Modular Microcontroller Family

# GPT

# GENERAL PURPOSE TIMER

REFERENCE
MANUAL

MOTOROLA

# GPT
# GENERAL PURPOSE TIMER REFERENCE MANUAL

# PREFACE

The General-Purpose Timer (GPT) is an integral module of Motorola's family of modular microcontrollers. The *GPT Reference Manual* describes the capabilities, operation, and functions of the GPT.

This reference manual is organized as follows:

# TABLE OF CONTENTS

# TABLE OF CONTENTS (Continued)

# TABLE OF CONTENTS (Concluded)

# LIST OF ILLUSTRATIONS

# LIST OF ILLUSTRATIONS (Concluded)

# LIST OF TABLES

# SECTION 1
# FUNCTIONAL OVERVIEW

The General-Purpose Timer (GPT), a module in Motorola's family of modular microcontrollers, is a simple yet flexible 11-channel timer for use in systems where a moderate level of CPU control is required. The GPT can be broken into several nearly independent submodules: the compare/capture unit, the pulse accumulator, and the pulse-width modulation unit. Figure 1-1 is a block diagram of the GPT.

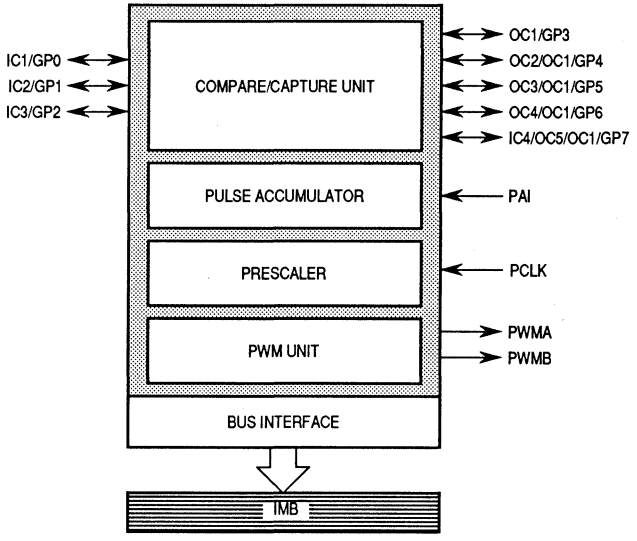



**Figure 1-1. GPT Block Diagram**

The compare/capture unit features three input capture channels, four output compare channels, and one channel that can be selected as an input capture or output compare channel. These channels share a 16-bit free-running counter (TCNT) which derives its clock from a nine-stage prescaler or from the external clock input pin, PCLK.

The pulse accumulator channel logic includes its own 8-bit counter and can operate in either event counting mode or gated time accumulation mode.

The pulse-width modulation submodule has two outputs that are periodic waveforms whose duty cycles may be independently selected and modified by user software. The PWM unit has its own 16-bit free-running counter which is clocked by an output of the nine-stage prescaler (the same prescaler used by the compare/capture unit) or by the clock input pin, PCLK.

If not needed for timing functions, any of the GPT pins can be used for general-purpose input/output (I/O). The input capture and output compare pins are bi-directional and can be used to form an 8-bit parallel port. The PWM pins are outputs only. The pulse accumulator input (PAI) and PCLK pins are inputs only.

The GPT bus interface provides the connection to the intermodule bus (IMB). This bus provides a standard interface between different modules and the CPU. Important features of the bus include multiple bus masters, exception processing support, address space partitioning, multiple interrupt levels, vectored interrupts, and extendable (wait states) bus cycles. New modules designed to conform to the IMB protocol can quickly be combined with other processors, peripherals, and memories to meet almost any controller application.

Table 1-1 shows the registers and counters in the GPT. The addresses shown are on word boundries; however, all registers and counters can be accessed using byte or word operations. Counters TCNT and PWMCNT, and registers TICx, TOCx, and TI4O5 must be accessed by word operations to ensure coherency. Coherency is the reading or writing of data identical in age. Using byte accesses when reading a counter such as the TCNT, there is a possibility that data in the byte not being accessed will change while the other byte is read. To prevent this, both bytes must be accessed at the same time.

Two control registers, the module configuration register (MCR) and the interrupt control register (ICR) can only be accessed while the processor is in supervisor mode. Refer to **SECTION 7 INTERRUPTS** and **SECTION 9 SPECIAL MODES** for information on these registers.

## Table 1-1. GPT Register Map

| | ADDRESS | WORD 15 — BYTE n — 8 \| 7 — BYTE n + 1 — 0 |
|---|---|---|
| S | $YFF900 | MCR |
| S | $YFF902 | RESERVED |
| S | $YFF904 | ICR |
| U | $YFF906 | PDDR \| PDR |
| U | $YFF908 | OC1M \| OC1D |
| U | $YFF90A | TIMER COUNTER (TCNT) |
| U | $YFF90C | PACTL \| PACNT |
| U | $YFF90E | TIC1 |
| U | $YFF910 | TIC2 |
| U | $YFF912 | TIC3 |
| U | $YFF914 | TOC1 |
| U | $YFF916 | TOC2 |
| U | $YFF918 | TOC3 |
| U | $YFF91A | TOC4 |
| U | $YFF91C | TI4O5 |
| U | $YFF91E | TCTL1 \| TCTL2 |
| U | $YFF920 | TMSK1 \| TMSK2 |
| U | $YFF922 | TFLG1 \| TFLG2 |
| U | $YFF924 | CFORC/PWMC |
| U | $YFF926 | PWMA REGISTER \| PWMB REGISTER |
| U | $YFF928 | PWM COUNTER (PWMCNT) |
| U | $YFF92A | PWMA BUFFER REGISTER \| PWMB BUFFER REGISTER |
| U | $YFF92C | PRESCALER (Lower 9 Bits) |
| | $YFF92E<br>$YFF93F | RESERVED |

S = Supervisor-accessible only

U = User or Supervisor depending on state of SUPV in the MCR

Y = m111, where m is the state of the modmap bit in the module configuration register of the system integration module (Y = $7 or $F).

## 1.1 Features

- Modular Architecture
- Input Capture/Output Compare Unit
  - — Three Input Capture Pins
  - — Four Output Compare Pins
  - — One Input Capture/Output Compare Pin
- One Pulse Accumulator/Event Counter Pin
- Two-Channel PWM Unit
  - — Programmable Clock Logic
  - — 8-Bit Resolution
  - — Independent Clock Source
- Dedicated Clock Input Pin
- Nine-Stage Prescaler
  - — Independent Prescaler Taps for Capture/Compare Unit and the PWM Unit

## 1.2 Input Capture (IC) Concepts

An input capture function has three basic parts: edge select logic, an input capture latch, and a 16-bit free-running counter. The edge select logic determines the type of input transition to which the circuit responds. When an input transition occurs, an input capture function latches the contents of the counter 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. Refer to Figure 1-2.

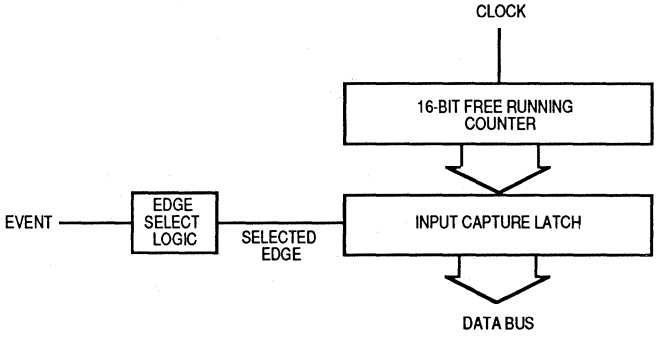



**Figure 1-2. Input Capture Simplified Block Diagram**

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, two alternate polarity edges are captured. For example, to measure the high time of a pulse, the input transition is captured at the rising edge and subtracted from the time captured for 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 track the overflows of the 16-bit counter to extend its range.

Another use for the input capture function is to establish a time reference. In this case, an input capture function is used in conjunction with an output compare function. For example, if the user wishes to activate an output signal a specific number of clock cycles after detecting an input event (edge), the input capture function is used to record the time at which the edge occurred. A number corresponding to the desired delay is added to this captured value and stored to

an output compare register. Because both input captures and output compares 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.

## 1.3 Output Compare (OC) Concepts

A 16-bit free-running counter provides the timing reference for output compares. Output compare functions are used to program a specific time an event occurs. An output compare function has a dedicated 16-bit compare register and a 16-bit comparator. When the contents of the compare register match the value of the free-running counter, the comparator sets an output compare flag. Refer to Figure 1-3.

Other events can occur when the flag is set. An interrupt can be generated if enabled. State changes can optionally occur on pins associated with the output compare function.

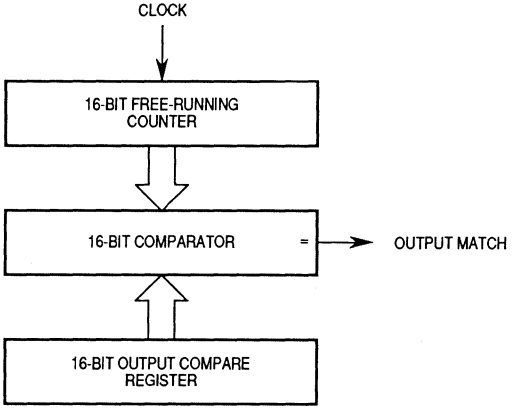



**Figure 1-3. Output Compare Simplified Block Diagram**

The output compare function can generate an output of a specific duration and polarity. A 16-bit value corresponding to the time a pin state change will occur is written to the output compare register. The output compare function is configured to automatically generate a high or low output on the pin or toggle its state when the match occurs. The output compare register is reprogrammed to a new value after the compare occurs. When the next match takes place, the pin returns to the previous state. The new value corresponds to the time the next compare occurs. 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.

## 1.4 Pulse Accumulator Input (PAI) Concepts

The pulse accumulator contains an 8-bit counter and edge select logic. The pulse accumulator has two modes of operation: event counting and gated mode. In event counting mode, an 8-bit counter is incremented when an event occurs. In gated mode an internal clock source increments the 8-bit counter while a selected level is present at the pulse accumulator. When the input is negated, the counter is stopped. Two flags are generated: one to indicate the occurrence of an event, and the other to indicate counter overflow. Either of these flags, when enabled, can cause the processor to be interrupted. Refer to Figure 1-4.

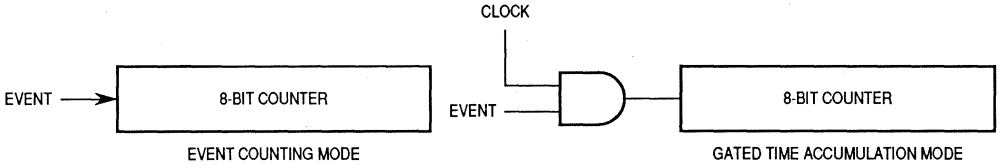



**Figure 1-4. Pulse Accumulator Simplified Block Diagram**

The pulse accumulator can be used 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 (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. Because only 255 events can be counted before the counter overflows, the overflow flag can be used to extend the counter range beyond 8 bits.

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.

## 1.5 Pulse-Width Modulation (PWM) Concepts

A pulse-width modulated waveform is created when the high to low time ratio of a periodic rectangular signal can be varied. If the waveform can be incrementally changed by 1/256 of its period, it has 8 bits of resolution. Refer to Figure 1-5.

As shown in the pulse-width modulation simplified block diagram (Figure 1-6), there are two comparators per PWM function: the zero detector and the 8-bit comparator. The PWM unit has a 16-bit counter. Each PWM function can use 8 bits; each can use either valid set of 8 bits. Every time the 8-bit counter overflows from $FF to $00, the zero detector sets the output latch. 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 8-bit register. When a match occurs the latch is reset. By changing the value in the 8-bit register, the duty cycle is continuously variable in n/256 increments.

When the 8-bit register contains $00, the output latch stays in the reset condition (pin low all the time). When the 8-bit register is loaded with $01, the output latch will stay high for one count time. When the register contains $80 (128 decimal), the latch remains high for 128 counts of the timer before it is reset. Writing to a special control bit is required to obtain a 100% duty cycle (output high all of the time).

By varying the input clock frequency to the PWM counter, the period of the PWM signal will also vary.

The PWM output can be used to electronically control the speed of a motor. The PWM waveform drives a switching amplifier which in turn controls the speed and direction of the motor. By adding a low-pass filter to a PWM output, the unit can be used as a D/A converter, the longer the high time of the output waveform the higher the average value of output voltage produced.

