDPOL bit is cleared by reset.

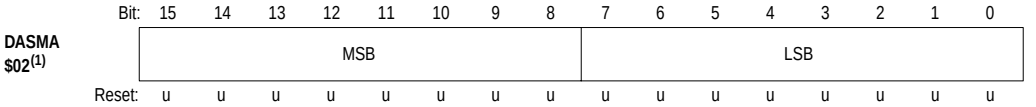
MODE[3:0] — Mode select bits

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.

The mode select bits are cleared by reset.

DASM control register bits				Bits of resolution	Time base bits ignored	DASM mode of operation
MOD3	MOD2	MOD1	MOD0			
0	0	0	0	—	—	DIS – Disabled
0	0	0	1	16	—	IPWM – Input pulse width measurement
0	0	1	0	16	—	IPM – Input period measurement
0	0	1	1	16	—	IC – Input capture
0	1	0	0	16	—	OCB – Output compare, flag on B compare
0	1	0	1	16	—	OCAB – Output compare, flag on A and B compare
0	1	1	0	—	—	
0	1	1	1	—	—	
1	0	0	0	16	—	OPWM – Output pulse width modulation
1	0	0	1	15	15	OPWM – Output pulse width modulation
1	0	1	0	14	15, 14	OPWM – Output pulse width modulation
1	0	1	1	13	15-13	OPWM – Output pulse width modulation
1	1	0	0	12	15-12	OPWM – Output pulse width modulation
1	1	0	1	11	15-11	OPWM – Output pulse width modulation
1	1	1	0	9	15-9	OPWM – Output pulse width modulation
1	1	1	1	7	15-7	OPWM – Output pulse width modulation

8.6.2 DASMA — DASM data register A



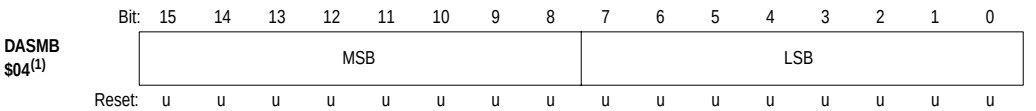
(1) Offset from the base address of the DASM submodule.

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.

8

8.6.3 DASMB — DASM data register B



(1) Offset from the base address of the DASM submodule.

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.

8.7 DASM examples

8.7.1 IC mode example

```
*****************************************************************************
*
*        DASM_IC (for CPU16 based devices)
*
*        Demonstration of the DASM CTM sub-module operating in IC mode.
*        The DASM is configured to capture the first falling input edge,
*        then generate an interrupt.
*
*        Timings assume 16.777MHz system clock
*
*****************************************************************************

* Set up the bus interface unit sub-module, BIUSM
         ldd   #$0C01            ; CTM not stopped, ignore FREEZE,
         std   BIUMCR            ; vector base =$4x, IARB2-0=4, BUS B displayed

* Set up the counter prescaler module, CPSM
         ldd   #$0008            ; Set PRUN to start prescaler and set
         std   CPCR              ; PCLK dividers to /2 /4 /8 /16 /32 and /64

* Set up the free running counter module, FCSM
         ldd   #$0905            ; No interrupts, arb3=1, timebase B driven
         std   fcsm25sic         ; Clock using PCLK6 (/64 clock, 3.8147µs)

         ldd   #$0900            ; MODE = %0000 = DIS
         std   dasm10sic         ; Disable DASM module before re-configuring

* Ensure that CPU will respond to a level 1 interrupt
         andp  #$FF1F            ; AND CCR with $FF1F to clear interrupt mask

* DASM IC mode initialization
         ldd   #$1913            ; MODE = %0011   Select IC mode
                                 ; EDPOL = 1      Capture -ve edge
                                 ; BSL = 1        Use time base bus B
                                 ; IARB3 = 1
                                 ; IL = %001      Lowest priority interrupt
         std   dasm10sic
loop     bra   loop              ; Hang here (until interrupt)

*****************************************************************************
*
*        IC1 interrupt routine.
*        The interrupt vector for the DASM module should
*        contain the entry address <ic1>
*
*****************************************************************************

ic1      bclr  dasm10sic,#$80    ; Clear DASM FLAG
         lde   dasm10a           ; Get the edge time into e
         rti                     ; Return from interrupt
```

8.7.2 IPM mode example

```
*************************************************************************
*
*        DASM_IPM (for CPU16 based devices)
*
*        Demonstration of the DASM CTM sub-module operating in IPM mode.
*        The DASM is configured to measure periods between input falling
*        edges. An interrupt is generated after each measured period.
*        The interrupt routine <ipm1> calculates the period result after
*        each interrupt.
*
*        Timings assume 16.777MHz system clock
*
*************************************************************************

* Set up the bus interface unit sub-module, BIUSM
         ldd   #$0C01            ; CTM not stopped, ignore FREEZE,
         std   BIUMCR            ; vector base =$4x, IARB2-0=4, BUS B displayed

* Set up the counter prescaler module, CPSM
         ldd   #$0008            ; Set PRUN to start prescaler and set
         std   CPCR              ; PCLK dividers to /2 /4 /8 /16 /32 and /64

* Set up the free running counter module, FCSM
         ldd   #$0905            ; No interrupts, arb3=1, timebase B driven
         std   fcsm25sic         ; Clock using PCLK6 (/64 clock, 3.8147µs)

         ldd   #$0900            ; MODE = %0000 = DIS
         std   dasm10sic         ; Disable DASM module before re-configuring

* Ensure that CPU will respond to a level 1 interrupt
         andp  #$FF1F            ; AND CCR with $FF1F to clear interrupt mask

* DASM IPM mode initialization
         ldd   #$1912            ; MODE = %0010   Select IPM mode
                                 ; EDPOL = 1      Measure between -ve edges
                                 ; BSL = 1        Use time base bus B
                                 ; IARB3 = 1
                                 ; IL = %001      Lowest priority interrupt
         std   dasm10sic
loop     bra   loop              ; Hang here (until interrupt)

*************************************************************************
*
*        IPM1 interrupt routine.
*        The interrupt vector for the DASM module should
*        contain the entry address <ipm1>
*
*************************************************************************

ipm1     bclr  dasm10sic,#$80    ; Clear DASM FLAG
         ldd   dasm10b           ; Get the period start time
         lde   dasm10a           ; ..and the period end time
result   sde                     ; Subtract to get the period width in e
         rti                     ; Return from interrupt
```

8.7.3 OCB mode example

```
*****************************************************************************
*
*        DASM_OCB (for CPU16 based devices)
*
*        Demonstration of the DASM CTM sub-module operating in OCB mode.
*        Direct read of time-base and FLAG polling also used.
*
*           pin         A                 B         A                 B..
*           forced       _________________           _________________
*           low         |                 |         |                 |
*           xxxxx       |                 |         |                 |
*           |___________|                 |_________|                 |_..
*           +---------->       $200      +---------->      $200
*               $100*   +----------------->   $100   +----------------->
*
*        The pin is initially forced low by selecting OCB mode with the EDPOL
*        clear. The current timebase value is then read from BIUTBR. The rising
*        edge (A) is scheduled $100 counts from this value, and the falling
*        edge is scheduled $200 counts after the rising edge.
*        The pulse train is maintained continuously by polling the DASM FLAG
*        bit. When it has been by a B compare another two compares are
*        scheduled.
*
*        * The time between forcing the pin low and the first rising edge is
*        slightly less than $100 counts. This is because of the software delay
*        between forcing the the pin low (by selecting OCB mode in DASMSIC) and
*        reading BIUTBR.
*
*        Timings assume 16.777MHz system clock
*
*****************************************************************************

* Set up the bus interface unit sub-module, BIUSM
        ldd   #$0C01            ; CTM not stopped, ignore FREEZE,
        std   BIUMCR            ; vector base =$4x, IARB2-0=4, BUS B displayed

* Set up the counter prescaler module, CPSM
        ldd   #$0008            ; Set PRUN to start prescaler and set
        std   CPCR              ; PCLK dividers to /2 /4 /8 /16 /32 and /64

* Set up the free running counter module, FCSM
        ldd   #$0905            ; No interrupts, arb3=1, timebase B driven
        std   fcsm25sic         ; Clock using PCLK6 (/64 clock, 3.8147µs)

* DASM OCB mode initialization
        ldd   #$0900            ; MODE = %0000 = DIS
        std   dasm10sic         ; Disable DASM module before re-configuring

        ldd   #$0904            ; MODE = %0100   Select OCB mode
                                ; EDPOL = 0      Generate positive pulse
                                ; FORCA,B = 0    Don't force pin now
                                ; BSL = 1        Use time base bus B
                                ; WOR = 0        Totem pole output
                                ; IARB3 = 1
                                ; IL = %000      Interrupts disabled
        std   dasm10sic
        bset  BIUMCR+1,#1       ; Allow timebase bus 2 to be read
```

```
        ldd   BIUTBR            ; Read current timer count
        addd  #$100
        std   dasm10a           ; Set rising edge $100 counts later
        addd  #$200
        std   dasm10b           ; Set falling edge $200 after rising edge

loop    brclr dasm10sic,#$80,loop; Wait for DASM FLAG to be set...

        bclr  dasm10sic,#$80    ; then clear
        ldd   dasm10b           ; Get previous falling edge time
        addd  #$100
        std   dasm10a           ; Set rising edge $100 counts later
        addd  #$200
        std   dasm10b           ; Set falling edge $200 after rising edge
        bra   loop              ; ..and wait for FLAG again
```

8.7.4 PWM mode example

```
*************************************************************************
*
*        DASM_PWM (for CPU16 based devices)
*
*        Demonstration of the DASM CTM sub-module operating in 7-bit PWM mode.
*        Data shown for generating PWM duty cycles from 0% to 100%
*
*        Timings assume 16.777MHz system clock
*
*************************************************************************

* Set up the bus interface unit sub-module, BIUSM
         ldd   #$0C01            ; CTM not stopped, ignore FREEZE,
         std   BIUMCR            ; vector base =$4x, IARB2-0=4, BUS B displayed

* Set up the counter prescaler module, CPSM
         ldd   #$0008            ; Set PRUN to start prescaler and set
         std   CPCR              ; PCLK dividers to /2 /4 /8 /16 /32 and /64

* Set up the free running counter module, FCSM
         ldd   #$0905            ; No interrupts, arb3=1, timebase B driven
         std   fcsm25sic         ; Clock using PCLK6 (/64 clock, 3.8147µs)

* DASM PWM mode initialization
         ldd   #$0900            ; MODE = %0000 = DIS
         std   dasm10sic         ; Disable DASM module before re-configuring

         ldd   #$090F            ; MODE = %1111   Select 7-bit OPWM mode
                                 ; EDPOL = 0      Generate positive pulse
                                 ; FORCA,B = 0    Don't force pin now
                                 ; BSL = 1        Use time base bus B
                                 ; WOR = 0        Totem pole output
                                 ; IARB3 = 1
                                 ; IL = %000      Interrupts disabled
         std   dasm10sic
         ldd   #$0000            ; A edge at $0000
         std   dasm10a

         ldd   #$0040            ; B edge at $0040, 64/128 = 50% duty cycle
*                                ; Other example PWM values:
*        ldd   #$0000            ; B edge at $0000, 0/128 = 0% duty cycle
*        ldd   #$0001            ; B edge at $0001, 1/128 = 0.78% duty cycle
*        ldd   #$007F            ; B edge at $007F, 127/128 = 99.2% duty cycle
*        ldd   #$8000            ; Special case, bit 15 set = 1002% duty cycle
         std   dasm10b
```

9
PULSE WIDTH MODULATION SUBMODULE (PWMSM)

The PWMSM is one of a family of submodules designed specifically for use in CTM systems in Motorola's M68300 and M68HC16 family microcontrollers.