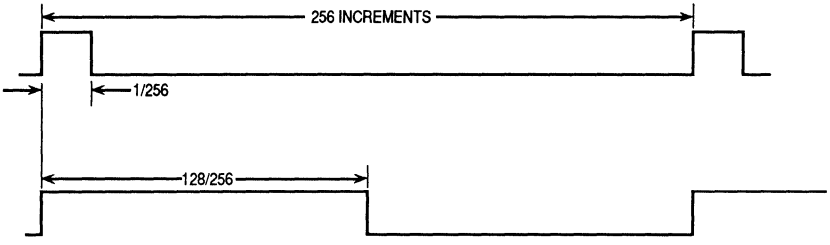



**Figure 1-5. Pulse-Width Modulation Example**

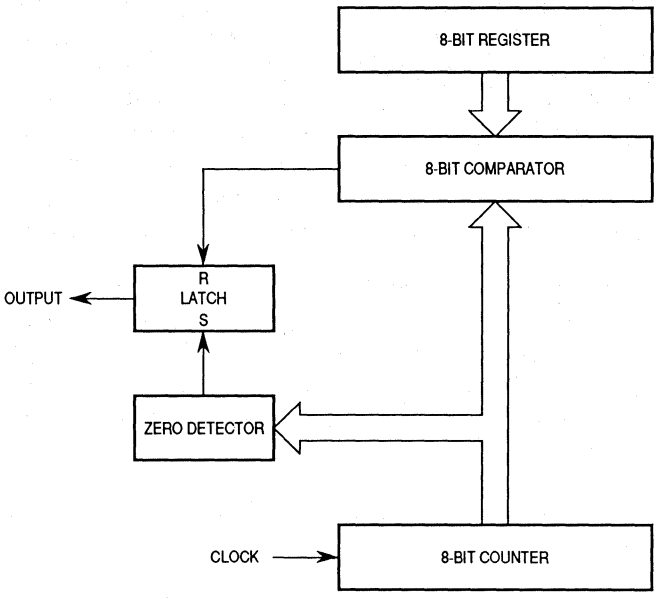



**Figure 1-6. Pulse-Width Modulation Simplified Block Diagram**

# SECTION 2
# SIGNAL DESCRIPTIONS

The GPT has 12 signal pins that provide connections to the internal functions of the module. This section contains brief descriptions of the GPT input and output signals in their functional groups.

## 2.1 Signal Groups

The block diagram in Figure 2-1 shows the primary and alternate functions of the signal pins. When the pins are not needed for their primary function they can be used for general-purpose input or output. The block diagram also shows which pins are bi-directional and which are either input or output only.

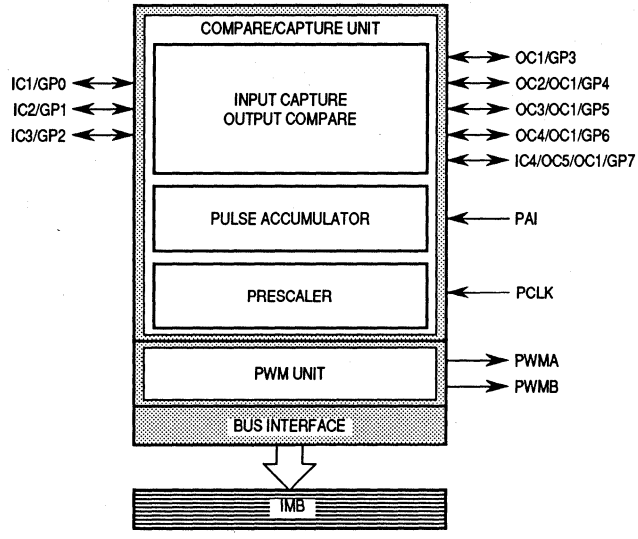



**Figure 2-1. Function Signal Groups**

## 2.2 Input Capture Pins (IC1–IC3)

These pins are used by the input capture functions of the GPT. Each pin is associated with a single input capture function. The pin inputs are conditioned in such a way that any pulse longer than two system clocks is guaranteed to pass, and any pulse shorter than one system clock is ignored. Each pin has a

dedicated 16-bit capture register to hold the captured counter value. When any of the pins are not needed for the input capture function they can be used for general-purpose I/O. Refer to **3.2 Input Capture Functions** for additional details of the input capture function.



### 2.3 Input Capture/Output Compare Pin (IC4/OC5)

This pin can be configured to be an input capture or an output compare function. It has one 16-bit register which is used for holding either the input capture value or the output match value. When used as an input, the signal is conditioned in such a way that any pulse longer than two system clocks is guaranteed to pass, and any pulse shorter than one system clock is ignored. If this pin is not needed for either the input capture or output compare function, it can be used for general-purpose I/O. Refer to **3.2 Input Capture Functions** and **3.3 Output Compare Functions** for additional details on the operation of these functions.

### 2.4 Output Compare Pins (OC1–OC4)

These pins are used for the output compare functions of the GPT and operate independently of each other. There is a dedicated 16-bit compare register and 16-bit comparator for each pin. Pins OC2, OC3, and OC4 are associated with a specific output compare function; whereas, the OC1 function can affect the output of any combination of output compare pins. Automatic preprogrammed pin actions occur on a successful match. The programmable pin actions differ between OC2–OC5 and OC1. If the OC1 pin is not needed for the output compare function, it can be used to output the clock selected for the timer counter register (TCNT). Any of the pins can be used for general-purpose I/O if not needed for the output compare function. For additional details on the operation of the output compare function refer to **3.3 Output Compare Functions**. Refer to **5.1 Prescaler** for details on this submodule.

### 2.5 Pulse Accumulator Input Pin (PAI)

The PAI pin is the signal input to the pulse accumulator. If not needed for this function, it can be used as a general-purpose input pin. The signal is conditioned in such a way that any pulse longer than two system clocks is guaranteed to pass, and any pulse shorter than one system clock is ignored. Any pulse shorter than one system clock is filtered out. For more details on the pulse accumulator function, refer to **4.1 Pulse Accumulator.**

### 2.6 Pulse-Width Modulation (PWMA, PWMB)

The PWMA and PWMB pins are used as outputs for the PWM functions. These outputs can be programmed to generate a periodic waveform with a variable frequency and duty cycle. The pins can be used for general-purpose output if not needed for the PWM function. PWMA can also be used to output the clock selected as the input to the PWM counter (PWMCNT). For more details on the

PWM functions refer to **SECTION 6 PULSE-WIDTH MODULATION (PWM) UNIT.**



## 2.7 Auxiliary Timer Clock Input (PCLK)

PCLK is an external clock input which is dedicated to the GPT. The signal is conditioned in such a way that any pulse longer than two system clocks is guaranteed to pass, and any pulse shorter than one system clock is ignored. PCLK can be used as the clock source for the capture/compare unit or the PWM unit in place of one of the prescaler outputs. If this pin is not used as a clock input, it can be used as a general-purpose input pin. See **5.1 Prescaler** and **8.1.1 Uni-Directional I/O** for additional information on PLCK.

2

# SECTION 3
# COMPARE/CAPTURE UNIT

3

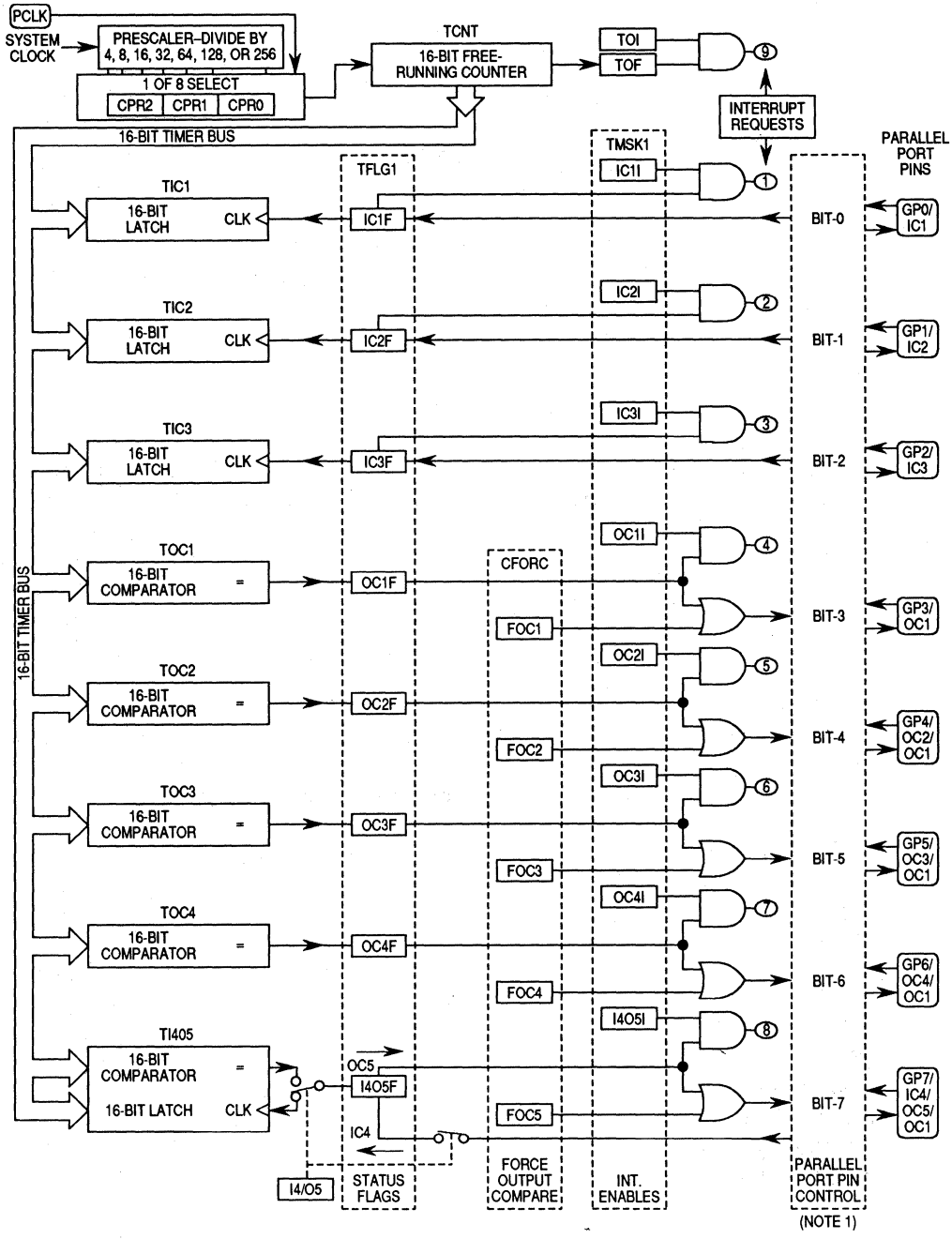
The compare/capture unit is one of the major submodules of the GPT. It contains the timer counter (TCNT), the input capture (IC) functions, and the output compare (OC) functions. Refer to Figure 3-1.

## 3.1 Timer Counter

The timer counter (TCNT) is the key timing component in the compare/capture unit. The timer counter is a 16-bit free-running counter that starts counting after the processor comes out of reset. The counter cannot be stopped during normal operation. Refer to **SECTION 9 SPECIAL MODES** on how to stop the counter. After reset, the GPT is configured to use the system clock divided by four as the input to the counter. The prescaler divides the system clock and provides selectable input frequencies. User software can configure the system to use one of seven outputs from the prescaler or an external clock through the PCLK input pin. Refer to **SECTION 5 PRESCALER** for more details on prescaler operation.

The counter appears as a register in the GPT and can be read any time with user software without affecting its value. Because the GPT is interfaced to the IMB and the IMB supports a 16-bit bus, a word read gives a coherent value. If coherency is not needed, byte accesses can be made. The counter is set to $0000 on reset and is a read-only register. There are two exceptions: test mode and freeze mode. Any value can be written to the timer counter while in these modes. Refer to **9.1 Test Mode** and **9.3 Freeze Mode** for information on test and freeze mode operation.

When the counter rolls over from $FFFF to $0000 the timer overflow bit is set in timer interrupt flag register 2 (TFLG2). An interrupt can be enabled by setting the corresponding interrupt enable bit (TOI) in the timer interrupt mask register 2 (TMSK2). Refer to **7.1.2.1 Timer Interrupt Flag Registers 1–2 (TFLG1/TFLG2)** and **7.1.3.1 Timer Interrupt Mask Registers 1–2 (TMSK1/TMSK2)** for more information on these registers; refer to **7.1 Interrupts** for information on interrupt operation.




NOTE: Parallel port pin actions are controlled by DDR, OC1M, OC1D, and TCTL1 registers.

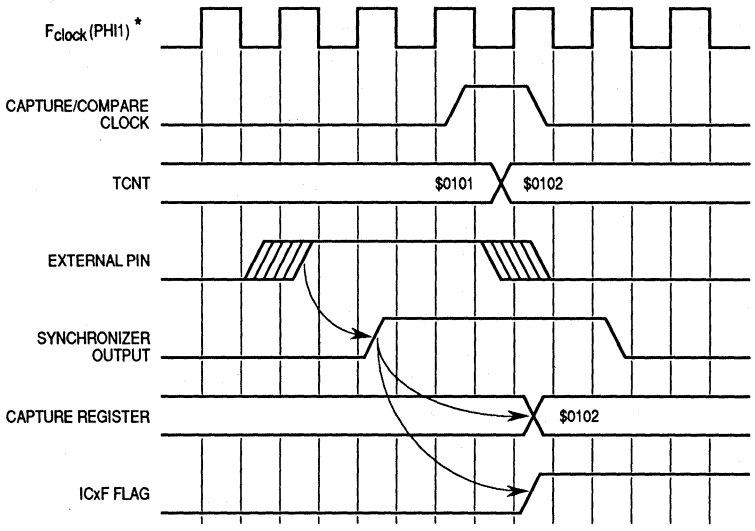
**Figure 3-1. Compare/Capture Unit Block Diagram**

## 3.2 Input Capture Functions

Each GPT input capture pin IC1, IC2, IC3, and also IC4/OC5 when used as an input capture function, has a dedicated 16-bit latch, input edge-detection/selection logic, and interrupt generation logic. All of the input capture functions use the same 16-bit timer counter (TCNT). The latch captures the contents of the TCNT when the selected event occurs at the corresponding input capture pin. Refer to **1.2 Input Capture (IC) Concepts** for additional information on the basic operation of an input capture function. The edge detection logic contains control bits which allow user software to select the edge polarity to be recognized. These are the EDGExA and EDGExB bits in timer control register 2 (TCTL2). Each of the input capture functions can be independently configured to detect rising edges only, falling edges only, any edge (rising or falling), or disable the input capture function. Refer to Figure 3-5 and Table 3-1 for the required bit patterns. The input capture functions operate independently of each other and can capture the same TCNT value if the input edges are all detected within the same timer count cycle.

The interrupt generation logic includes a status flag, which indicates that an edge is detected, and a local interrupt enable bit, which determines if the corresponding input capture function will generate a hardware interrupt request. The input capture sets the ICxF bit in the timer interrupt flag register 1(TFLG1) and can cause an interrupt if the corresponding ICxI bit is set in the timer interrupt mask register 1 (TMSK1). If the interrupt request is inhibited (the ICxI bit cleared), the input capture is operating in polled mode where software must read the status flag to recognize that an edge was detected. Refer to **7.1 Interrupts** for additional details on interrupt operation.

Because input capture events are generally asynchronous to the timer counter, they are conditioned by a synchronizer and digital filter. This synchronizes the input events to the system clock so that actual latching of the TCNT contents will occur on the opposite half cycle of the system clock from when the counter is being incremented. The input is conditioned in such a way that any event longer than two system clocks is guaranteed to be captured and any signal shorter than one system clock will be filtered out and have no effect. Notice the relationship of the system clock to the output of the synchronizer as shown in Figure 3-2. The value latched into the capture register by an input capture corresponds to the value of the counter several system clock cycles after the input transition which triggered the edge detection logic. There can be up to one clock cycle of uncertainty in latching of the input transition. The maximum time is determined by the system clock frequency.




* PHI1 is the same frequency as the system clock; however, it does not have the same timing.

**Figure 3-2. Input Capture Timing Example**

Because the input capture register is a 16-bit register, a word access is required to ensure coherency. If this is not required, each byte can be accessed independently using byte accesses. The input capture registers can be read at g their values.

An input capture occurs every time a selected edge is detected, even if the input capture flag is already set. This means that the value read from the input capture register corresponds to the most recent edge at the pin, which may not be the edge that caused the input capture flag to be set.

If any of the pins IC1–IC3 are not needed for an input capture function, they can be used as general-purpose input/output. The data direction of the pin is determined by the state of bits in the data direction register (DDR). For more information on general-purpose I/O refer to **8.1 General-Purpose I/O.**

### 3.2.1 Input Capture Registers (TIC1–3)

The input capture registers are 16-bit read-only registers which are used to latch the value of TCNT when a specified transition is detected on the corresponding input capture pin. Reads of these registers should be word accesses to ensure coherency. They are reset to $FFFF. Refer to Figure 3-3.

**TIC1–TIC3** — Input Capture Registers 1–3 **$YFF90E, $YFF910, $YFF912**

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

**Figure 3-3. Input Capture Register 1–3 (TIC1–3)**



### 3.2.2 Input Capture 4/Output Compare Register (TI4O5)

Input capture register 4/output compare 5 (TI4O5) serves as either an input capture register or as an output compare register depending on the function chosen for the I4/O5 pin. Refer to Figure 3-4. To make this pin serve as an input capture pin, the I4/O5 bit in the pulse accumulator control register (PACTL) must be set to a logic level one. Set the bit to a logic level zero for the pin to be used as output compare. Refer to **4.1.1 Pulse Accumulator Register/Pulse Accumulator Counter (PACTL/PACNT)**. TI4O5 is reset to $FFFF.

**TI4O5** — Input Capture 4/Output Compare 5 Register **$YFF91C**

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

**Figure 3-4. Input Capture 4/Output Compare 5 Register (TI4O5)**

### 3.2.3 Timer Control Register 2 (TCTL2)

TCTL2 is used to select the edge to which the input capture logic will respond. Table 3-1 shows that the input capture logic can be disabled, an input capture can occur on a rising edge, a falling edge, or any edge. Refer to Figure 3-5.

**TCTL2** — Timer Control Register 2 **$YFF91E**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * | * | EDGE4B | EDGE4A | EDGE3B | EDGE3A | EDGE2B | EDGE2A | EDGE1B | EDGE1A |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

3

* These bits are part of TCTL1.

NOTE: The address shown is on a word boundary; however, the register can be accessed by a word or byte address ($YFF91F) cycle.

**Figure 3-5.Timer Control Register 2 (TCTL2)**

EDGExA, EDGExB — Input Capture Edge Control Bits

These bits are encoded to configure the input edge sensing logic for the corresponding input capture.

**Table 3-1. Edge Bits**

| EDGExB–A | Configuration |
|---|---|
| 00 | Capture Disabled |
| 01 | Capture on Rising Edge Only |
| 10 | Capture on Falling Edge Only |
| 11 | Capture on Any (Rising or Falling) Edge |

## 3.3 Output Compare Functions

Each of the GPT output compare pins OC1, OC2, OC3, OC4, and IC4/OC5, when used as an output compare function, has a dedicated 16-bit compare register, a 16-bit comparator and interrupt generation logic. When the programmed contents of an output compare register matches TCNT, an output compare status flag (OCxF) bit in TFLG1 is set. Certain automatic actions are initiated for that output compare function. These automatic actions can be a hardware interrupt request and state changes at the related timer output pin. Refer to **7.1.2.1 Timer Interrupt Flag Registers 1–2 (TFLG1/TFLG2)** for details of the TFLG1 register.

An interrupt is generated on a successful output match, if the interrupt enable bit (OCxI) for this output compare function is set in TMSK1. Refer to **7.1 Interrupts** for details on interrupt operation.

The output compare logic is designed to prevent false compares during data transition times.

Control bits in the CFORC register allow for early forced compares. Refer to **3.3.4 Timer Compare Force Register (CFORC)**.

The operation of OC1 is slightly different from that of the other output compare functions. Refer to **3.3.3 Output Compare 1 (OC1)** for a description of this output function and its operation.



### 3.3.1 Timer Control Register 1 (TCTL1)

The automatic actions for OC2–OC4, and IC4/OC5 if configured as an output compare function, are controlled by pairs of bits, OMx and OLx, in timer control register 1 (TCTL1). Refer to Figure 3-6. The automatic pin actions for each output compare are independently selectable. Refer to Table 3-2 for the bit combinations of available output actions.

**$YFF91E**

**TCTL1** — Timer Control Register 1

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OM5 | OL5 | OM4 | OL4 | OM3 | OL3 | OM2 | OL2 | * | * | * | * | * | * | * | * |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |

* These bits are part of TCTL2.

NOTE: The address shown is on a word boundary; however, the register can be accessed by a word or byte cycle.

**Figure 3-6. Timer Control Register 1 (TCTL1)**

OMx — Output Compare Mode Bits

OLx — Output Compare Level Bits

These bits are encoded to specify the output action to be taken as a result of a successful OCx compare.

**Table 3-2. OMx, OLx Bits**

| OMx–OLx | Action Taken on Successful Compare |
|---|---|
| 00 | Timer Disconnected from Output Logic |
| 01 | Toggle OCx Output Line |
| 10 | Clear OCx Output LIne to 0 |
| 11 | Set OCx Output Line to 1 |



If an output compare function is disconnected from the corresponding pin (OMx:OLx = 0:0), then a bit in the parallel data direction register (PDDR) determines whether the pin is configured as an input or output. If OMx:OLx is not 0:0, then the output compare function overrides the use of the pin for general-purpose input/output. An attempt to write to the parallel data register (PDR) will not affect the pin. The value is saved in the PDR and driven out on the pin if the output compare function is later disabled (OMx:OLx = 0:0) and PDDRx is configured as an output.

### 3.3.2 Output Compare Registers (TOC1–4) (TI4O5)

All output compare registers are 16-bit read/write registers which are initialized to $FFFF by reset. Refer to Figure 3-7. If an output compare register is not used for an output compare function, it can be used as a storage location. A read or write must be a 16-bit operation to ensure coherency. If coherency is not needed, byte accesses can be used.

For output compare functions, 16-bit read/write output compare registers TOC1–TOC4 and TI4O5 are written to a desired match value and compared against TCNT to control specified pin actions.

**TOC1–TOC4** — Output Compare Registers 1–4 **$YFF914, $YFF916, $YFF918, $YFF91A**

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

**TI4O5** — Input Capture 4/Output Compare 5 Register **$YFF91C**

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

**Figure 3-7. Output Compare Registers (TOC1–4/TI4O5)**

### 3.3.3 Output Compare 1 (OC1)

Output compare 1 is unique because it can automatically affect any or all of the five output compare pins (OC1–OC5) as a result of a successful output match. The two registers that control this capability are the OC1 action mask register (OC1M) and the OC1 action data register (OC1D). Refer to Figure 3-8.

Register OC1M specifies the output pins that are affected as a result of a successful OC1 compare; register OC1D specifies the data to be transferred to the affected pins. If an OC1 match and another output match occur at the same time and both attempt to alter the same pin, the OC1 function controls the state of the pin.

This function allows control of multiple I/O pins automatically with a single output match. It also allows more than one output compare function to control the state of a single I/O pin. This allows pulses as short as one timer count to be generated.

**OC1M/OC1D** — OC1 Action Mask Register/OC1 Action Data Register **$YFF908**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC1M7 | OC1M6 | OC1M5 | OC1M4 | OC1M3 | 0 | 0 | 0 | OC1D7 | OC1D6 | OC1D5 | OC1D4 | OC1D3 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: The address shown is on a word boundary; however, the register can be accessed by a word or byte cycle.

**Figure 3-8. Action Mask and Action Data Registers (OC1M/OC1D)**

OC1Mx — OC1 Mask Bits

0 = Corresponding bit in the parallel data port is not affected by OC1 compare.
1 = Corresponding bit in the parallel data port is affected by OC1 compare.

OC1Dx — OC1 Data Bits

0 = If OC1Mx is set, transfer 0 to the corresponding parallel data port bit on OC1 match.
1 = If OC1Mx is set, transfer 1 to the corresponding parallel data port bit on OC1 match.

An interrupt can also be generated on a successful output compare if the corresponding interrupt enable bit (OCxI) is set in TMSK1.

If not needed by the output compare one function, the OC1 pin can be used to output the clock selected as the input to the timer counter register (TCNT). Note that this clock does not have a 50% duty cycle. For more details on this feature, refer to the functional description of the prescaler in **5.1 Prescaler.**

### 3.3.4 Timer Compare Force Register (CFORC)

The CFORC register (Figure 3-9) allows forced early compares. FOC5–FOC1 correspond to the five output compares. These bits are set for each output compare that is to be forced. The action taken as a result of a forced compare is the same as if there was a match between the OCx register and the free-running counter, except that the corresponding interrupt status flag bits are not set. The forced channels will trigger their programmed pin actions to occur immediately after the write to CFORC.

The CFORC register is implemented as the upper byte of a 16-bit register which also contains the PWM control register (PWMC). It can be accessed as 8 bits or a word access can be used. Reads of the force compare bits (FOCx) have no meaning and always return zeros. These bits are self negating.

The CFORC bits should not be used on an output compare function that is programmed to toggle its output on a successful compare, because a normal compare occurring immediately before or after the force may result in undesirable operation.

**CFORC** — Compare Force Register **$YFF924**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FOC5 | FOC4 | FOC3 | FOC2 | FOC1 | 0 | FPWMA† | FPWMB† | * | * | * | * | * | * | * | * |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* These bits are part of PWMC.