9.1 PWMSM features

- Output pulse width modulated (PWM) signal generation with no software involvement
- Pulse width value provided by software:
  - double-buffered for glitch-free pulse width changes
  - 2-cycle minimum pulse width (e.g. 119 ns, for a 16.78 MHz MCU clock)
  - Up to 16-bit resolution on pulse width
- PWM period value provided by software:
  - double-buffered for glitch-free period changes
  - wide range of periods (e.g. 238 ns to 3 seconds, for a 16.67 MHz MCU clock)
- Maximum 50% duty cycle output frequency of 4.19 MHz (for a 16.78 MHz MCU clock)
- 0% and 100% duty cycles selected by software
- Optional interrupt after each pulse
- Output pulse polarity selected by software
- Output pin status can be read by software
- Output pins may be used as standard output port pins when PWM is not required

9.2 PWMSM description

The PWMSM allows pulse width modulated signals to be generated over a wide range of frequencies, independently of other CTM output signals. The output pulse width can vary from 0% to 100%, with 16 bits of resolution. The minimum pulse width is twice the minimum MCU system clock period (i.e., the minimum pulse width is 119 ns when using a 16.78 MHz clock).

The PWMSM is composed of:

- an output flip-flop with output polarity control,
- clock prescaler and selection logic,
- a 16-bit up-counter,
- two registers to hold the current and next pulse width values,
- two registers to hold the current and next pulse period values,
- a pulse width comparator,
- a system state sequencer,
- logic to create 0% and 100% pulses,
- interrupt logic,
- a status, interrupt and control register,
- and a submodule bus interface section.

The PWMSM includes its own time base counter and does not use the CTM time base buses, however the PWMSM does use the prescaled clock signal PCLK1 generated in the CPSM (see Section 4). A block diagram of the PWMSM is shown in Figure 9-1.

9

9.2.1 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.

9.2.2 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,

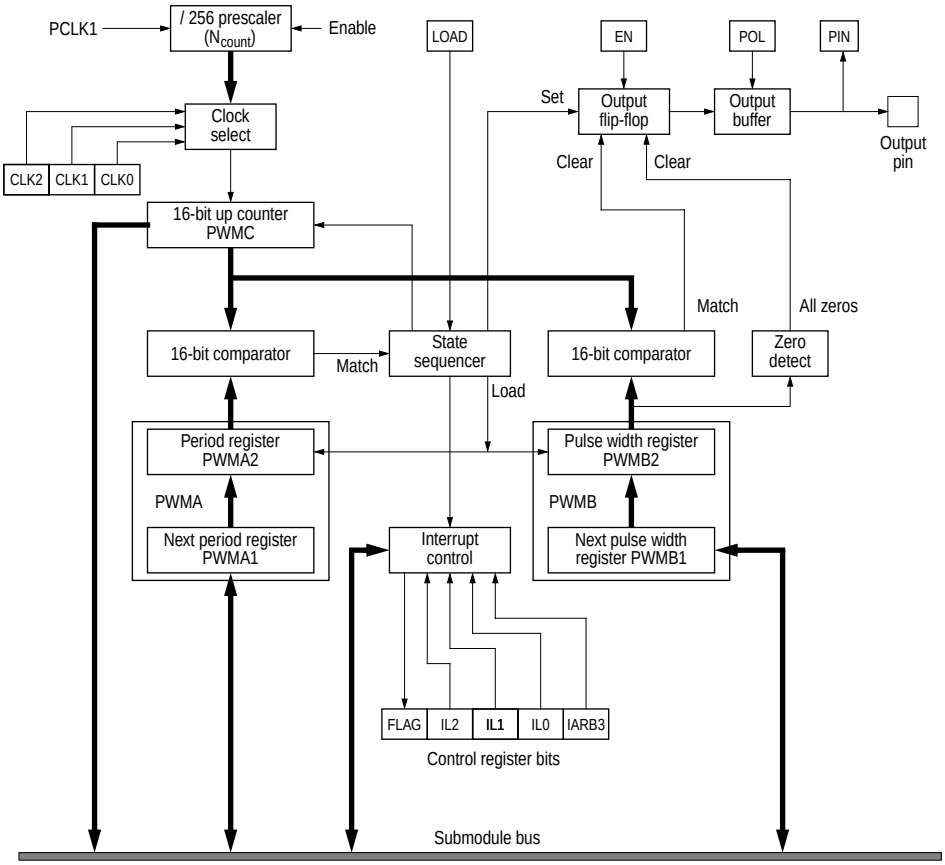


Figure 9-1 Pulse width modulation submodule block diagram

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).

9.2.3 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 $0001 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 $0001 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.

9.2.4 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 $0001.

Period values $0000 and $0001 are special cases. When PWMA2 contains $0000, an output period of 65536 PWM clock periods is generated.

When PWMA2 contains $0001, a period match occurs on every PWM clock period: the counter never increments beyond $0001 and the output level never changes.

Note: A value of $0002 in the period register and a value of $0001 in the pulse register are the conditions necessary to obtain the maximum possible output frequency for a given PWM clock period.

9

9.2.5 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 PWM SIC 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 = $0001). 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 = $FFFF and PWMA2 = $0000.

9.2.5.1 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 $0000. 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.

Note: 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 = $0000); 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.

Note: 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.

9.2.6 PWMSM coherency

Byte access of registers is discussed in Section 1.3.1, 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.

9.2.7 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.

9.2.8 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.

9.3 PWM frequency, pulse width and resolution

Table 9-1 and Table 9-2 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 9-1 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	4195kk
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 9-2 PWM pulse and frequency ranges (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

9.3.1 PWM frequency

The relationship between the PWM output frequency (f_{PWMO}) and the MCU system clock frequency (f_{SYS}) is given by Equation 1.

$$f_{PWMO} = \frac{f_{SYS}}{N_{CLOCK} \bullet N_{COUNTER}} \quad [1]$$

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

9.3.2 PWM pulse width

The minimum output pulse width (t_{PWMIN}) and the MCU system clock frequency (f_{SYS}) is given by Equation 2.

$$t_{PWMIN} = \frac{N_{CLOCK}}{f_{SYS}} \quad [2]$$

9.3.3 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 3.

$$PWMA1 = \frac{f_{SYS}}{N_{CLOCK} \bullet f_{PWMO}} \quad [3]$$

The value to be loaded into the PWM pulse width register (PWMB1) to obtain a given period is given by Equation 4.

$$PWMB1 = \frac{t_{PWMO}}{t_{PWMIN}} = \frac{\text{Duty cycle \%}}{100} \cdot PWMA1 \quad [4]$$

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

9.4 PWMSM register map and registers

The PWMSM register map comprises four 16-bit registers as shown in Table 9-3. 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 9-3 PWMSM register map

Address (1)	15 — 8	7 — 0
$00	Status, interrupt and control register (PWMSIC)	
$02	PWM period register(PWMA)	
$04	PWM pulse width register (PWMB)	
$06	PWM counter register (PWMC)	

(1) Offset from the base address of the PWMSM submodule.

9.4.1 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.

PWMSIC $00(1)	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		FLAG	IL2	IL1	IL0	IARB3				PIN		LOAD	POL	EN	CLK2	CLK1	CLK0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the PWMSM submodule.

FLAG — Period completion status bit

The FLAG bit is a status bit that indicates when the PWM output period has been completed.

1 (set) – PWM period completed.

0 (clear) – PWM period not 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.

Note: 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.

IL[2:0] — Interrupt level bits

The three interrupt level bits select the interrupt level of requests made by the PWMSM.

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

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 (see Section 3). The IARB3 bit is cleared by reset.

PIN — Output pin status bit

The PIN bit is a status bit that indicates the logic state present on the output pin.

1 (set) – Logic one state present on the output pin.

0 (clear) – Logic zero 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.

LOAD — Period and pulse width register load control bit

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.

1 (set) – Load period and pulse width registers.

0 (clear) – No action.

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 $0001,
- 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 $0000.

Note: Writing a one to the LOAD bit when the EN bit = 0, i.e. when the PWMSM is disabled, has no effect.

POL — Output pin polarity control bit

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 9-4).

Table 9-4 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

EN — PWMSM enable control bit

The EN bit is a control bit that allows the software to enable and disable the PWMSM as required.

1 (set) – Enable the PWMSM and start generation of PWM output pulses.

0 (clear) – Disable the PWMSM and stop 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 $0001,
- 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 $0001,
- 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).

Note: 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 = $0000).

CLK[2:0] — Clock rate selection bits

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 9-5 shows the counter clock sources and rates in detail.

9.4.2 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

Table 9-5 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)

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.

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMA $02(1)		MSB								LSB							
	Reset:	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

(1) Offset from the base address of the PWMSM submodule.

9.4.3 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.

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMB $04(1)		MSB								LSB							
	Reset:	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

(1) Offset from the base address of the PWMSM submodule.

9.4.4 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 $0001 on reset and is set to that value and held whenever the PWMSM is disabled (EN = 0).

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMC $06(1)		MSB								LSB							
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(1) Offset from the base address of the PWMSM submodule.

10
ELECTRICAL SPECIFICATIONS

10.1 FCSM and MCSM timing information

Table 10-1 FCSM timing characteristics

(V_{DD} = 5.0Vdc ± 10%, Vss = 0Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequency	f_{CLK}	0	16.67	MHz
Input pin frequency(1)	f_{PCNTR}	0	$f_{CLK}/4$	MHz
Input pin low time(1)	t_{PINL}	$2.0/f_{CLK}$	—	µs
Input pin high time(1)	t_{PINH}	$2.0/f_{CLK}$	—	µs
Clock pin to counter increment	t_{PINC}	$4.5/f_{CLK}$	$6.5/f_{CLK}$	µs
Clock pin to new TBB value	t_{PTBB}	$5.0/f_{CLK}$	$7.0/f_{CLK}$	µs
Clock pin to COF set ($FFFF)	t_{PCOF}	$4.5/f_{CLK}$	$6.5/f_{CLK}$	µs
Pin to IN bit delay	t_{PINB}	$1.5/f_{CLK}$	$2.5/f_{CLK}$	µs
Flag to IMB interrupt request	t_{FIRQ}	$1.0/f_{CLK}$	$1.0/f_{CLK}$	µs
Counter resolution(2)	t_{CRES}	—	$2.0/f_{CLK}$	µs

(1) Value applies when using external clock.

(2) Value applies when using internal clock. Minimum counter resolution depends on prescaler divide ratio selection.