† Refer to the **SECTION 6 PULSE-WIDTH MODULATION (PWM) UNIT** for information on FPWMA and FPWMB.

NOTE: The address shown is on a word boundary; however, the register can be accessed by a word or byte cycle.

**Figure 3-9. Timer Compare Register (CFORC)**

FOCx — Force Output Compare Bits

0 = Has no meaning

1 = Causes pin action programmed for OCx, except that the OCxF flag is not set

FPWMx — Force PWM Value

0 = PWM pin x is used for PWM functions; normal operation.

1 = PWM pin x is used for discrete output.

For more information on the FPWMA and FPWMB bits refer to **SECTION 6 PULSE-WIDTH MODULATION (PWM) UNIT.**

### 3.4 Input Capture 4/Output Compare Function 5 (IC4/OC5)

The input capture 4/output compare 5 pin has multiple functions. As discussed in the previous sections, this pin can be used as an input capture pin, an output compare pin or as a general-purpose I/O pin. These features are controlled by the DDRI4O5 bit in the parallel data direction register (PDDR) and a function enable bit I4O5 in the pulse accumulator control register (PACTL). The function enable bit configures the pin for the input capture (IC4) or output compare function (OC5). After reset both bits are cleared to zero configuring the pin as an input, but also enabling the OC5 function. When the OC5 function is programmed to produce an output on the IC4/OC5 pin (OM5:OL5 bits non-zero in TCTL1), DDRI4O5 in the PDDR is overridden. In all other aspects, OC5 works the same as the other output compare functions.



This pin is configured as input capture function 4 (IC4) if the function enable bit I4/O5 of the PACTL register is set to one. If DDRI4O5 is set to one (pin configured as an output) and IC4 is enabled, writes to the pin can cause an input capture event if the proper edge is selected. Except as noted, IC4 works the same as the other input capture functions.

The 16-bit register used with the IC4/OC5 function acts as either the input capture register or as the output compare register depending on which function is selected. When used as the input capture 4 register, it cannot be written to except in Test or Freeze mode. Refer to **SECTION 9 SPECIAL MODES** for information on Freeze and test mode.

# SECTION 4
# PULSE ACCUMULATOR

## 4.1 Pulse Accumulator

The pulse accumulator counter (PACNT) is an 8-bit read/write up counter register that can operate in an external event counting or gated time accumulation mode. The bits in the pulse accumulator control register (PACTL) control the operation of PACNT. In the event counting mode the counter increments each time a selected edge (PEDGE bit) on the pulse accumulator input (PAI) pin is detected. Figure 4-1 shows a block diagram of the pulse accumulator.

Because the PACNT register can be accessed at any time, a value representing the number of edges to be counted can be written to it. As the edges are counted the counter will approach $FF and finally roll over to $00. If interrupts are enabled, an overflow interrupt will be generated.

The maximum clocking rate is the system clock divided by four. In the gated time accumulation mode there are four clock sources available. Bits PACLK1–0 select which of these sources will be used. The selected clock increments the PACNT when PAI is in the active state. The PAMOD and PEDGE control bits (Table 4-1) determine the active state. Refer to Table 4-2 for the bit patterns and clocks available to the pulse accumulator.

The PACTL and PACNT registers are implemented as one 16-bit register, but may be accessed with byte or word access cycles. Both registers are cleared at reset, but the PAIS and PCLKS bits will show the state of the PAI and PCLK pins, respectively.

Two maskable interrupts are available from the pulse accumulator. The pulse accumulator overflow flag indicates that the pulse accumulator count has rolled over from $FF to $00. This can be used to extend the range of the counter beyond 8 bits.

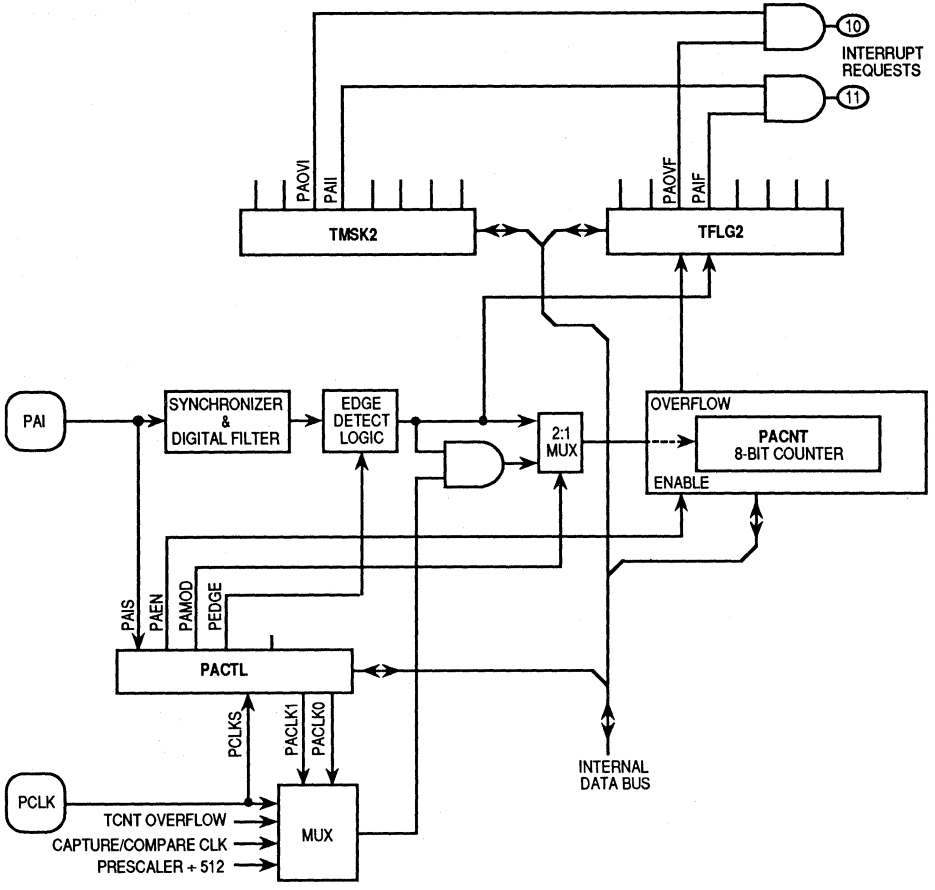



**Figure 4-1. Pulse Accumulator Block Diagram**

The pulse accumulator flag indicates that a selected edge is detected at the PAI pin. In the event counting mode, this second interrupt is generated when the edge being counted is detected. In gated time accumulation mode, it is generated at the end of the timing period, i.e., when the PAI input changes from the active to the inactive state. Hardware interrupt requests for these two conditions are enabled by the PAOVI and PAII bits in the TMSK2 register. The status bits PAOVF and PAIF bits are located in the TFLG2 register. These two bits indicate when the above events have occurred. For more information on interrupt operation refer to **7.1 Interrupts.**

If not needed for the pulse accumulator function, the PAI pin can be used for general-purpose input. The state of the PAI pin is reflected by the state of the

PAIS bit in the PACTL register. For more information on general-purpose I/O refer to **8.1.1 Uni-Directional I/O.**

### 4.1.1 Pulse Accumulator Register/Pulse Accumulator Counter (PACTL/PACNT)

The PACTL register (Figure 4-2) is used to select the operational mode of the pulse accumulator. PACNT is the pulse accumulator 8-bit counter. Control bit I4/O5 configures input capture 4/output compare 5 pin as an input capture or output compare. Refer to **3.4 Input Capture 4/Output Compare Function 5 (IC4/OC5).**

**PACTL/PACNT** — Pulse Accumulator Control Register/Pulse Accumulator Counter **$YFF90C**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAIS | PAEN | PAMOD | PEDGE | PCLKS | I4/O5* | PACLK1 | PACLK0 | PULSE ACCUMULATOR COUNTER | | | | | | | |
| RESET: | | | | | | | | | | | | | | | |
| U | 0 | 0 | 0 | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Refer to **3.4 Input Capture 4 /Output Compare Function 5 (IC4/OC5)** for information on this bit.

NOTE: The address shown is on a word boundary; however, the register can be accessed by a word or byte cycle.

**Figure 4-2. Pulse Accumulator Control Register/Pulse Accumulator Counter (PACTL/PACNT)**

PAIS — PAI Pin State (Read-Only)

PAEN — Pulse Accumulator System Enable
- 0 = Pulse accumulator disabled
- 1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode
- 0 = External event counting
- 1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control

This bit has different meanings based on the state of the PAMOD bit.

**Table 4-1. Pulse Accumulator Mode Select**

| PAMOD | PEDGE | Action on Clock |
|---|---|---|
| 0 | 0 | PAI falling edge increments counter. |
| 0 | 1 | PAI rising edge increments counter. |
| 1 | 0 | A zero on PAI inhibits counting (gated mode). |
| 1 | 1 | A one on PAI inhibits counting (gated mode). |

PCLKS — PCLK Pin State (Read-Only)

I4/O5 — Input Capture 4/Output Compare 5

0 = Output compare 5 function enabled

1 = Input capture 4 function enabled

Refer to **3.4 Input Capture 4/Output Compare Function 5 (IC4/OC5)** for details on the operation of this function.

PACLK1–0 — Pulse Accumulator Clock Select (Gated Mode)

4

**Table 4-2. Gated Mode Clock Source**

| PACLK[1:0] | Pulse Accumulator Clock Selected |
|---|---|
| 00 | System Clock Divided by 512 |
| 01 | Same Clock Used to Increment TCNT |
| 10 | TOF Flag from TCNT |
| 11 | External Clock, PCLK |

Pulse Accumulator Counter

This is an 8-bit read/write counter used for external event counting or gated time accumulation.

# SECTION 5
# PRESCALER

## 5.1 Prescaler

Both the capture/compare and the PWM units have their own independent 16-bit free-running counters as the main timing component. These counters derive their clocks from the prescaler or the external input pin PCLK. Figure 5-1 shows the block diagram of the prescaler.



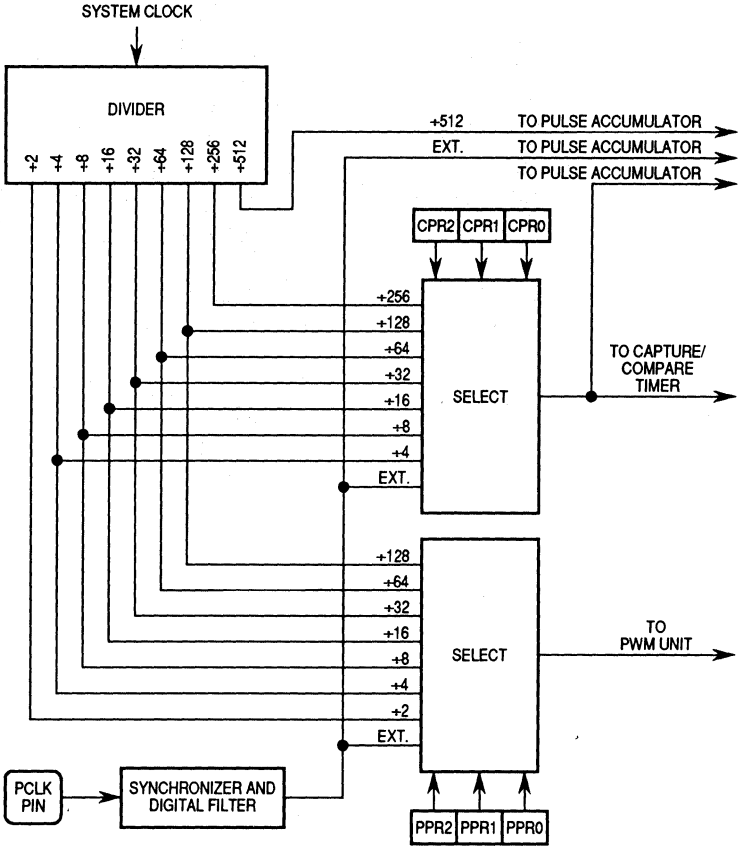



**Figure 5-1. Prescaler Block Diagram**

The system clock is divided by a nine-stage divider chain which provides outputs of the system clock divided by 2, 4, 8, 16, 32, 64, 128, 256, and 512. Connected to the outputs of the divider are two multiplexers, one for the compare/capture unit, the other for the PWM unit. These provide one of seven taps, plus an external input from the PCLK pin to the PWM and compare/capture units. The outputs of the multiplexers (muxs) are controlled by bits CPR[2:0] in the timer interrupt mask register 2 (TMSK2) for TCNT and bits PPR[2:0] in the PWM control register (PWMC) for the PWM counter (PWMCNT). Table 3-1 and 3-2 show the encoding of these bits. Refer to **7.1.3.1 Timer Interrupt Mask Registers 1–2 (TMSK1/TMSK2)** for more information on TMSK2 and **6.3.1 PWM Control Register (PWMC)** for details on this register.