Table 10-2 MCSM timing characteristics

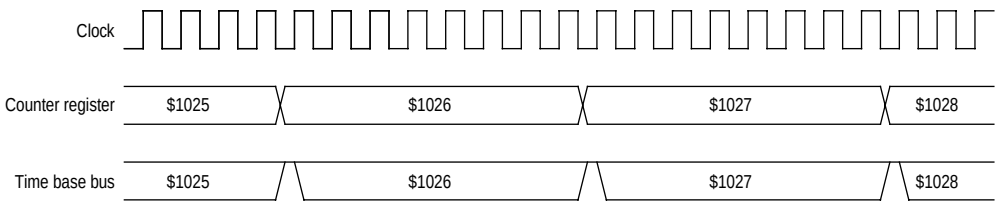
(V_{DD} = 5.0Vdc ± 10%, Vss = 0Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequency	f_{CLK}	0	16.67	MHz
Input pin frequency(1)	f_{PCNTR}	0	$f_{CLK}/4$	MHz
Input pin low time(1)	t_{PINL}	$2.0/f_{CLK}$	—	µs
Input pin high time(1)	t_{PINH}	$2.0/f_{CLK}$	—	µs
Clock pin to counter increment	t_{PINC}	$4.5/f_{CLK}$	$6.5/f_{CLK}$	µs
Clock pin to new TBB value	t_{PTBB}	$5.0/f_{CLK}$	$7.0/f_{CLK}$	µs
Clock pin to COF set ($FFFF)	t_{PCOF}	$4.5/f_{CLK}$	$6.5/f_{CLK}$	µs
Load pin to new counter value	t_{PLOAD}	$2.5/f_{CLK}$	$3.5/f_{CLK}$	µs
Pin to IN bit delay	t_{PINB}	$1.5/f_{CLK}$	$2.5/f_{CLK}$	µs
Flag to IMB interrupt request	t_{FIRQ}	$1.0/f_{CLK}$	$1.0/f_{CLK}$	µs
Counter resolution(2)	t_{CRES}	—	$2.0/f_{CLK}$	µs

(1) Value applies when using external clock.

(2) Value applies when using internal clock. Minimum counter resolution depends on prescaler divide ratio selection.

Not writing to counter register:



Writing to counter register:

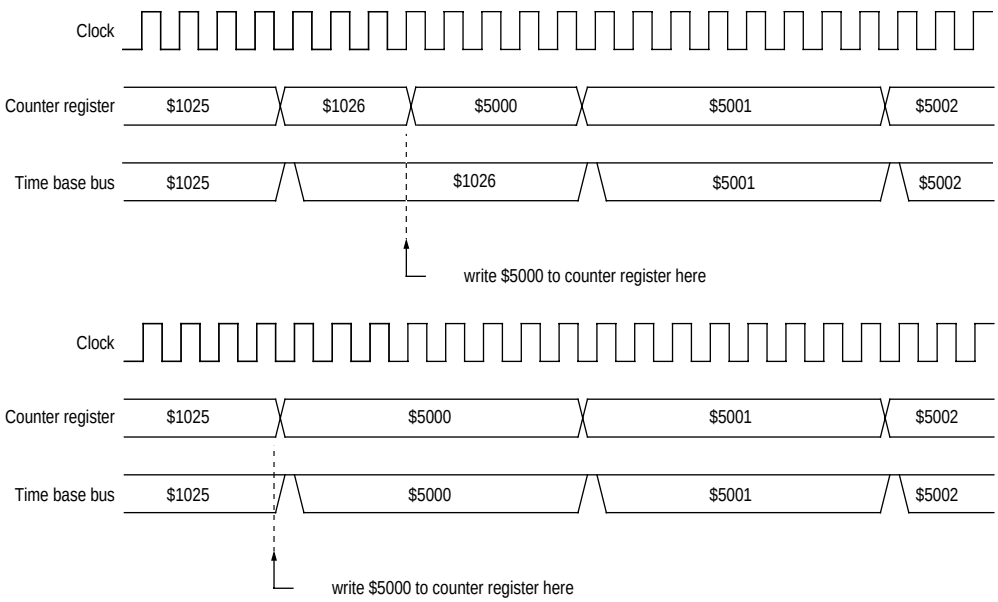


Figure 10-1 FCSM and MCSM time base timing diagram example

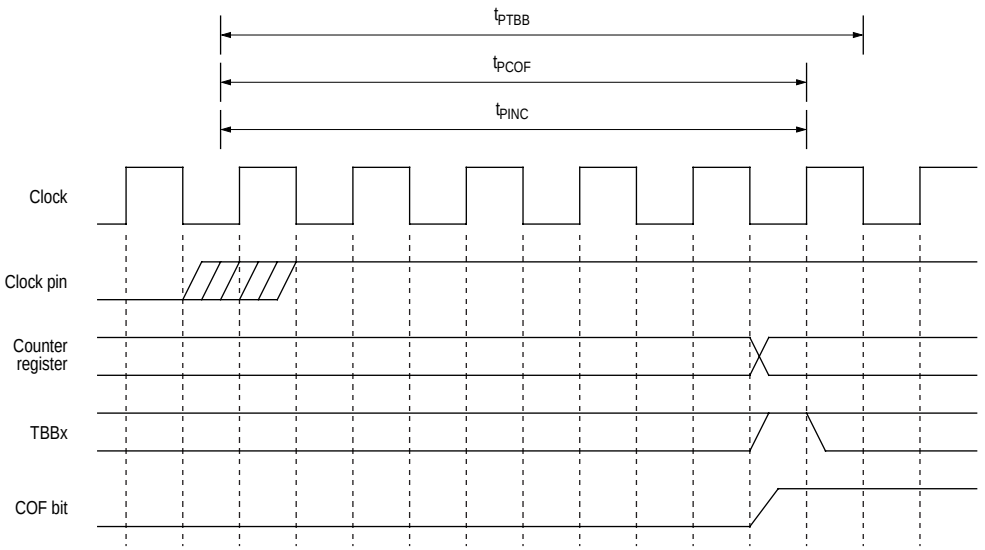


Figure 10-2 FCSM and MCSM clock pin to counter timing diagram

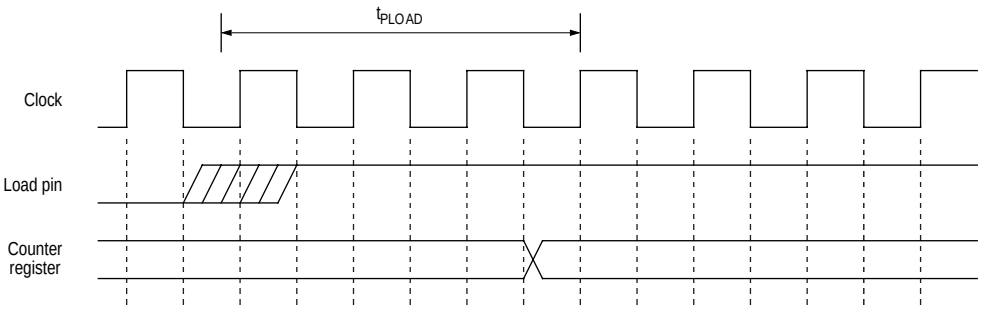


Figure 10-3 MCSM load pin to counter timing diagram

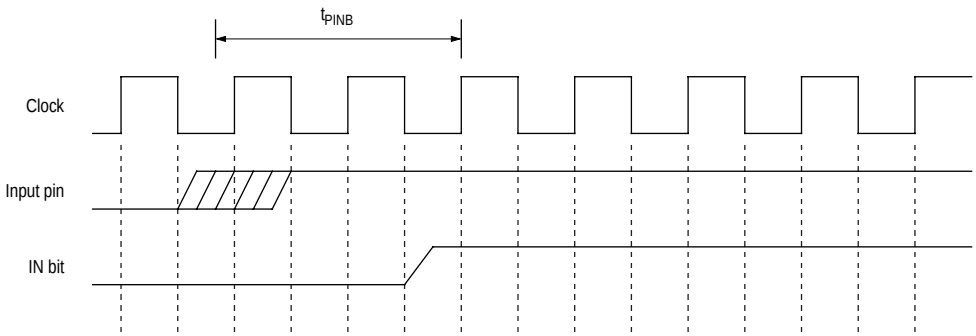


Figure 10-4 FCSM and MCSM pin to IN bit timing diagram

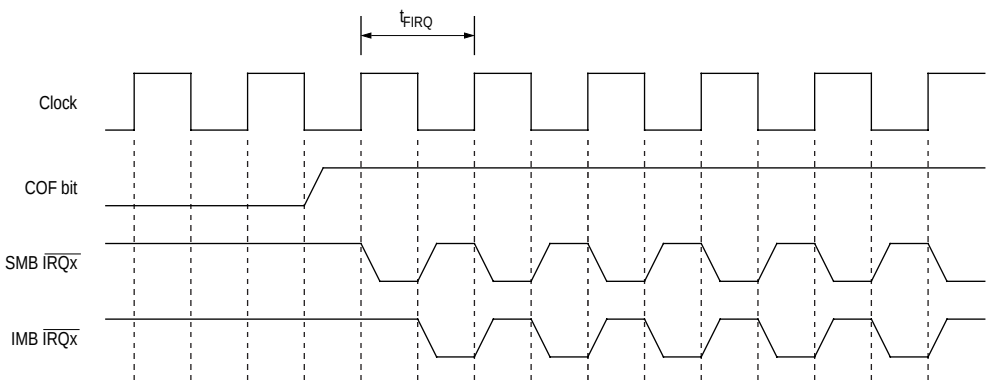


Figure 10-5 FCSM and MCSM COF bit to interrupt request timing diagram

10.2 SASM timing information

Table 10-3 SASM timing characteristics

(V_{DD} = 5.0Vdc ± 10%, Vss = 0Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequency	f_{CLK}	0	16.67	MHz
Input capture mode:				
Input pin low time	t_{PINL}	$2.0/f_{CLK}$	—	µs
Input pin high time	t_{PINH}	$2.0/f_{CLK}$	—	µs
Input capture resolution(1)	t_{RESCA}	—	$2.0/f_{CLK}$	µs
Pin to input capture delay	t_{PCAPT}	$2.5/f_{CLK}$	$4.5/f_{CLK}$	µs
Pin to FLAG set	t_{PFLAG}	$2.5/f_{CLK}$	$4.5/f_{CLK}$	µs
Pin to IN bit delay	t_{PINB}	$1.5/f_{CLK}$	$2.5/f_{CLK}$	µs
Output compare mode:				
OCT output pulse	t_{OCT}	$2.0/f_{CLK}$	—	µs
Compare resolution(1)	t_{RESCM}		$2.0/f_{CLK}$	µs
TBB change to FLAG set	t_{CFLAG}	$1.5/f_{CLK}$	$1.5/f_{CLK}$	µs
TBB change to pin change(2)	t_{CPIN}	$1.5/f_{CLK}$	$1.5/f_{CLK}$	µs
Flag to IMB interrupt request	t_{FIRQ}	$1.0/f_{CLK}$	$1.0/f_{CLK}$	µs

(1) Minimum resolution depends on counter and prescaler divide ratio selection

(2) Time given from when new value is stable on time base bus

10

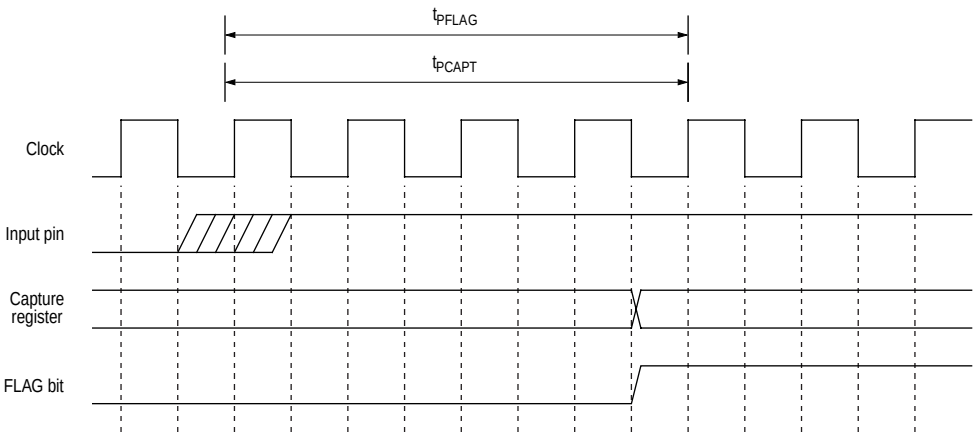


Figure 10-6 SASM input capture timing diagram

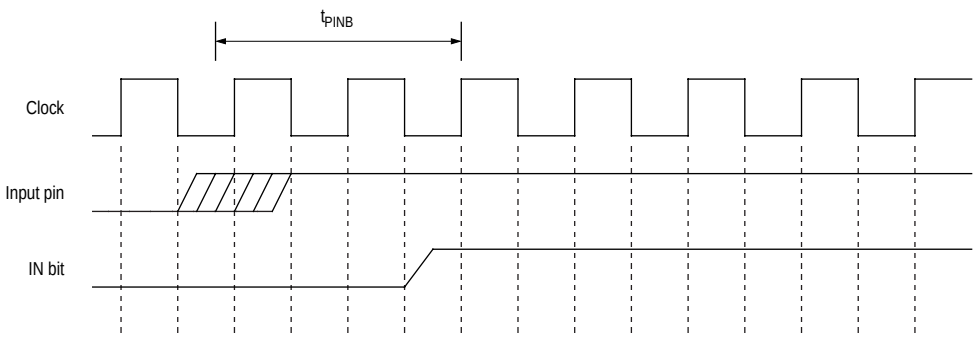


Figure 10-7 SASM pin to IN bit timing diagram

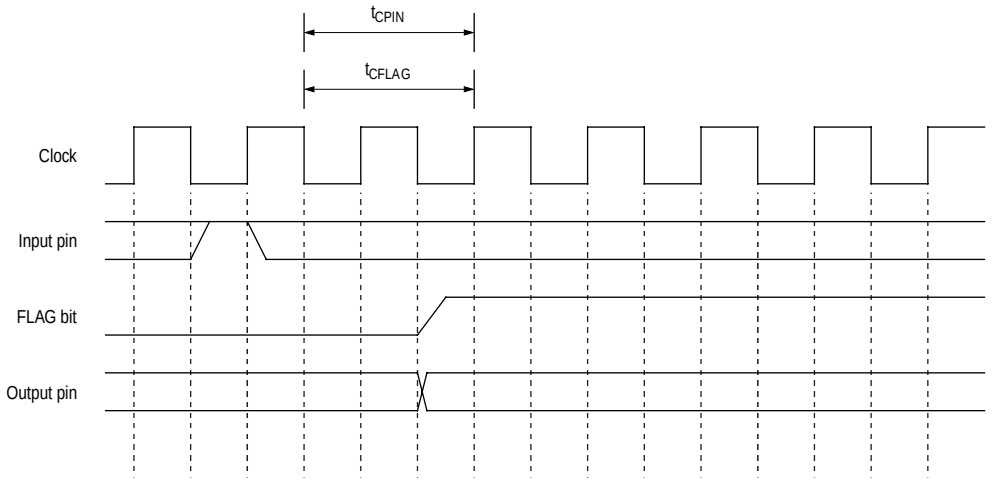


Figure 10-8 SASM output compare timing diagram

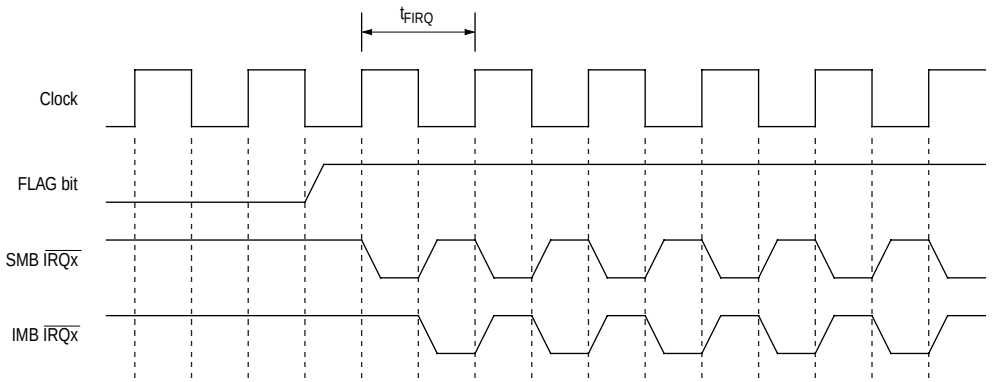


Figure 10-9 SASM FLAG bit to interrupt request timing diagram

10.3 DASM timing information

Table 10-4 DASM timing characteristics

(V_{DD} = 5.0Vdc ± 10%, Vss = 0Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequency	f_{CLK}	—	16.67	MHz
Input modes: (IPLM, IPM, IC):				
Input pin low time	t_{PINL}	$2.0/f_{CLK}$	—	µs
Input pin high time	t_{PINH}	$2.0/f_{CLK}$	—	µs
Input capture resolution(1)	t_{RESCA}	—	$2.0/f_{CLK}$	µs
Pin to input capture delay	t_{PCAPT}	$2.5/f_{CLK}$	$4.5/f_{CLK}$	µs
Pin to FLAG set	t_{PFLAG}	$2.5/f_{CLK}$	$4.5/f_{CLK}$	µs
Pin to IN bit delay	t_{PINB}	$1.5/f_{CLK}$	$2.5f_{CLK}$	µs
Output modes: (OC, OPWM):				
OCT output pulse	t_{OCT}	$2.0/f_{CLK}$	—	µs
Compare resolution(1)	t_{RESCM}	—	$2.0/f_{CLK}$	µs
TBB change to FLAG set	t_{CFLAG}	$1.5/f_{CLK}$	$1.5/f_{CLK}$	µs
TBB change to pin change(2)	t_{CPIN}	$1.5/f_{CLK}$	$1.5/f_{CLK}$	µs
Flag to IMB interrupt request(2)	t_{FIRQ}	$1.0/f_{CLK}$	$1.0/f_{CLK}$	µs

(1) Minimum resolution depends on counter and prescaler divide ratio selection.

(2) Time given from when new value is stable on time base bus.