5

**Table 5-1. Prescaler Select for Compare/Capture Unit**

| CPR [2:0] | System Clock Divide-By Factor |
|---|---|
| 000 | 4 |
| 001 | 8 |
| 010 | 16 |
| 011 | 32 |
| 100 | 64 |
| 101 | 128 |
| 110 | 256 |
| 111 | PCLK* |

* PCLK is an external clock input pin.

**Table 5-2. Prescaler Select for PWM Unit**

| PPR[2:0] | System Clock Divide-By Factor |
|---|---|
| 000 | 2 |
| 001 | 4 |
| 010 | 8 |
| 011 | 16 |
| 100 | 32 |
| 101 | 64 |
| 110 | 128 |
| 111 | PCLK* |

* PCLK is an external clock input pin.

After reset, these bits are cleared, and the GPT is configured to use the system clock divided by 4 for TCNT, and the system clock divided by 2 for PWMCNT.

Initialization software may change the division factor by writing to these bits. The bits for the PWM unit can be written at any time, but the bits for the compare/capture unit can only be written once except when the GPT is in test or freeze mode.

**Note**

Changing the prescaler control bits while the prescaler is running may cause an extra count if the input clock previously selected was a logic level 0, and the new input clock logic level is 1.

Refer to **SECTION 9 SPECIAL MODES** for information on how to stop the prescaler.

The 9-bit prescaler may be read at any time. In test or freeze mode the prescaler can be written. Word accesses must be used to ensure coherency. If coherency is not needed byte accesses can be used. The value of the prescaler will be in bits 8:0; whereas, bits 15:9 are unimplemented and will be read as zeros.



**Note**

Changing the value of the prescaler while it is running may cause an extra count if the prescaler tap bit selected is at a logic level 0, and the new value written is a logic level 1.

Two bits, STOPP and INCP in the GPT module configuration register (GMCR), can be used to more directly control the prescaler, i.e., for testing purposes. When set the STOPP bit stops the prescaler from counting. The INCP bit increments the prescaler by one count if it is stopped. This bit is self negating. There should be at least one bus cycle between writing the prescaler control bits and setting the INCP bit while single-stepping (while the STOPP bit is set) to guarantee proper operation of the single-stepping feature. Writing this sequence on successive cycles may result in unpredictable behavior. Refer to **SECTION 9 SPECIAL MODES**.

Another feature of the GPT is the capability to output the selected prescaler outputs (including the PCLK pin) from the two multiplexers to external pins. The CPROUT bit in the TMSK2 register configures the OC1 pin to output the TCNT clock from the timer mux and the PPROUT bit in the PWMC register configures the PWMA pin to output the PWM clock from the PWM mux. These two bits can be written at any time. The clock signals on OC1 and PWMA do not have a 50% duty cycle. They have the period of the selected clock, but are high for only one system clock time.

The prescaler also supplies three clock signals to the pulse accumulator clock select mux. These are the system clock divided by 512, the external clock signal from the PCLK pin, and the compare/capture clock signal.

5

# SECTION 6
# PULSE-WIDTH MODULATION (PWM) UNIT

The pulse width modulation (PWM) unit has two PWM outputs, PWMA and PWMB, which are controlled by the same clock output of the prescaler multiplexer (mux). This clock is the input to a 16-bit counter which is used by both of the PWM channels. The output of the counter is multiplexed to provide two operational modes: fast mode or slow mode. This gives a clocking rate 1/256 (fast mode) or 1/32768 (slow mode) of the output of the prescaler mux. The duty cycle ratios of the two PWM channels are individually controlled by software. The PWM pins can be used as output pins if not used for PWM functions. The PWMA pin may also be used to output the clock used to drive the PWM counter. Figure 6-1 is a block diagram of the pulse-width modulation unit.



## 6.1 Counter

The 16-bit counter in the PWM unit is similar to the timer counter in the compare/capture unit. After reset, the GPT is configured to use the system clock divided by two as the input to the free-running counter. Initialization software may reconfigure the counter to use one of seven prescaler outputs or an external clock from the PLCK input pin. Refer to **5.1 Prescaler** for additional information on the prescaler.

Software can read the counter at any time without affecting its value. A read of the PWM count register (PWMCNT) must be a word access to ensure coherency, but byte accesses can be made if coherency is not needed. The counter is cleared to $0000 during reset and is a read-only register except in freeze or test mode. Refer to **SECTION 9 SPECIAL MODES**. Any value may be written to the counter when in either of these modes. Note, however, that any writes to PWNCNT which result in changing the most significant bit of the 8 bits used for the PWM frequency from a logic level one to a logic level zero, will be seen as a zero detect by the PWM logic.

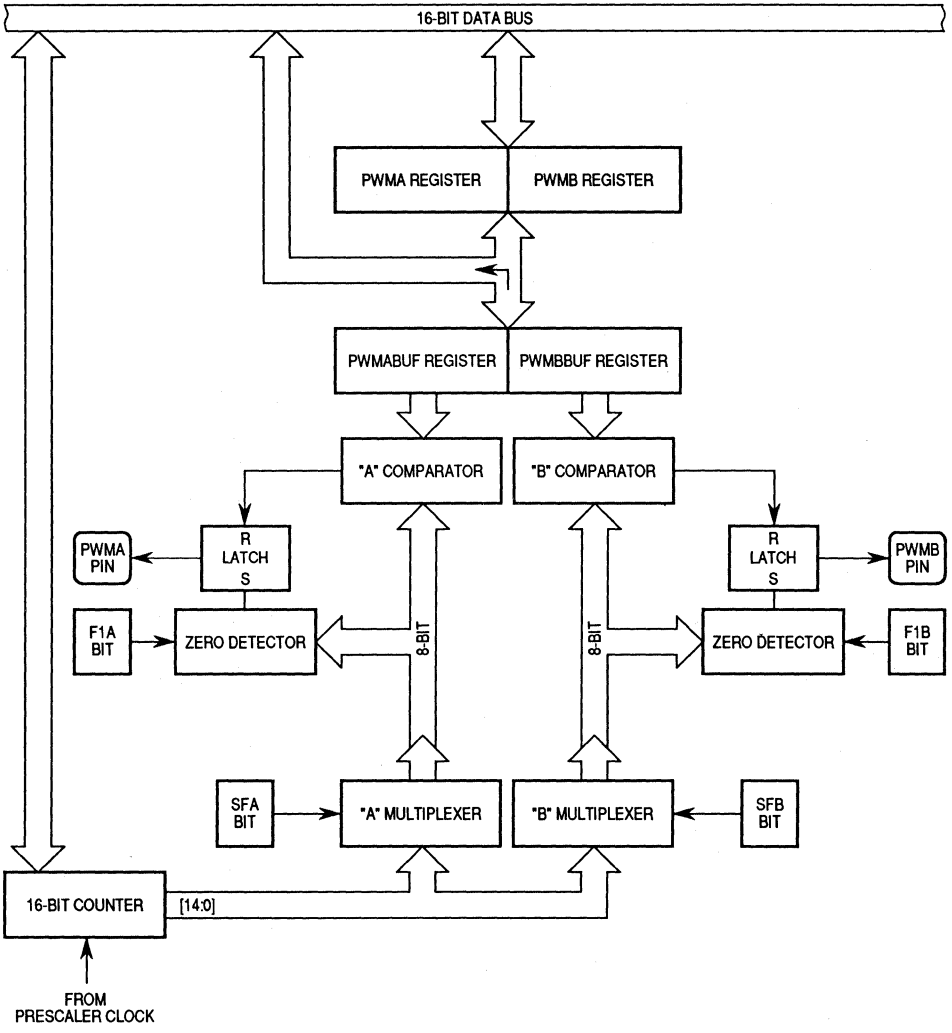



**Figure 6-1. PWM Block Diagram**

Fifteen of the 16 bits of the counter are output to multiplexers A and B which independently provide the fast and slow modes of the PWM unit. The fast/slow mode is selected by the SFA bit for PWMA and the SFB bit for PWMB in the PWM control register (PWMC). These two bits and the PPR[2:0] bits in the PWMC register control the output frequency of the PWM unit. In the fast mode, bits [7:0] of PWMCNT clock the PWM logic; in slow mode bits [14:7] are used to clock the PWM logic. This makes the period of a PWM channel in slow mode

128 times longer than when in fast mode. Figure 6-2 shows fast and slow modes of one of the channels. Table 6-1 shows a range of PWM output frequencies using a 16.78-MHz system clock.

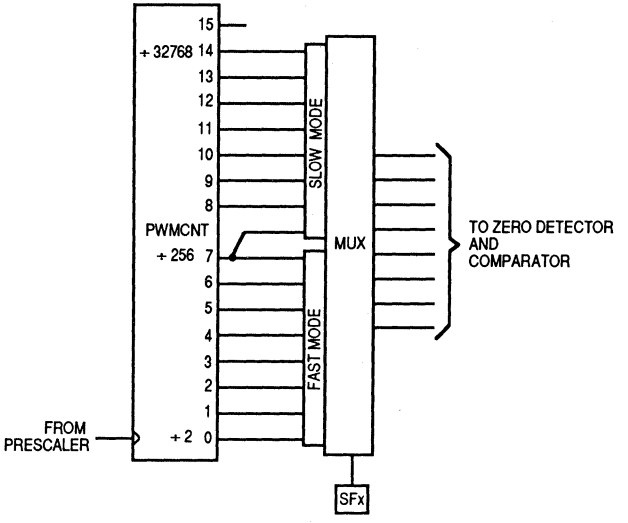



**Figure 6-2. Fast/Slow Mode**

**Table 6-1. PWM Frequency Range Using 16.78-MHz System Clock**

| PPR[2:0] | Prescaler Tap | Fast Mode | Slow Mode |
|---|---|---|---|
| 000 | Div 2 = 8.39 MHz | 32.8 kHz | 256 Hz |
| 001 | Div 4 = 4.19 MHz | 16.4 kHz | 128 Hz |
| 010 | Div 8 = 2.10 MHz | 8.19 kHz | 64.0 Hz |
| 011 | Div 16 = 1.05 MHz | 4.09 kHz | 32.0 Hz |
| 100 | Div 32 = 524 kHz | 2.05 kHz | 16.0 Hz |
| 101 | Div 64 = 262 kHz | 1.02 kHz | 8.0 Hz |
| 110 | Div 128 = 131 kHz | 512 Hz | 4.0 Hz |
| 111 | PCLK | PCLK/256 | PCLK/32768 |

## 6.2 PWM Function

The pulse-width values of the PWM outputs are associated with registers PWMA and PWMB. Each of these registers is 8 bits in length. They are implemented as two bytes of a 16-bit register and may be accessed as separate bytes or as one 16-bit register. A value of $00 loaded into either of these registers results in a continuously low output on the corresponding output pin. A value of $80 results in a 50% duty cycle output to the maximum value of $FF, which corresponds to an output which is at "1" for 255/256 of the cycle. A 100% duty cycle is available by using the F1A (for PWMA) and F1B (for PWMB) bits in the PWMC register. The F1A and F1B are the lower two bits of the PWMC register. Setting these bits to a one forces a continuously high output on the corresponding output pins after the end of the current PWM cycle. Returning the F1x bit to a zero returns the PWM to its normal mode of operation.

When the MCU writes to register PWMA or PWMB, the new value will only be picked up at the end of a complete cycle. This prevents the PWM from generating glitches (erroneous data) when being updated. The current duty cycle value is in the PWMx buffer register (PWMxBUF). The new value is transferred from the PWMx register to PWMxBUF at the end of the current cycle.



The data and control registers, PWMA, PWMB, and PWMC are reset to $00 during reset. These registers may be written or read at any time. PWMC is implemented as the lower byte of a 16-bit register. The upper byte is the CFORC register. The buffer registers, PWMABUF and PWMBBUF, are read-only at all times and can be accessed as separate bytes or as one 16-bit register.

## 6.3 PWM Registers

This section provides additional information on the registers in the PWM unit.

### 6.3.1 PWM Control Register (PWMC)

The PWMC register controls functioning of the PWM unit. Refer to Figure 6-3.

**PWMC** — PWM Control Register **$YFF924**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FOC5* | FOC4* | FOC3* | FOC2* | FOC1* | 0 | FPWMA† | FPWMB† | PPROUT | PPR2 | PPR1 | PPR0 | SFA | SFB | F1A | F1B |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* These bits are part of the CFORC register. Refer to **3.3.4 Timer Compare Force Register (CFORC)** for details on these bits.

† FPWMA and FPWMB are also considered part of CFORC; however, they affect operation of the PWM unit and can be accessed as part of PWMC when word accesses are used.

NOTE: The address shown is on a word boundary; however, the register can be accessed by a word or byte cycle.