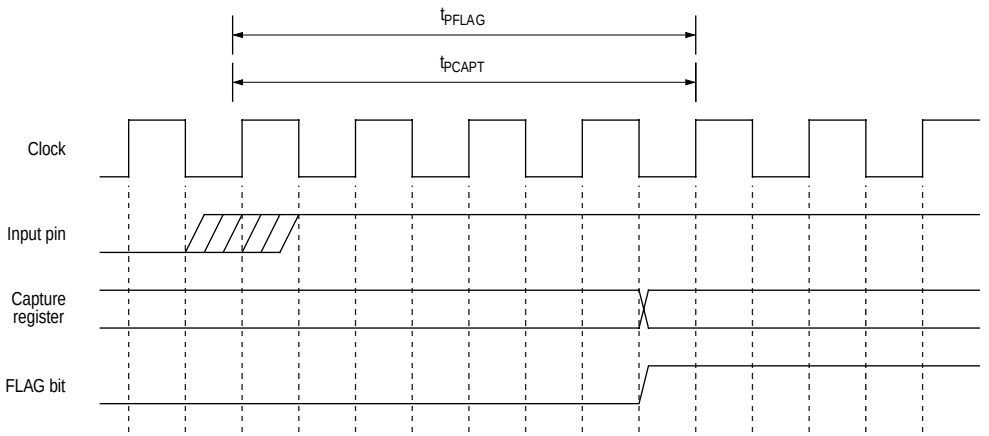


Figure 10-10 DASM input capture timing diagram

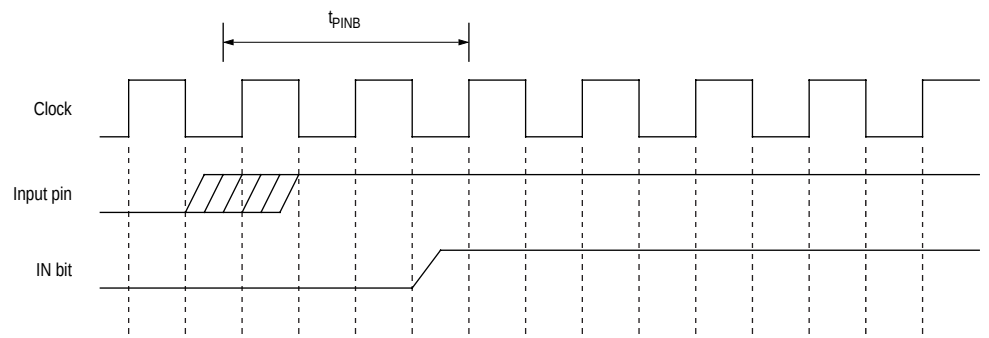


Figure 10-11 DASM pin to IN bit timing diagram

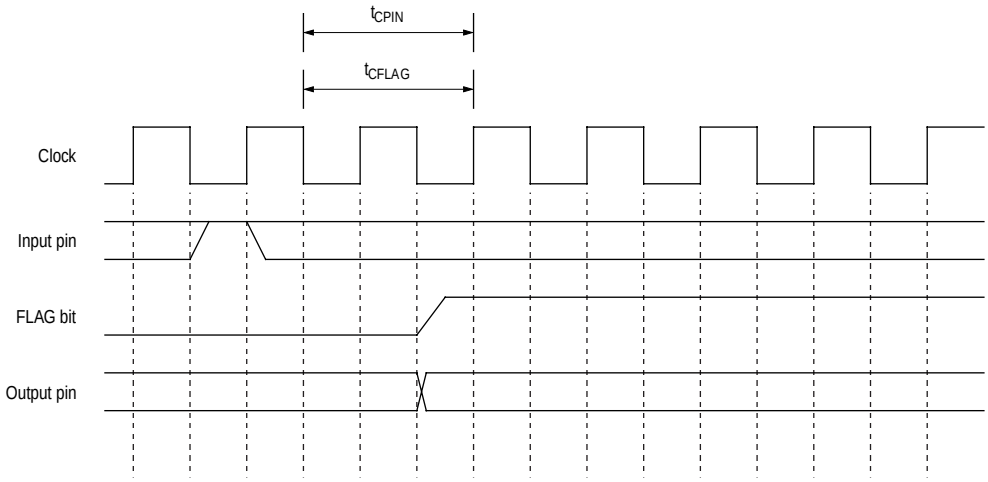


Figure 10-12 DASM output compare timing diagram

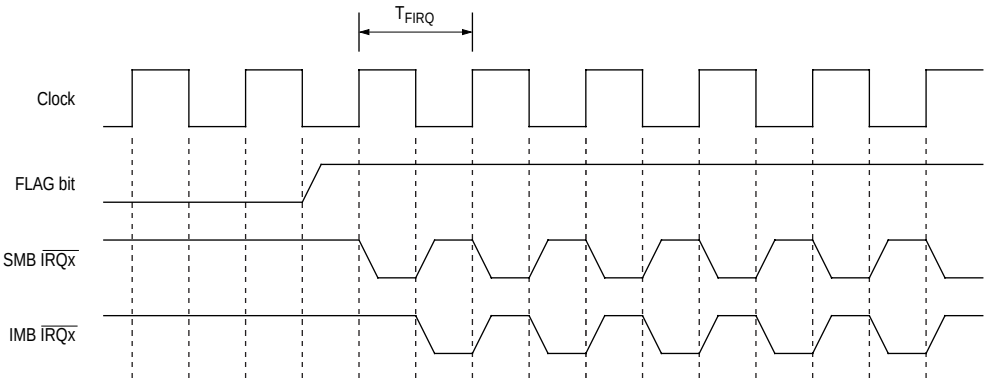


Figure 10-13 DASM FLAG bit to interrupt request timing diagram

10.4 PWMSM timing information

Table 10-5 PWMSM timing characteristics

(V_{DD} = 5.0Vdc ± 10%, Vss = 0Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequency	f_{CLK}	0	16.67	MHz
PWMSM output resolution (1)	t_{PWMR}	—	$2.0/f_{CLK}$	µs
PWMSM output pulse (2)	t_{PWMO}	$2.0/f_{CLK}$	—	µs
CPSM enable to output set	t_{PWMP} (3)	$3.5/f_{CLK}$		µs
	t_{PWMP} (4)	$6.5/f_{CLK}$		µs
PWM enable to output set	t_{PWME} (5)	$3.5/f_{CLK}$	$4.5/f_{CLK}$	µs
	t_{PWME} (6)	$5.5/f_{CLK}$	$6.5/f_{CLK}$	µs
FLAG to IMB interrupt request	t_{FIRQ}	$1.0/f_{CLK}$		µs

(1) Minimum output resolution depends on counter and prescaler divide ratio selection.

(2) Excluding the case where the output is always zero.

(3) With PWMSM enabled before enabling CPSM and DIV23 = 0.

(4) With PWMSM enabled before enabling CPSM and DIV23 = 1.

(5) With CPSM enabled before enabling PWMSM and DIV23 = 0.

(6) With CPSM enabled before enabling PWMSM and DIV23 = 1.

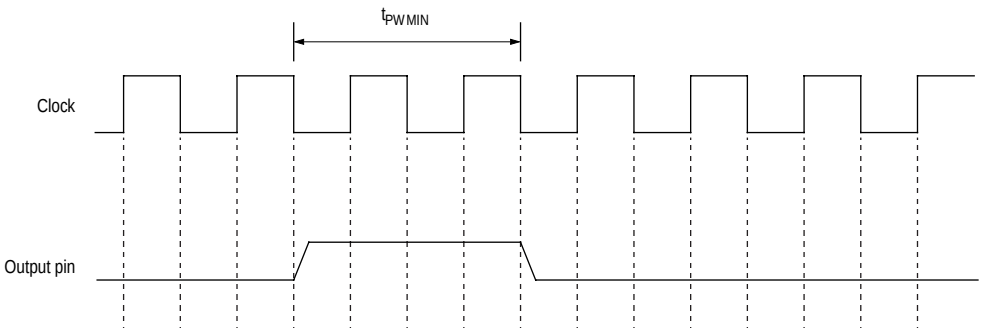


Figure 10-14 PWMSM minimum output pulse example timing diagram

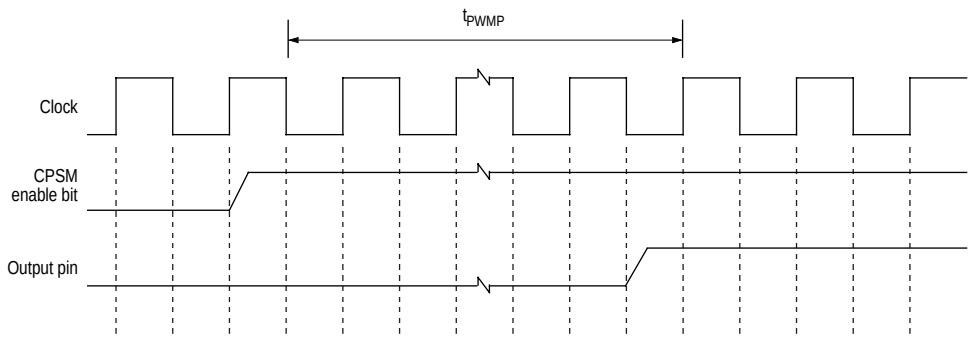


Figure 10-15 PWMSM CPSM enable to PWM output set timing diagram

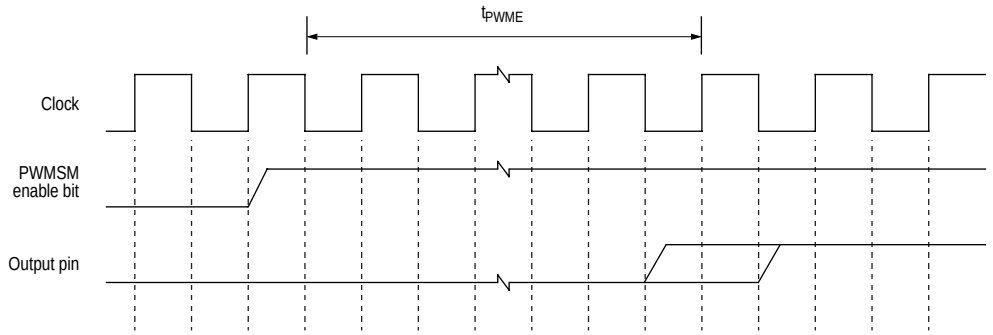


Figure 10-16 PWMSM enable to output set timing diagram

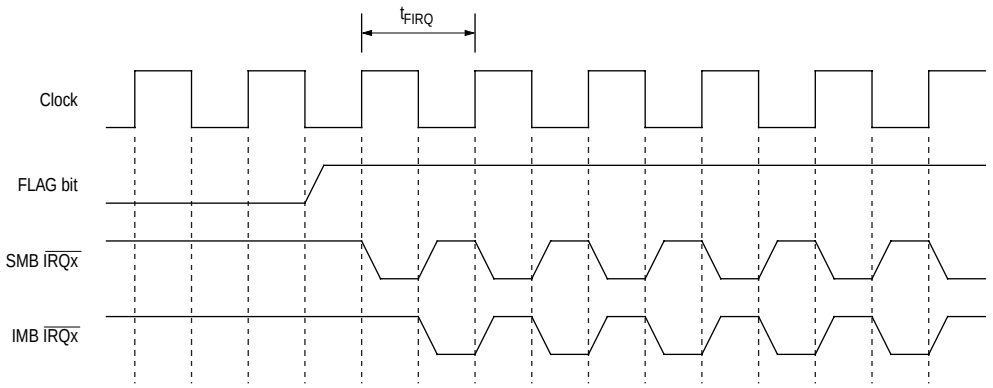
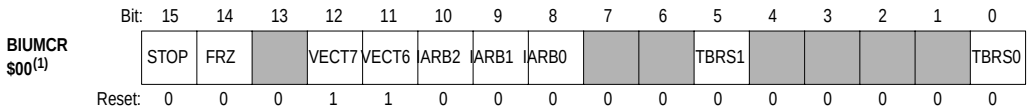


Figure 10-17 PWMSM FLAG bit to interrupt request timing diagram

A
REGISTER AND BIT SUMMARY

A.1 BIUSM registers and bits

A.1.1 BIUMCR — BIUSM module configuration register



(1) Offset from the base address of the CTM.

A.1.1.1 STOP — Stop enable

1 (set) – Stops operation of the CTM.

0 (clear) – Allows operation of the CTM.

A.1.1.2 FRZ — Freeze enable

1 (set) – Halts the CTM sub module when the FREEZE signal appears on the IMB.

0 (clear) – Ignores the FREEZE signal on the IMB.

A.1.1.3 VECT7, VECT6 — Interrupt vector base number bits

VECT7	VECT6	Resulting vector base number
0	0	$00
0	1	$40
1	0	$80
1	1	$C0

A.1.1.4 IARB[2:0] — Interrupt arbitration identification bits

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.

A.1.1.5 TBRS1, TBRS0 — Time base register bus select bits

TBRS1	TBRS0	Time base bus
0	0	TBB1
0	1	TBB2
1	0	TBB3
1	1	TBB4

A.1.2 BIUTEST — BIUSM test configuration register

Reserved for factory testing of the CTM.

A

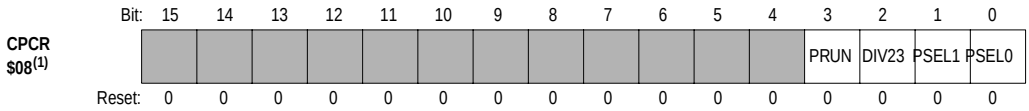
A.1.3 BIUTBR — BIUSM time base register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIUTBR $04(1)		MSB								LSB							
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the CTM.

A.2 CPSM registers and bits

A.2.1 CPCR — CPSM control register



(1) Offset from the base address of the CTM.

A.2.1.1 PRUN — Prescaler running bit

1 (set) – Prescaler is running.

0 (clear) – Prescaler divider is held in reset and is not running.

A.2.1.2 DIV23 — Divide by 2 or divide by 3 bit

1 (set) – First prescaler stage divides by 3.

0 (clear) – First prescaler stage divides by 2.

A.2.1.3 PSEL1, PSEL0 — Prescaler division ratio select bits

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

A.2.2 CPTR — CPSM test register

Reserved for factory testing of the CPSM.

A.3 FCSM registers and bits

A.3.1 FCSMSIC — FCSM status/interrupt/control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCSMSIC $00(1)		COF	IL2	IL1	IL0	IARB3		DRVA	DRVB	IN					CLK2	CLK1	CLK0
	Reset:	0	0	0	0	0	0	0	0	u	0	0	0	0	0	0	0

(1) Offset from the base address of the FCSM submodule.

A.3.1.1 COF — Counter overflow flag bit

1 (set) – Counter overflow has occurred.

0 (clear) – Counter overflow has not occurred.

A.3.1.2 IL[2:0] — Interrupt level bits

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

A

A.3.1.3 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.

A.3.1.4 DRVA, DRVB — Drive time base bus bits

DRVA	DRVB	Bus selected
0	0	Neither time base bus A nor time base bus B is driven
0	1	Time base bus B is driven
1	0	Time base bus A is driven
1	1	Both time base bus A and time base bus B are driven

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

A.3.1.5 IN — Input pin status bit

This read-only status bit reflects the logic state of the FCSM input pin CTMC

A.3.1.6 CLK[2:0] — Counter clock select bits

CLK2	CLK1	CLK0	Free running counter clock source
0	0	0	Prescaler output 1 (÷ 2 or ÷ 3)
0	0	1	Prescaler output 2 (÷ 4 or ÷ 6)
0	1	0	Prescaler output 3 (÷ 8 or ÷ 12)
0	1	1	Prescaler output 4 (÷ 16 or ÷ 24)
1	0	0	Prescaler output 5 (÷ 32 or ÷ 48)
1	0	1	Prescaler output 6 (÷ 64 to ÷ 768)
1	1	0	CTMC pin input, negative edge
1	1	1	CTMC pin input, positive edge

A.3.2 FCSMCNT — FCSM counter register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCSMCNT $02(1)		MSB								LSB							
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the FCSM submodule.

A

A.4 MCSM registers and bits

A.4.1 MCSMSIC — MCSM status/interrupt/control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSMSIC $00(1)		COF	IL2	IL1	IL0	IARB3		DRVA	DRVB	IN2	IN1	EDGEN	EDGEP		CLK2	CLK1	CLK0
	Reset:	0	0	0	0	0	0	0	0	u	u	0	0	0	0	0	0

(1) Offset from the base address of the MCSM submodule.

A.4.1.1 COF — Counter overflow flag bit

1 (set) – Counter overflow has occurred.

0 (clear) – Counter overflow has not occurred.

A.4.1.2 IL[2:0] — Interrupt level bits

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

A

A.4.1.3 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.

A.4.1.4 DRVA, DRVB — Drive time base bus bits

DRVA	DRVB	Bus selected
0	0	Neither time base bus A nor time base bus B is driven
0	1	Time base bus B is driven
1	0	Time base bus A is driven
1	1	Both time base bus A and time base bus B are driven

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

A.4.1.5 IN2 — Clock input pin status bit

This read-only status bit reflects the logic state of the MCSM clock input pin CTMC.

A.4.1.6 IN1 — Modulus load input pin status bit

This read-only status bit reflects the logic state of the MCSM modulus load input pin CTML.

A.4.1.7 EDGEN, EDGEP — Modulus load edge sensitivity bits

EDGEN	EDGEP	IN1 edge detector sensitivity
0	0	None
0	1	Positive edge only
1	0	Negative edge only
1	1	Positive and negative edge

A.4.1.8 CLK[2:0] — Counter clock select bits

CLK2	CLK1	CLK0	Free running counter clock source
0	0	0	Prescaler output 1 (÷ 2 or ÷ 3)
0	0	1	Prescaler output 2 (÷ 4 or ÷ 6)
0	1	0	Prescaler output 3 (÷ 8 or ÷ 12)
0	1	1	Prescaler output 4 (÷ 16 or ÷ 24)
1	0	0	Prescaler output 5 (÷ 32 or ÷ 48)
1	0	1	Prescaler output 6 (÷ 64 to ÷ 768)
1	1	0	CTMC pin input, negative edge
1	1	1	CTMC pin input, positive edge

A.4.2 MCSMCNT — MCSM counter register

(1) Offset from the base address of the MCSM submodule.

A.4.3 MCSMML — MCSM modulus latch register

(1) Offset from the base address of the MCSM submodule.

A

A.5 SASM registers and bits

A.5.1 SICA — SASM status/interrupt/control register A

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SICA $00(1)		FLAG	IL2	IL1	IL0	IARB3	IEN		BSL	IN		FORCE	EDOUT			MODE1	MODE0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the SASM submodule.

A.5.1.1 FLAG — Event flag bit

1 (set) – An input capture or output compare event has occurred.

0 (clear) – An input capture or output compare event has not occurred.

A.5.1.2 IL[2:0] — Interrupt level bits

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

A.5.1.3 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.

A.5.1.4 IEN — Interrupt enable bit

1 (set) – Interrupts enabled.

0 (clear) – Interrupts disabled.

A.5.1.5 BSL — Time base bus select bit

1 (set) – Time base bus B selected.

0 (clear) – Time base bus A selected.

A.5.1.6 IN — Input pin status bit

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.

A.5.1.7 FORCE — Force compare control bit

1 (set) – Force output flip-flop to behave as if an output compare has just occurred.

0 (clear) – No action.

A.5.1.8 EDOUT — Edge detect and output level bit

In IC mode:

1 (set) – Input capture on rising edge.

0 (clear) – Input capture on falling edge.

In OC and OP modes:

1 (set) – Output a logic one on the next output compare on channel A.

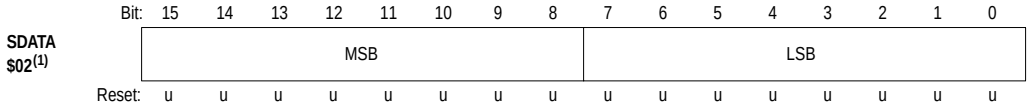
0 (clear) – Output a logic zero on the next output compare on channel A.

A

A.5.1.9 MODE1, MODE0 — SASM operating mode select bits

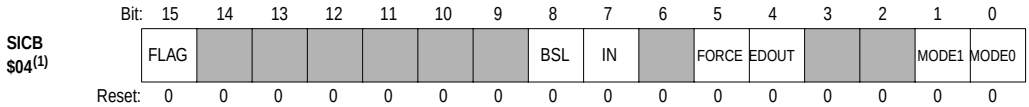
MODE1	MODE0	SASM channel operating mode
0	0	Input capture (IC)
0	1	Output port (OP)
1	0	Output compare (OC)
1	1	Output compare and toggle (OCT)

A.5.2 SDATA — SASM data register A



(1) Offset from the base address of the SASM submodule.

A.5.3 SICB — SASM status/interrupt/control register B



(1) Offset from the base address of the SASM submodule.

A.5.3.1 FLAG — Event flag bit

1 (set) – An input capture or output compare event has occurred on channel B.

0 (clear) – An input capture or output compare event has not occurred on channel B.

A.5.3.2 BSL — Time base bus select bit

1 (set) – Time base bus B selected.

0 (clear) – Time base bus A selected.

A.5.3.3 IN — Input pin status bit

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.

A.5.3.4 FORCE — Force compare control bit

1 (set) – Force output flip-flop to behave as if an output compare has just occurred on channel B.

0 (clear) – No action.

A.5.3.5 EDOUT — Edge detect and output level bit

In IC mode:

1 (set) – Input capture on rising edge.

0 (clear) – Input capture on falling edge.

In OC and OP modes:

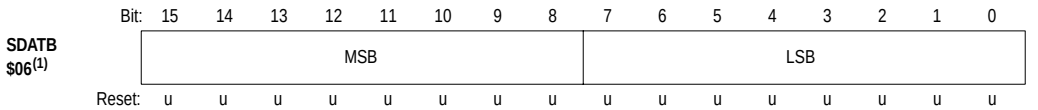
1 (set) – Output a logic one on the next output compare on channel B.

0 (clear) – Output a logic zero on the next output compare on channel B.

A.5.3.6 MODE1, MODE0 — SASM operating mode select bits

MODE1	MODE0	SASM channel operating mode
0	0	Input capture (IC)
0	1	Output port (OP)
1	0	Output compare (OC)
1	1	Output compare and toggle (OCT)

A.5.4 SDATB — SASM data register B



(1) Offset from the base address of the SASM submodule.

A.6 DASM registers and bits

A.6.1 DASMSIC — DASM status/interrupt/control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DASMSIC $00(1)		FLAG	IL2	IL1	IL0	IARB3		WOR	BSL	IN	FORCA	FORCB	EDPOL	MODE3	MODE2	MODE1	MODE0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the DASM submodule.

A.6.1.1 FLAG — Flag status bit

1 (set) – An input capture or output compare event has occurred.

0 (clear) – An input capture or output compare event has not occurred.

A.6.1.2 IL[2:0] — Interrupt level bits

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

A.6.1.3 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.

A.6.1.4 WOR — Wired-OR bit

1 (set) – Output buffer is open-drain.

0 (clear) – Output buffer is totem pole.

A.6.1.5 BSL — Bus select bit

1 (set) – The DASM is connected to time base bus B.

0 (clear) – The DASM is connected to time base bus A.

A.6.1.6 IN — Input pin status bit

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.

A.6.1.7 FORCA — Force A bit

1 (set) – Force output flip-flop to behave as if an output compare has just occurred on channel A.

0 (clear) – No action.

A.6.1.8 FORCB — Force B bit

1 (set) – Force output flip-flop to behave as if an output compare has just occurred on channel B.

0 (clear) – No action.

A.6.1.9 EDPOL — Edge polarity bit

In the IPWM mode:

1 (set) – Channel A captures on a falling edge.
Channel B captures on a rising edge.

0 (clear) – Channel A captures on a rising edge.
Channel B captures on a falling edge.

In the IPM and IC modes:

1 (set) – Channel A captures on a falling edge.

0 (clear) – Channel A captures on a rising edge.

In the OCB, OCAB and OPWM modes:

1 (set) – 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.

0 (clear) – 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.

A.6.1.10 MODE[3:0] — Mode select bits

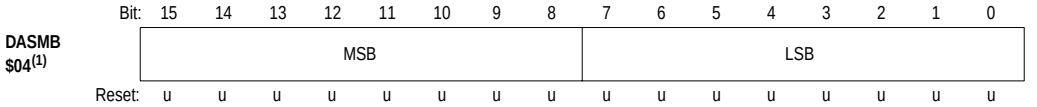
DASM control register bits				Bits of resolution	Time base bits ignored	DASM mode of operation
MOD3	MOD2	MOD1	MOD0			
0	0	0	0	—	—	DIS – Disabled
0	0	0	1	16	—	IPWM – Input pulse width measurement
0	0	1	0	16	—	IPM – Input period measurement
0	0	1	1	16	—	IC – Input capture
0	1	0	0	16	—	OCB – Output compare, flag on B compare
0	1	0	1	16	—	OCAB – Output compare, flag on A and B compare
0	1	1	0	—	—	
0	1	1	1	—	—	
1	0	0	0	16	—	OPWM – Output pulse width modulation
1	0	0	1	15	15	OPWM – Output pulse width modulation
1	0	1	0	14	15, 14	OPWM – Output pulse width modulation
1	0	1	1	13	15-13	OPWM – Output pulse width modulation
1	1	0	0	12	15-12	OPWM – Output pulse width modulation
1	1	0	1	11	15-11	OPWM – Output pulse width modulation
1	1	1	0	9	15-9	OPWM – Output pulse width modulation
1	1	1	1	7	15-7	OPWM – Output pulse width modulation

A.6.2 DASMA — DASM data register A

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DASMA $02(1)		MSB								LSB							
	Reset:	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

(1) Offset from the base address of the DASM submodule.