**Figure 6-3. PWM Control Register (PWMC)**

FPWMx — Force PWM Value

0 = PWM pin x is used for PWM functions; normal operation.

1 = PWM pin x is used for discrete output. The value of the F1x bit will be driven out on the PWMx pin. This is true for PWMA regardless of the state of the PPROUT bit.

FPWMA and FPWMB are considered part of CFORC; however, they affect operation of the PWM unit and can be accessed as part of PWMC when word accesses are used.

PPROUT — PWM Prescaler Clock Output Enable

0 = Normal PWM operation on PWMA.

1 = Output of prescaler mux for PWM counter is driven out on the PWMA pin.

If not needed for the PWM function, the PWMA pin can be used to output the clock selected as the input to the PWM count register. The clock does not have a 50% duty cycle. It reflects the period of the selected clock and a pulse high time equal to one system clock time. Also note that if the FPWMA bit in CFORC register is set, the value of the F1A bit in PWMC will be driven out on the pin, regardless of the state of PPROUT. Refer to **5.1 Prescaler** for more information on the prescaler.

PPR[2:0] — PWM Prescaler

These bits select a prescaler tap or the external clock, PCLK, to be the input to the PWMCNT. Refer to Table 6-2. These bits are read/write and can be accessed at any time. Changing the prescaler control bits while the prescaler is running may cause an extra count if the input clock previously selected was at logic level zero, but the new input clock is at logic level one. Refer to **SECTION 9 SPECIAL MODES** for information on how to stop the prescaler.

The PPR bits are not double buffered, therefore, it is necessary to select their values before any writes to the PWM registers. Not doing so could temporarily output improper values from the PWM. Refer to **5.1 Prescaler** for more information on the prescaler.

**Table 6-2. PPR Bits**

| PPR [2:0] | System Clock Divide-By Factor |
|---|---|
| 000 | 2 |
| 001 | 4 |
| 010 | 8 |
| 011 | 16 |
| 100 | 32 |
| 101 | 64 |
| 110 | 128 |
| 111 | PCLK* |

* PCLK is the external clock input pin.

SFx — Slow/Fast Bits

The SFx bits allow the PWM channels to operate in either fast or slow mode.

0 = The higher speed of PWMx is selected. (The PWMx period is 256 PWMCNT increments long.)

1 = The slower speed of PWMx is selected. (The PWMx period is 32,768 PWMCNT increments long.)



F1x — Force Logic One on PWMx

0 = Normal PWMx operation or force a zero on the PWMx pin for discrete output.

1 = Force one on the PWMx pin; a 100% duty cycle

The F1x bit allows the user to force a value on the PWMA pin. This bit can be used to force a logic level of one or zero on the PWMx pin when used as general-purpose output. In this case, the F1x bit is driven out on the pin. Setting this bit during PWM generation forces a continuously high output on the PWMx pin after the end of the current cycle. This is a 100% PWM duty cycle. Resetting this bit causes normal PWM operation to resume on the pin after the current cycle.

F1x can also be read. When read, this bit reflects the state of the PWMx pin.

### 6.3.2 PWM Registers A/B (PWMA/PWMB)

**PWMA/PWMB** — PWM Registers A/B **$YFFF926, $YFFF927**

These read/write registers contain the pulse-width value of the next PWM output waveform on the corresponding PWM pin. A value of $00 loaded into one of these registers results in a continuously low output on the corresponding pin. A value of $80 results in a 50% duty cycle output to the maximum value of $FF. This maximum value corresponds to an output which is high for 255/256 of the period. If a 100% duty cycle is required, the F1x bit in the PWMC register can be set.

### 6.3.3 PWM Buffer Registers A/B (PWMBUFA/PWMBUFB)

**PWMBUFA/B** — PWM Buffer Registers A/B **$YFF92A, $YFF92B**

These read-only registers contain the values associated with the duty cycles of the corresponding PWMs in progress. They are updated from PWMA/B at the end of each PWM cycle.

## 6.4 PWM Pins

If not needed for PWM outputs, pins PWMA and PWMB can be used for general-purpose output. Two bits in the CFORC register control whether the pins are used for PWMs or for discrete output. If used for discrete output, the values of the F1A and F1B bits in the PWM control register (PWMC) will be driven out on these pins. This feature is separately selected for the two PWM pins. When read, the F1A and F1B bits reflect the states of the PWMA and PWMB pins, respectively. Refer to **8.1.1 Uni-Directional I/O** for additional information.

6

# SECTION 7
# INTERRUPTS

## 7.1 Interrupts

The GPT is capable of generating one of seven interrupt priority levels on the intermodule bus (IMB). The GPT contains control registers which set the interrupt priority level and the arbitration priority of the module, enable the interrupts of the 11 interrupt sources and adjust their priority internally.

The interrupt priority level of the GPT can be one of eight levels 7–0. This level is selected by the interrupt request level (IRL) bits in the interrupt configuration register (ICR). A level 7 is the highest priority and a level 0 disables interrupts. (Refer to Figure 7-3 for the location of these bits in the ICR.)

When an interrupt is requested and is at a higher level than the current interrupt level set by the interrupt mask in the CPU status register, the CPU initiates an interrupt acknowledge (IACK) cycle. The module will decode the IACK cycle and compare the CPU recognized level to the level that the module is currently requesting. Refer to the appropriate CPU manual for more information on the interrupt mask. If a match occurs, arbitration with other modules begins.



Interrupting modules present their arbitration ID on the IMB and the module with the highest ID wins. If the GPT wins the arbitration, it generates a uniquely encoded interrupt vector that indicates which timer channel is requesting service. The encoding scheme is such that the high nibble of the interrupt vector (called the interrupt vector base address) comes from a 4-bit field in the interrupt configuration register (ICR). The encoded value of the low-order nibble, the interrupt source number, indicates which of the 11 interrupt sources of the GPT is requesting service. Figure 7-2 shows a block diagram of the interrupt hardware.

The arbitration ID for the GPT module is selected by the IARB bits in the GPT module configuration register (MCR) (Figure 7-1). The IMB is designed to arbitrate between a maximum of 16 modules, so the IARB field of the MCR can have a value from 0 through 15. A module with an IARB of 15 always wins the arbitration when two or more modules with the same interrupt level contend for the interrupt request. Each module should have a unique Arbitration ID.

**NOTE**

If two or more modules have the same arbitration ID and generate an interrupt at the same priority level, unpredictable operation can occur.

An ID of zero disables the module from arbitrating for the interrupt, and if the module does generate an interrupt and the CPU mask level is set at a lower value, a spurious interrupt exception is generated. Refer to the appropriate CPU manual for a discussion of exception processing. Refer to **SECTION 9 SPECIAL MODES** for additional details about the MCR.

**MCR** — Module Configuration Register **$YFF900**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STOP* | FRZ1* | FRZ0* | STOPP* | INCP* | 0 | 0 | 0 | SUPV* | 0 | 0 | 0 | IARB3 | IARB2 | IARB1 | IARB0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* Refer to **SECTION 9 SPECIAL MODES** for information on these bits.

**Figure 7-1. IARB Bits of MCR**

7

IARB [3:0] — Interrupt Arbitration ID

System software must set this field between $F–$1; $F is the highest priority. This field is initilized to zero during reset, which disables arbitration and causes interrupts generated by the GPT to be treated as spurious.

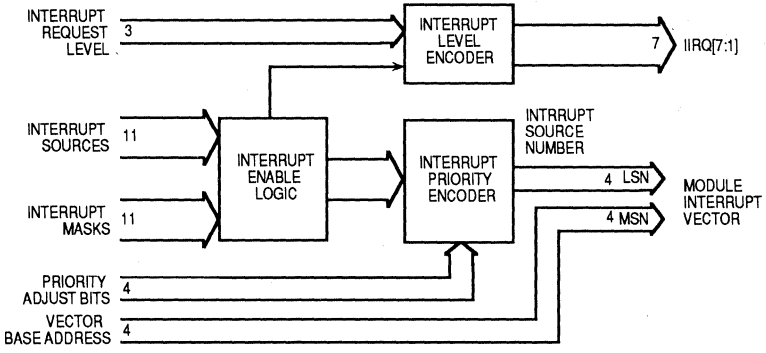



**Figure 7-2. GPT Interrupt Vector Generation**

For simultaneous interrupts, a hard-wired priority scheme is implemented. This scheme assures that the vector returned to the CPU will point to the interrupt with the highest priority. Refer to Table 7-1 for the priority arrangement, along with the associated vector address for each channel. Hardware prevents the vector from changing (due to a new interrupt) while it is being driven out on the IMB.

The priority adjust bits (PAB) in the interrupt configuration register (ICR) can change the priority of one of the channels. The user can place any single interrupt source at the highest priority level by changing the PAB value. Other than this one exception, all other priority relationships remain the same. Table 7-1 shows the priority relationships of the timer channels.

**Table 7-1. Timer Interrupt Priorities and Vector Addresses**

| Interrupt Source | | Priority Level | Vector Address |
|---|---|---|---|
| **Name** | **Function** | | |
| — | Adjusted Channel | 0 (Highest) | $X0 |
| IC1 | Input Capture 1 | 1 | $X1 |
| IC2 | Input Capture 2 | 2 | $X2 |
| IC3 | Input Capture 3 | 3 | $X3 |
| OC1 | Output Compare 1 | 4 | $X4 |
| OC2 | Output Compare 2 | 5 | $X5 |
| OC3 | Output Compare 3 | 6 | $X6 |
| OC4 | Output Compare 4 | 7 | $X7 |
| IC4/OC5 | Input Capture 4/Output Compare 5 | 8 | $X8 |
| TOF | Timer Overflow Flag | 9 | $X9 |
| PAOVF | Pulse Accumulator Overflow Flag | 10 | $XA |
| PAIF | Pulse Accumulator Input Flag | 11 (Lowest) | $XB |

X = 4-bit vector base address (VBA)



### 7.1.1 GPT Interrupt Configuration Register (ICR)

The ICR controls the interrupt operation of the GPT.

**ICR** — Interrupt Configuration Register **$YFF904**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAB | | | | 0 | IRL | | | VBA | | | | 0 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 7-3. GPT Interrupt Configuration Register (ICR)**

PAB — Priority Adjust Bits

The GPT has 11 sources capable of generating an interrupt. These bits specify the single interrupt source that has the highest priority level. All of the other interrupt sources maintain their relative priority.

For example, a $4 written to PAB will make OC1 the highest priority channel. When OC1 generates an interrupt, the low nibble of the vector will be $0. The remaining channels maintain their original relative priority and vector addresses.

IRL — Interrupt Request Level

These bits specify the priority level of the GPT module's interrupts. The GPT can have any of eight priority levels. Level 7 is the highest priority and is nonmaskable. Level zero disables interrupts. The interrupt request level specifies the priority level presented to the CPU. The interrupt request level is initialized to zero during reset.

All GPT internal interrupts are prioritized as shown in Table 7-1. The interrupt with the highest priority generates the interrupt level to the IMB specified by this field.

VBA — Interrupt Vector Base Address

This field specifies the most significant nibble of all the vector numbers that can be generated by the different interrupt sources of the GPT module. This value concatenated with the vector address shown in Table 7-1 is the module interrupt vector. The "X" in the table represents the VBA.

7

### 7.1.2 Interrupt Status Flags

When an event occurs in the GPT (such as a timer overflow, an input capture, or any event that could generate an interrupt), that event sets a status flag in the timer interrupt flag registers (TFLG1/TFLG2). User software can read the status registers to detect an event. If an event occurred, the polling routine can transfer control to a software routine that services that event.

If interrupts are enabled for that event, the CPU enters an interrupt service routine. Using interrupts does not require continuously polling the status flags to detect the occurrence of an event.

The interrupt status flags must be cleared in a particular sequence. A read operation must first be executed on the asserted status flag and then a write of the negated state must be executed. With the GPT, all of the status flags are at a logic level one when asserted; therefore, a logic level zero must be written to clear the flag. If a new event occurs between the CPU reading the status and clearing it, the flag will not be cleared, indicating the occurrence of a new event.

The GPT, and never the CPU, asserts the interrupt status flags. The CPU can only clear them. The term "clear" means to negate or, in the present case, set the flag to a logic level zero. Figure 7-4 shows the timer interrupt flag registers.

### 7.1.2.1 Timer Interrupt Flag Registers 1–2 (TFLG1/TFLG2)

The timer interrupt flag register indicates when an event occurs in the GPT and is divided into two 8-bit registers: TFLG1 and TFLG2. The registers can be addressed as one 16-bit register or as individual 8-bit registers. The registers initialize to zero at reset.

This register along with the timer interrupt mask registers (TMSK1/TMSK2) allow the GPT to operate in a polled or interrupt-driven system. For each bit in TFLGx there is a corresponding bit in the TMSKx register in the same bit position. If the mask bit is set and an associated event occurs, a hardware interrupt request is generated. Refer to **7.1.3 Enabling Interrupts** for more detail.