A.6.3 DASMB — DASM data register B



(1) Offset from the base address of the DASM submodule.

A

A.7 PWMSM registers and bits

A.7.1 PWMSIC — PWMSM status, interrupt and control register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMSIC $00(1)		FLAG	IL2	IL1	IL0	IARB3				PIN		LOAD	POL	EN	CLK2	CLK1	CLK0
	Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) Offset from the base address of the PWMSM submodule.

A.7.1.1 FLAG — Period completion status bit

1 (set) – PWM period completed.

0 (clear) – PWM period not completed.

A.7.1.2 IL[2:0] — Interrupt level bits

IL2	IL1	IL0	Selected level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

A.7.1.3 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.

A.7.1.4 PIN — Output pin status bit

1 (set) – Logic one state present on the PWMSM output pin.

0 (clear) – Logic zero state present on the PWMSM output pin.

A.7.1.5 LOAD — Period and pulse width register load control bit

1 (set) – Load PWMSM period and pulse width registers.

0 (clear) – No action.

A.7.1.6 POL — Output pin polarity control bit

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

A.7.1.7 EN — PWMSM enable control bit

1 (set) – Enable the PWMSM and start generation of PWM output pulses.

0 (clear) – Disable the PWMSM and stop generation of PWM output pulses.

A.7.1.8 CLK[2:0] — Clock rate selection bits

PWMSM CLK bits			CPSM bit DIV23	PWMSM clock	Clock source
CLK2	CLK1	CLK0			
0	0	0	0	$f_{SYS} \div 2$	PCLK1
0	0	1	0	$f_{SYS} \div 4$	Prescaler (÷2)
0	1	0	0	$f_{SYS} \div 8$	Prescaler (÷4)
0	1	1	0	$f_{SYS} \div 16$	Prescaler (÷8)
1	0	0	0	$f_{SYS} \div 32$	Prescaler (÷16)
1	0	1	0	$f_{SYS} \div 64$	Prescaler (÷32)
1	1	0	0	$f_{SYS} \div 128$	Prescaler (÷64)
1	1	1	0	$f_{SYS} \div 512$	Prescaler (÷256)
0	0	0	1	$f_{SYS} \div 3$	PCLK1
0	0	1	1	$f_{SYS} \div 6$	Prescaler (÷2)
0	1	0	1	$f_{SYS} \div 12$	Prescaler (÷4)
0	1	1	1	$f_{SYS} \div 24$	Prescaler (÷8)
1	0	0	1	$f_{SYS} \div 48$	Prescaler (÷16)
1	0	1	1	$f_{SYS} \div 96$	Prescaler (÷32)
1	1	0	1	$f_{SYS} \div 192$	Prescaler (÷64)
1	1	1	1	$f_{SYS} \div 768$	Prescaler (÷256)

A.7.2 PWMA — PWM period register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMA $02(1)		MSB								LSB							
	Reset:	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

(1) Offset from the base address of the PWMSM submodule.

A.7.3 PWMB — PWM pulse width register

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMB $04(1)		MSB								LSB							
	Reset:	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

(1) Offset from the base address of the PWMSM submodule.

A.7.4 PWMC — PWM counter register

(1) Offset from the base address of the PWMSM submodule.

A

B
CTM EXAMPLE – CTM2

CTM2 is an example of a typical CTM module implementation. A block diagram of CTM2 is shown in Figure B-1; it comprises the following submodules:

- a bus interface unit submodule (BIUSM),
- a clock prescaler submodule (CPSM),
- a free-running counter submodule (FCSM),
- two modulus counter submodules (MCSM) and
- 10 double action submodules (DASM).

Note: CTM2 does not contain any single action submodules (SASM) nor any pulse width modulation submodules (PWMSM).

Figure B-1 shows how the time base buses interconnect the counters and the channels in CTM2. Time base buses 2 and 3 are connected together to form one global time base bus (TBBB) that is accessible to every submodule. Time base buses 1 and 4 are not connected together but are known collectively as time base bus A (TBBA). Each submodule has access to two time base buses, TBBB and either TBB1 or TBB4 (see Table B-1).

Table B-2 shows how the CTM2 interrupt priority and vectors are structured.

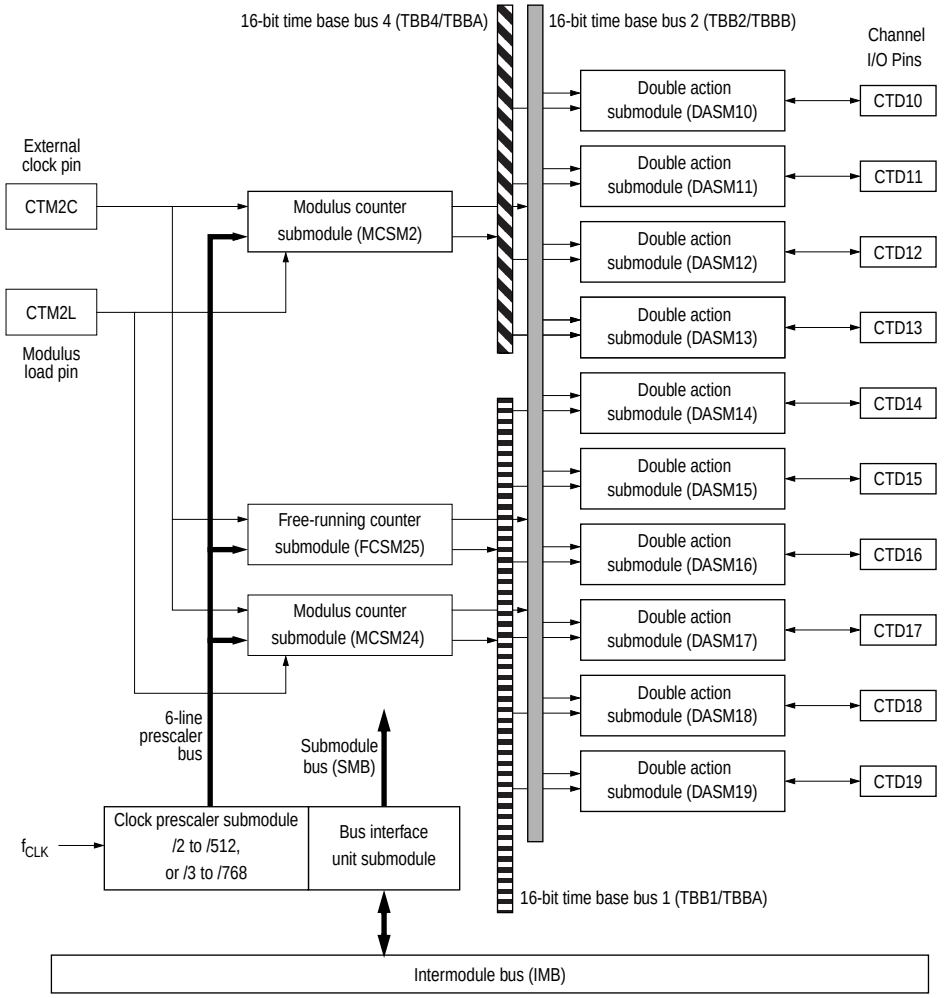


Figure B-1 Configurable timer module 2 (CTM2)

B

Table B-1 Time base bus allocation

Submodule	Local/global time base bus allocation	
	Global bus A (TBBA)	Global bus B (TBBB)
MCSM 2	TBB4	TBB2
DASM 10, 11, 12, 13	TBB4	TBB2
DASM 14, 15, 16, 17, 18, 19	TBB1	TBB2
MCSM 24	TBB1	TBB2
FCSM 25	TBB1	TBB2

Table B-2 CTM2 interrupt priority, vector allocation and pin allocation

Submodule name	Submodule binary base address (1)	Submodule binary vector number (2)	Submodule interrupt arbitration sequence number (3)	Corresponding pin name
BIUSM	z00000000	none	none	none
CPSM	z00001000	none	none	none
MCSM2	z00010000	xx000010	2	CTM2C, CTM2L
DASM 10	z01010000	xx001010	10	CTD10
DASM 11	z01011000	xx001011	11	CTD11
DASM 12	z01100000	xx001100	12	CTD12
DASM 13	z01101000	xx001101	13	CTD13
DASM 14	z01110000	xx001110	14	CTD14
DASM 15	z01111000	xx001111	15	CTD15
DASM 16	z10000000	xx010000	16	CTD16
DASM 17	z10001000	xx010001	17	CTD17
DASM 18	z10010000	xx010010	18	CTD18
DASM 19	z10011000	xx010011	19	CTD19
MCSM 24	z11000000	xx011000	24	CTM2C, CTM2L
FCSM 25	z11001000	xx011001	25	CTM2C

(1) z represents the high address bits defining the CTM2 base address.

(2) xx are the two interrupt vector base number bits VECT7 and VECT6 contained in the BIUSM.

(3) Interrupt arbitration #2 is highest priority, arbitration #25 is lowest priority.

Table B-2 also shows how the I/O pins are allocated in CTM2.

Note: In CTM2 all three counter submodules share one external clock input pin, CTM2C.

B.1 CTM2 registers

Table B-3 CTM2 register map

Address	15 8	7 0
$YFF400 – $YFF407	BIUSM registers	
$YFF408 – $YFF40F	CPSM registers	
$YFF410 – $YFF417	MCSM 2 registers	
$YFF418 – $YFF44F		
$YFF450 – $YFF457	DASM 10 registers	
$YFF458 – $YFF45F	DASM 11 registers	
$YFF460 – $YFF467	DASM 12 registers	
$YFF468 – $YFF46F	DASM 13 registers	
$YFF470 – $YFF477	DASM 14 registers	
$YFF478 – $YFF47F	DASM 15 registers	
$YFF480 – $YFF487	DASM 16 registers	
$YFF488 – $YFF48F	DASM 17 registers	
$YFF490 – $YFF497	DASM 18 registers	
$YFF498 – $YFF49F	DASM 19 registers	
$YFF4A0 – $YFF4BF		
$YFF4C0 – $YFF4C7	MCSM 24 registers	
$YFF4C8 – $YFF4CF	FCSM 25 registers	
$YFF4D0 – $YFF4FF		

Reserved registers always return zeros when read.

B

B.1.1 CTM2 bus interface unit submodule registers

Table B-4 BIUSM register map

Address	15 8	7 0
$YFF400	Module configuration register (MCR)	
$YFF402	BIUSM test register (TCR)	
$YFF404	Time base register (TBR)	
$YFF406		

B.1.2 CTM2 counter prescaler submodule registers

Table B-5 CPSM register map

Address	15 8	7 0
$YFF408	CPSM control register (CPCR)	
$YFF40A	CPSM test register (CPTR)	
$YFF40C		
$YFF40E		

B.1.3 CTM2 free-running counter submodule registers

Table B-6 FCSM register map

Address	15 8	7 0
$YFF4C8	FCSM status/interrupt/control register (FCSMSIC)	
$YFF4CA	FCSM counter (FCSMCNT)	
$YFF4CC		
$YFF4CE		

B.1.4 CTM2 modulus counter submodule registers

Table B-7 MCSM register map

Address (1)	15 8	7 0
$00	MCSM status/interrupt/control register (MCSMSIC)	
$02	MCSM counter (MCSMCNT)	
$04	MCSM modulus latch (MCSMML)	
$06		

(1) Offset from the base address of the MCSM.