**TFLG1/TFLG2** — Timer Interrupt Flag Registers 1–2 **$YFF922**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I4O5F | OC4F | OC3F | OC2F | OC1F | IC3F | IC2F | IC1F | TOF | 0 | PAOVF | PAIF | 0 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 7-4. Timer Interrupt Flag Registers (TFLG1/TFLG2 )**

OCxF — Output Compare x Flag
This flag is set each time TCNT matches the value in output compare register x.

ICxF — Input Capture x Flag
This flag is set each time a selected edge is detected at the input capture x pin.

I4O5F — Input Capture 4/Output Compare 5 Flag
If the I4/O5 pin is configured as input capture 4, this flag is set each time a selected edge is detected at the I4/O5 pin. If the I4/O5 pin is configured as output compare 5, this flag is set each time TCNT matches the value in output compare register 5.

TOF — Timer Overflow Flag
This flag is set each time TCNT advances from a value of $FFFF to $0000.

PAOVF — Pulse Accumulator Overflow Flag
This flag is set each time the pulse accumulator counter advances from a value of $FF to $00.

PAIF — Pulse Accumulator Flag
In event counting mode, this bit is set when an active edge is detected on the PAI pin. In gated time accumulation mode, this bit is set at the end of the timed period (when going from the active (counting) state to the inactive (no longer counting) state).

### 7.1.3 Enabling Interrupts

Even though the interrupt level in the ICR register may be set to a level between one and seven, the GPT will not generate an interrupt to the CPU unless the individual channels have their respective interrupts enabled. However, the GPT can still operate in polled mode, as described above.

Interrupts are enabled by setting the interrupt enable bits in timer interrupt mask register (TMSK1–2). The bit indicating the occurrence of an event is set in the TFLG register. This is the interrupt status flag. If the corresponding bit in the TMSK register is set, the CPU will receive a hardware interrupt request.

Note that upon exit from an interrupt service routine, the corresponding interrupt status flag must be cleared. If not cleared, the same interrupt will be pending, and the CPU will respond as if a new interrupt occurred.

#### 7.1.3.1 Timer Interrupt Mask Registers 1–2 (TMSK1/TMSK2)

The timer interrupt mask register (Figure 7-5) enables specific interrupts in the GPT. It is divided into two 8-bit registers: TMSK1 and TMSK2. The registers can be addressed as one 16-bit register or as individual 8-bit register. The registers initialize to zero at reset.



This register also controls the operation of the timer prescaler. Refer to **5.1 Prescaler** for additional details on prescaler operation and control.

**TMSK1/TMSK2** — Timer Interrupt Mask Registers 1–2 **$YFF920**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I4O5I | OC4I | OC3I | OC2I | OC1I | IC3I | IC2I | IC1I | TOI | 0 | PAOVI | PAII | CPROUT* | CPR2* | CPR1* | CPR0* |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* These bits control the operation of the compare/capture mux in the prescaler. Refer to **5.1 Prescaler** for information on these bits.

**Figure 7-5. Timer Interrupt Mask Registers 1–2 (TMSK1/TMSK2)**

OCxI — Output Compare x Interrupt Enable
- 0 = OCx interrupts disabled
- 1 = OCx interrupts requested when OCxF flag is set

ICxI — Input Capture x Interrupt Enable
- 0 = ICx interrupts disabled
- 1 = ICx interrupts requested when ICxF flag is set

I4O5I — Input Capture 4/Output Compare 5 Interrupt Enable
- 0 = IC4 or OC5 interrupt disabled (depending on I4/O5 pin function)
- 1 = IC4 or OC5 interrupt requested when I4O5I flag is set (depending on I4/O5 pin function)

TOI — Timer Overflow Interrupt Enable
- 0 = Timer overflow interrupts disabled
- 1 = Interrupts requested when TOF flag is set

PAOVI — Pulse Accumulator Overflow Interrupt Enable
- 0 = Pulse accumulator overflow interrupts disabled
- 1 = Interrupts requested when PAOVF flag is set

PAII — Pulse Accumulator Interrupt Enable
- 0 = Pulse accumulator interrupts disabled
- 1 = Interrupts requested when PAIF flag is set

7

# SECTION 8
# GENERAL-PURPOSE I/O

Any GPT pin, if not used for a GPT function, can be used as general-purpose I/O. Some pins are bi-directional, others are output only or input only. The bi-directional pins are associated with the compare/capture functions.

## 8.1 General-Purpose I/O

Pins associated with input capture and output compare functions can be used either for timing functions or as general-purpose I/O pins. All are bi-directional when used for general-purpose I/O; together they are treated as an 8-bit parallel data port. The direction of each pin is controlled by the corresponding data direction bit in the parallel data direction register (PDDR). Refer to Figure 8-1.

When the bits in the PDDR are set to zero, the corresponding bits in the PDR are configured as inputs. A one configures the bits as outputs.

Parallel data is read from and written to the parallel data register (PDR) (Figure 8-1). Pin data can be read from the PDR, even when the pins are configured for an alternate timer function. Data read from the parallel data register always reflects the state of the external pin; whereas, data written to the parallel data register may not always affect the external pin.



Data written to the PDR does not directly affect pins used by the output compare functions, but the data is latched so that if the output compare function is later disabled, the last data written to the PDR is driven out on the associated output pin (if the pin is configured as an output pin). Data written to the PDR can cause input captures if the corresponding pin is configured as an input capture function.

**PDDR/PDR** — Parallel Data Direction Register/Parallel Data Register **$YFF906**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DDRI4O5 | DDRO4 | DDRO3 | DDRO2 | DDRO1 | DDRI3 | DDRI2 | DDRI1 | IC4/OC5 | OC4 | OC3 | OC2 | OC1 | IC3 | IC2 | IC1 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 8-1. Parallel Data Direction/Parallel Data Registers (PDDR/PDR)**

DDRx [15:8] — Data Direction for Input Capture/Output Compare Pins

0 = Input only

1 = Output

Bits [7:0] — Parallel Data Port

These bits represent input or output data at the pins depending on the state of the corresponding bit in the PDDR register.

For details on the operation of the input capture and output compare functions refer to **SECTION 3 COMPARE/CAPTURE UNIT.**

### 8.1.1 Uni-Directional I/O

The input capture and output capture pins are bi-directional. They can be input or outputs. There are two GPT pins that can serve as general-purpose input and two pins that can serve as outputs.

The pulse accumulator input (PAI) and the external clock input (PCLK) pins provide general purpose input. The state of these pins can be read by accessing the PAIS and PCLKS bits in the pulse accumulator control register (PACTL). Refer to Figure 8-2.

**PACTL** — Pulse Accumulator Control Register **$YFF90C**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAIS | PAEN* | PAMOD* | PEDGE* | PCLKS | I4/O5* | PACLK1* | PACLK0* | PUSLE ACCUMULATOR COUNTER* | | | | | | | |
| RESET: | | | | | | | | | | | | | | | |
| U | 0 | 0 | 0 | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* Refer to **4.1.1 Pulse Accumulator Register/Pulse Accumulator Counter (PACTL/PACNT)** for information on these bits.

**Figure 8-2. PAIS and PCLKS Bits**

PAIS — PAI Pin State (Read-Only)

PCLKS — PCLK Pin State (Read-Only)

The pulse-width modulation A and B (PWMA/PWMB) output pins can serve as general-purpose output pins. The force PWM value (FPWMx) and the force logic one (F1x) bits in the compare force (CFORC) and PWM control (PWMC) registers, respectively, control their operation. Refer to Figure 8-3 for bit locations.

The FPWMx bit must be written to a logic level one to allow the pins to be used as discrete ouputs. The logic level written to the F1x bits are then output to the corresponding pin.

**CFORC/PWMC** — Compare Force Register/PWM Control Register **$YFF924**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FOC5 * | FOC4 * | FOC3 * | FOC2 * | FOC1 * | 0 | FPWMA | FPWMB | PPROUT * | PPR2 * | PPR1 * | PPR0 * | SFA * | SFB * | F1A | F1B |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* Refer to **3.4.4 Timer Compare Force Register (CFORC)** and **6.3.1 PWM Control Register (PWMC)** for information on these bits.

**Figure 8-3. Force PWM Bits**

FPWMx — Force PWM Value

0 = PWM pin x is used for PWM functions; normal operation.

1 = PWM pin x is used for discrete output. The value of the F1x bit will be driven out on the PWMx pin.

F1x — Force Logic One on PWMx

0 = Normal PWMx operation or force a zero on the PWMx pin for discrete output

1 = Force one on the PWMx pin; a 100% duty cycle PWM or discrete output

8

# SECTION 9
# SPECIAL MODES

The GPT can enter a number of special modes in addition to its normal operational mode. These modes are test mode (selected by a bit in the SIM module configuration register (MCR)), stop mode, freeze mode and single step mode. These different modes are selected by bits in the MCR. One other mode, supervisor mode, allows system software to limit access to certain registers in the GPT.

## 9.1 Test Mode

The test mode is intended for factory production testing of the GPT and other modules within the specific MCU on which the GPT is incorporated. The description of the test mode is for informational purposes only and is not intended to provide operational information. Applications should avoid using test mode.

Test mode is entered by a combination hardware and software method. A register bit (MCR of SIM) must be set while an external pin ($\overline{\text{TSTME}}$) is asserted.

Test mode on the GPT is only used to allow write access to registers and bits that otherwise are read-only. Because the timing functions are simple, no other special test logic is included in this module. However, the test register location is reserved by Motorola for future use. Because all of the registers are accessible in various operational modes, the prescaler, counters, and control registers can be read and written by diagnostic software. Also, the outputs of the prescaler muxes can be driven out on external pins. With these simple features, the GPT can be easily tested. For more information on test mode refer to the test submodule section in the appropriate MCU manual.



## 9.2 Stop Mode

In STOP mode the system clock is stopped in most of the module and will remain stopped until the STOP bit is negated by the CPU or by a reset. All counters and prescalers within the timer will stop counting while the STOP bit is asserted. Only the module configuration register (MCR) and the interrupt configuration register (ICR) should be accessed while in the stop mode. Accesses to other GPT registers may result in unpredictable behavior. The STOP bit should be cleared independently of the other control bits in this register to guarantee proper operation. Changing the state of other bits while changing the state of the STOP

bit may result in unpredictable behavior. For example, clearing the STOP bit while setting the STOPP bit may result in incrementing the prescaler.

### 9.3 Freeze Mode

MCUs in Motorola's family of modular microcontrollers have an alternate operating mode in addition to their normal mode. This mode is the background debug mode (BDM). In BDM all normal processing is suspended and the user can view the contents of registers and memory locations and can execute other operations. Refer to the appropriate CPU manual to see what operations are available. BDM must be enabled at reset before it can be entered.

Several sources within the MCU cause the CPU to go into background debug mode. These sources can be an external breakpoint, the BGND instruction, or a double bus fault. Refer to the appropriate CPU manual for details. When the CPU enters background debug mode, the FREEZE signal on the IMB and the FREEZE pin are asserted. Assertion of FREEZE is the first indication that the CPU has entered BDM. While in BDM, each module in the MCU can independently enter freeze mode.

While in freeze mode, a snapshot of most of the internal registers is available, and certain write operations not normally allowed can be executed. Freeze mode freezes the current state of the timer. The prescaler and the pulse accumulator do not increment, and changes to the pins while in freeze mode are ignored. (The input synchronizers for the input pins are not clocked.) All other timer functions controlled by the CPU will operate normally; for example, registers can be written to change pin directions, force output compares, read or write I/O pins.

To enter freeze mode, the CPU must be in background debug mode, and the FRZ0 bit in the GPT MCR must be set to a logic level one. The FRZ0 bit can be set prior to the processor entering background debug mode. The FRZ1 bit can be read and written, but has no function and is reserved for future use.



The prescaler and the pulse accumulator will remain stopped, and the input pins will be ignored until the FREEZE signal is negated (the CPU is no longer in BDM), the FRZ0 bit is negated or the MCU is reset. Activities begun prior to the entering of freeze mode will be completed. For example, if an input edge on an input capture pin is detected just as the FREEZE signal is asserted, the capture occurs, and the corresponding interrupt flag is set. This occurs even if it takes a few clocks after the beginning of freeze mode.

While the FREEZE signal is asserted, the CPU has write access to registers and bits that are normally read-only, or write-once. The write-once bits can be written to as often as needed. It is not necessary for the FRZ0 bit to be set to allow write access to these registers and bits.