B.1.5 CTM2 double action submodule registers

Table B-8 DASM register map

Address (1)	15 8	7 0
$00	DASM status/interrupt/control register (DASMSIC)	
$02	DASM register A (DASMA)	
$04	DASM register B (DASMB)	
$06		

(1) Offset from the base address of the DASM.

B

C
GLOSSARY

This section contains abbreviations and specialist words used in this data sheet and throughout the industry.

$xxxx The digits following the '$' are in hexadecimal format.

%xxxx The digits following the '%' are in binary format.

A/D, **ADC** Analog-to-digital (converter).

Assert An asserted signal is driven to its active or true state, irrespective of that state being represented by a high or low voltage level.

BIU Bus interface unit; in a module, the interface between the IMB and the internal circuitry of the module.

BIUSM Bus interface unit submodule; in a module, the submodule that allows the module to be connected to the IMB. All information transfer between the IMB and the SMB is handled by the BIUSM.

Bootstrap mode In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.

Bus cycle A single transfer of data across the bus using the established protocols and timing. A bus cycle consists of a single byte or word transfer.

Byte Eight bits.

CCR Condition codes register; an integral part of the CPU.

CERQUAD A ceramic package type, principally used for EPROM and high temperature devices.

Clear '0' — the logic zero state; the opposite of 'set'.

Clock Two tics; a full clock cycle. For a 16.67MHz clock, a clock cycle has a duration of 60ns.

CMOS Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.

Coherency The caability of a system to handle data as if its data bus were the same width as the data block.

COP Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.

CPSM	Counter prescaler submodule.
CPU	Central processing unit.
CPU16	Motorola's 16-bit CPU core for IMB based devices.
CPU32	Motorola's 32-bit CPU core for IMB based devices.
CTM	Configurable timer module.
D/A, DAC	Digital-to-analog (converter).
Daisy chain	In the CTM, a hardware priority system whereby the relative priorities of a number of interrupt sources is determined by their position in the chain.
DASM	Double action submodule.
EBI	External bus interface; a module responsible for interfacing the IMB with the external bus. The external bus includes those generalized lines needed for system operation and external transactions, but does not include the lines specific to a module. EBIs will differ from chip to chip depending upon the need for handling interrupts, external bus mastership, bus errors, etc.
EEPROM	Electrically erasable programmable read only memory. *aka* 'EEROM'.
EPROM	Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.
ESD	Electrostatic discharge.
Expanded mode	In this mode the internal address and data bus lines are connected to external pins. This enables the device to be used in much more complex systems, where there is a need for external memory for example.
EVS	Evaluation system. One of the range of platforms provided by Motorola for evaluation and emulation of their devices.
FCSM	Free-running counter submodule.
HCMOS	High-density complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
I/O	Input/output; used to describe a bidirectional pin or function.
IMB	Inter module bus; the 68300 series standard internal bus. It allows exchange of data between master and slave modules.
IC	Input capture; a function provided by the timing system, whereby an external event is 'captured' by storing the value of a counter at the instant the event is detected.
Interrupt	An asynchronous external event handled by the MCU. The external event is detected by the MCU and causes a predetermined action to occur.
$\overline{\text{IRQ}}$	Interrupt request. (The overline indicates that this is an active-low signal format.)
kbyte	A kilo-byte (of memory); 1024 bytes.
LCD	Liquid crystal display.

LSB	Least significant byte.
Master	The module that initiates a bus request and controls the bus transaction with a slave module.
MCSM	Modulus counter submodule.
MCU	Microcontroller unit.
Module	A functional block compatible with the Motorola modular microcontroller family (MMF) that connects to the IMB.
MSB	Most significant byte.
Negate	A negated signal is driven to its inactive or false state, irrespective of that state being represented by a high or low voltage level.
Nibble	Half a byte; four bits.
NRZ	Non-return to zero.
Opcode	The opcode is a byte which identifies the particular instruction and operating mode to the CPU. See also: prebyte, operand.
Operand	The operand is a byte containing information the CPU needs to execute a particular instruction. There may be from 0 to 3 operands associated with an opcode. See also: opcode, prebyte.
OC	Output compare; a function provided by the timing system, whereby an external event is generated when an internal counter value matches a predefined value.
PLCC	Plastic leaded chip carrier package.
PLL	Phase-locked loop circuit. This provides a method of frequency multiplication, to enable the use of a low frequency crystal in a high frequency circuit.
Prebyte	This byte is sometimes required to qualify an opcode, in order to fully specify a particular instruction. See also: opcode, operand.
Pull-down, pull-up	These terms refer to resistors, sometimes internal to the device, which are permanently connected to either ground or V_{DD}.
PWM	Pulse width modulation; a technique where the width of the high and low periods of a waveform is varied, usually to enable a representation of an analog value.
QFP	Quad flat pack package.
RAM	Random access memory. Fast read and write, but contents are lost when the power is removed.
RFI	Radio frequency interference.
RTI	Real-time interrupt.
ROM	Read-only memory. This type of memory is programmed during device manufacture and cannot subsequently be altered.
RS-232C	A standard serial communications protocol.

SAR	Successive approximation register.
SASM	Single action submodule.
SCI	Serial communications interface.
Set	'1' — the logic one state; the opposite of 'clear'.
Silicon Glen	An area in the central belt of Scotland, so called because of the concentration of semiconductor manufacturers and users found there.
SIM	System integration module; a module on an IMB device that reduces the need for external glue logic by providing an external interface and a variety of other system related functions.
Single chip mode	In this mode the device functions as a self contained unit, requiring only I/O devices to complete a system.
Slave	The module that responds to a master's request.
SMB	Submodule bus; the internal CTM bus that allows exchange of information between submodules.
SPI	Serial peripheral interface.
State	Subunit of the bus cycle in which specific events occur. Each state has a duration of one clock tic, and represents a single state within the controlling microengine. For the IMB, the states defined include B1, B2, B2*, B3, B4 and B4*.
Submodule	A functional block that defines a CTM function. Most of the submodules can be placed several times in order to obtain the desired system.
Test mode	This mode is intended for factory testing.
Tic	A clock tic is defined as the single high or low state, irrespective of that state being represented by a high or low voltage level. For a 16.67MHz, a clock tic has a duration of 30ns.
Transaction	The transfer of data between two modules (a master and a slave) using the IMB and its established protocol. A single transaction may require multiple bus cycles.
TTL	Transistor-transistor logic.
UART	Universal asynchronous receiver transmitter.
VCO	Voltage controlled oscillator.
Watchdog	*See* 'COP'.
Wired-OR	A means of connecting outputs together such that the resulting composite output state is the logical OR of the state of the individual outputs.
Word	Two bytes; 16 bits.
$\overline{\text{XIRQ}}$	Non-maskable interrupt request. The overline indicates that this has an active-low signal format.

C

D
INDEX

In this index numeric entries are placed first; page references in *italics* indicate that the reference is to a figure.

D

E

F

G

I

L

M

D

O

P

D

R

S

T

V

W

Z

D

THIS PAGE INTENTIONALLY LEFT BLANK

CUSTOMER FEEDBACK QUESTIONNAIRE (CTMRM)

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this card (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organization	☐	☐	☐	☐	Tables	☐	☐	☐	☐
Readability	☐	☐	☐	☐	Table of contents	☐	☐	☐	☐
Understandability	☐	☐	☐	☐	Index	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐	Page size/binding	☐	☐	☐	☐
Illustrations	☐	☐	☐	☐	Overall impression	☐	☐	☐	☐

Comments: _____

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application ☐
System design ☐
Training purposes ☐
Other ☐ Please specify: _____

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely ☐ ☐ ☐ ☐ Not at all

Comments: _____

4. How easy is it to find the information you are looking for?

Easy ☐ ☐ ☐ ☐ Difficult

Comments: _____

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

		Too little detail				Too much detail
Table of Contents/List of Figures/List of Tables		☐	☐	☐	☐	☐
SECTION 1	FUNCTIONAL OVERVIEW	☐	☐	☐	☐	☐
SECTION 2	BUS INTERFACE UNIT SUBMODULE (BIUSM)	☐	☐	☐	☐	☐
SECTION 3	INTERRUPTS	☐	☐	☐	☐	☐
SECTION 4	COUNTER PRESCALER SUBMODULE (CPSM)	☐	☐	☐	☐	☐
SECTION 5	FREE-RUNNING COUNTER SUBMODULE (FCSM)	☐	☐	☐	☐	☐
SECTION 6	MODULUS COUNTER SUBMODULE (MCSM)	☐	☐	☐	☐	☐
SECTION 7	SINGLE ACTION SUBMODULE (SASM)	☐	☐	☐	☐	☐
SECTION 8	DOUBLE ACTION SUBMODULE (DASM)	☐	☐	☐	☐	☐
SECTION 9	PULSE WIDTH MODULATION SUBMODULE (PWMSM)	☐	☐	☐	☐	☐
SECTION 10	ELECTRICAL SPECIFICATIONS	☐	☐	☐	☐	☐
APPENDIX A	REGISTER SUMMARY	☐	☐	☐	☐	☐
APPENDIX B	CTM EXAMPLE – CTM2	☐	☐	☐	☐	☐
APPENDIX C	GLOSSARY	☐	☐	☐	☐	☐
APPENDIX D	INDEX	☐	☐	☐	☐	☐

Comments: _____

6. Have you found any errors? If so, please comment: _____

7. From your point of view, is anything missing from the document? If so, please say what: _____

– Cut along this line to remove –

8. How could we improve this document?

9. How would you rate Motorola's documentation?

	Excellent			Poor
– In general	☐	☐	☐	☐
– Against other semiconductor suppliers	☐	☐	☐	☐

10. Which semiconductor manufacturer provides the best technical documentation?

11. Which company (in any field) provides the best technical documentation?

12. How many years have you worked with microprocessors?

Less than 1 year ☐ 1–3 years ☐ 3–5 years ☐ More than 5 years ☐

– Second fold back along this line –

By air mail
Par avion

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.,
Colvilles Road,
Kelvin Industrial Estate,
EAST KILBRIDE,
G75 8BR.
GREAT BRITAIN.

F.A.O. Technical Publications Manager
(re: CTMRM/D)

MOTOROLA LTD.
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

13. Currently there is some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment.

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: Phone No:

Position: FAX No:

Department:

Company:

Address:

Thank you for helping us improve our documentation,
Graham Livey, Technical Publications Manager, Motorola Ltd., Scotland.

– Last tuck this edge into opposite flap –

Literature Distribution Centres:

EUROPE: Motorola Ltd., European Literature Centre, 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.

ASIA PACIFIC: Motorola Semiconductors (H.K.) Ltd., Silicon Harbour Center, No. 2, Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

JAPAN: Nippon Motorola Ltd., 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.

USA: Motorola Literature Distribution, P.O. Box 20912, Phoenix, Arizona 85036.

MOTOROLA