## 9.4 Single Step Mode (STOPP and INCP)

Two bits in the module configuration register (MCR) allow debugging of the GPT without the MCU entering BDM. As in freeze mode, the prescaler and the pulse accumulator stop counting, and changes at input pins are ignored when the STOPP bit is asserted. Reads of the GPT pins will return the state of the pin when the STOPP bit is set. After the STOPP bit is set, the INCP bit can be used to increment the prescaler and clock the input synchronizers once. The INCP bit is self-negating after the prescaler is incremented. The INCP bit can be repeatedly set and will increment the prescaler and input synchronizers each time. The INCP bit has no effect when the STOPP bit is not set.

There should be at least one bus cycle between writing prescaler control bits and setting INCP while single-stepping to guarantee proper operation. Writing back to back cycles may result in unpredictable behavior.

## 9.5 Supervisor Mode

Certain registers in the GPT are always in supervisor data space, and the other registers are programmable to be in supervisor or unrestricted data space. The supervisor (SUPV) bit in the MCR selects which space these programmable registers reside.

If the SUPV bit is set, the space is designated as supervisor only. Access is then permitted only when the CPU is operating in supervisor mode. Attempts to access the registers with user software (user data space) will cause the bus cycle to be transferred externally. Results can then be unpredictable depending on the hardware environment. If SUPV is clear, then both user and supervisor accesses are permitted. Attempting to access a supervisor-only register, i.e., ones denoted with an "S" from user software, causes the GPT to respond as if an unimplemented register was accessed. Writes have no effect and reads return zeros. Refer to the GPT register map, Table 1-1, to determine which registers are supervisor only or assignable to either data space.



**MCR** — GPT Module Configuration Register **$YFF900**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STOP | FRZ1 | FRZ0 | STOPP | INCP | 0 | 0 | 0 | SUPV | 0 | 0 | 0 | IARB3 | IARB2 | IARB1 | IARB0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 9-1. Module Configuration Register (MCR)**

STOP — Stop Clocks

0 = Internal clocks are not shut down.

1 = Internal clocks are shut down.

FRZ1 — This bit is not implemented at this time.

FRZ0 — FREEZE Response
- 0 = Ignore FREEZE.
- 1 = Freeze the current state of the GPT when FREEZE is asserted.

STOPP— Stop Prescaler
- 0 = Normal operation.
- 1 = Stop the prescaler and pulse accumulator from incrementing; ignore changes to input pins.

INCP — Increment Prescaler
- 0 = Has no meaning
- 1 = If STOPP is asserted, increment the prescaler once and clock the input synchronizers once.

SUPV — Supervisor/Unrestricted Data Space
- 0 = Registers with access controlled by the SUPV bit are unrestricted (FC2 is a don't care).
- 1 = Registers with access controlled by the SUPV bit are restricted to supervisor access.

IARB3–0 — Interrupt Arbitration ID
(Described in **7.1 Interrupts**)

# SECTION 10
# APPLICATIONS AND EXAMPLES

Microcontroller applications generally require interfacing to external events and then provide a response. This response may be based on the timing relationship between the event and a timing reference. A controller design may also require general-purpose I/O to select certain functions or to monitor the state of sensors such as limit switches.

Examples in this section show how the GPT can be used in different controller designs.

## 10.1 Electronic Motor Speed Control

An example of this timing relationship is the tachometer feedback signal in a motor controller circuit and the controller output signal that drives the motor. An input capture can automatically record the time of a tachometer pulse as each pulse occurs. This input capture can be used to determine the velocity and acceleration rate of the motor.

The value of the timer counter is latched into the input capture register each time a pulse is generated from the tachometer. The velocity can be determined by subtracting two successive input captures. This value multiplied by the time between TCNT counts gives the time between tach pulses. For example, if the MCU is operating at 16.77 MHz and the prescaler is set to divide by 256, the TCNT will be clocked at a 65536-Hz rate. This gives a time of 15.26 microseconds between each count. A scheme to drive a motor can be implemented by taking the ratio of the measured velocity in counts and a setpoint velocity in counts. This gives a value that represents the percentage of error from the setpoint in counts. This percentage of error can be used by the CPU to compute the value of a control signal used to control the velocity of the motor.



The tachometer can be any type that produces a digital signal. In Figure 10-1 an optical tach is used. The number of pulses produced per revolution should be such that the TCNT does not increment more than 65536 counts at the lowest speed. This keeps the software from having to consider the effects of the counter rolling over. Also, at the maximum motor speed, there must be adequate time between tach pulses for the TCNT to increment. The processor must also have enough time to service the interrupt before another interrupt occurs.

The output control signal to the motor driver can be a pulse-width modulated (PWM) signal. In the example, the motor is driven by a switching amplifier. The PWM signal switches the current on and off through the motor at the PWM signal's periodic rate. The average motor current is determined by the ratio of on time to off time. The longer the PWM signal is at logic level one, the higher the average motor current. The percentage of error term must be manipulated to take into account motor and load characteristics. The modified error term is used to change the current PWMx value. The new PWMx value keeps the motor at the proper velocity.

An unused function pin can be used as a general-purpose output to select the motor direction.

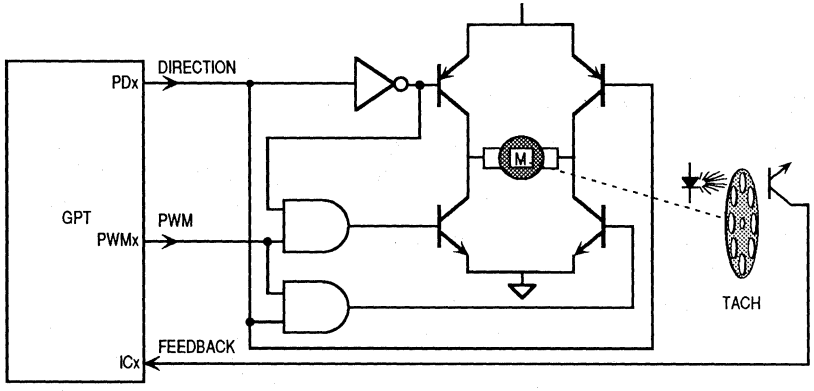



**Figure 10-1. Motor Control Example**

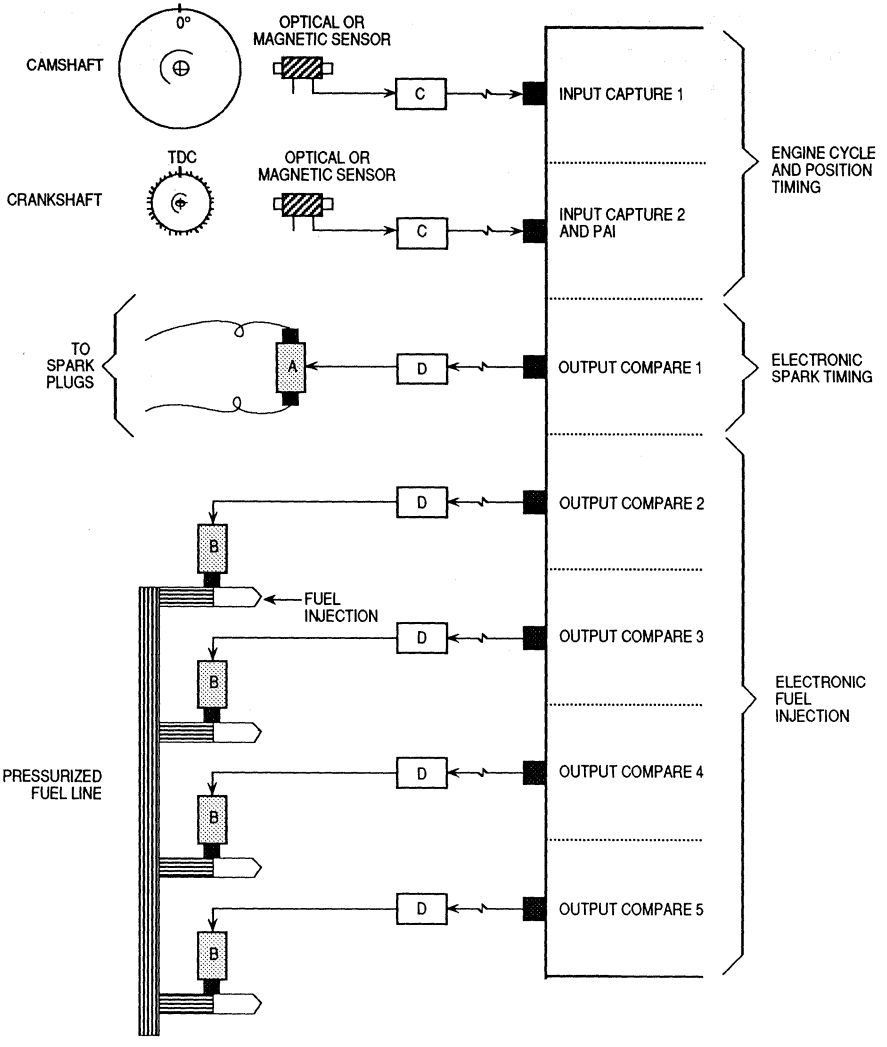
## 10.2 Engine Spark and Fuel Timing

10

A second application would be the use of timing sensors on an engine control system. A camshaft sensor gives engine cycle information, and a crankshaft sensor gives crankshaft position. The camshaft is coupled to the crankshaft and rotates at half the rate of the crankshaft. Piston number one top dead center (TDC) on the power stroke is indicated by an output pulse occurring from both sensors at the same time.

Both the camshaft and crankshaft sensor outputs are connected to an input capture pin. When an input capture occurs on both of these channels simultaneously, both will have the same input capture value. This gives an absolute timing reference for determining fuel injection and spark timing. In the above example, the crankshaft is also connected to the pulse accumulator input (PAI) pin.

The PAI pin can be used to count the number of teeth after the timing reference. If the pulse accumulator counter (PACNT) is loaded with $F7 (247 decimal), the counter will roll over after the ninth tooth is counted. If the crankshaft timing gear has 36 teeth, the ninth tooth represents TDC of the next piston on a four-cylinder engine. Eighteen teeth represents the next TDC, and 27 teeth the last piston in the firing order.




A – AUTO SPARK COIL
B – FUEL VALVE
C – SIGNAL CONDITIONING AND BUFFERING ELECTRONICS
D – HIGH-CURRENT DRIVER ELECTRONICS

**Figure 10-2. Engine Control Example**

## 10.3 Software Examples

This section contains software that shows how to initialize parts of the GPT. The software is interrupt driven. It also shows how to set up the interrupt vectors and adjust the priority of the different channels in the GPT. Portions of code dealing with calulating control values for the motor speed were lexcluded.

```
Motorola Assembler                          (3.0    ) Mon Apr 15 14:04:11 1991


abs.   LC   obj. code   source line
----   ---- ---------   -----------
1                      |* Example program/programs to show operation of GPT.
2                      |* This code written for use on a MC68331 BCC for demonstration
3                      |* purposes,but the same basic routines could be used on other
4                      |* Motorola modular microcontroller family MCUs.
5                      |
6                      |*
7                      |*
8                      |*
9                      |*
10                     |*
11                     |*
12                     |*
13                     |*       GENERAL REGISTER ADDRESS EQUATES
14                     |*
15                     |
16   00FF F900         |BASE        EQU     $FFF900             BASE ADDRESS OF GPT ON THE MC68331
17                     |
18   00FF F900         |GMCR        BASE                        GPT MCR REGISTER ADDRESS
19   00FF F904         |GICR        EQU     BASE+$04            INTERRUPT CONTROL REGISTER
20   00FF F906         |PDDR_PDR    EQU     BASE+$06            PORT DATA DIRECTION REG/PORT DATA REG
21   00FF F907         |PDR         EQU     BASE+$07            PORT DATA REG IF ADDR BYTE BOUNDARY
22   00FF F908         |OC1MD       EQU     BASE+$08            OUTPUT CAPTURE MASK AND DATA ACTION REG
23   00FF F90A         |TCNT        EQU     BASE+$0A            TIMER COUNTER REGISTER
24   00FF F90C         |PACTL_PACNT EQU     BASE+$0C            PULSE ACCUM CONTROL AND COUNTER REG
25   00FF F90E         |TIC1        EQU     BASE+$0E            TIMER INPUT CAPTURE 1 REGISTER
26   00FF F910         |TIC2        EQU     BASE+$10            TIMER INPUT CAPTURE 2 REGISTER
27   00FF F912         |TIC3        EQU     BASE+$12            TIMER INPUT CAPTURE 3 REGISTER
28   00FF F914         |TOC1        EQU     BASE+$14            TIMER OUTPUT COMPARE 1 REGISTER
29   00FF F916         |TOC2        EQU     BASE+$16            TIMER OUTPUT COMPARE 2 REGISTER
30   00FF F918         |TOC3        EQU     BASE+$18            TIMER OUTPUT COMPARE 3 REGISTER
31   00FF F91A         |TOC         EQU     BASE+$1A            TIMER OUTPUT COMPARE 4 REGISTER
32   00FF F91C         |TI4O5       EQU     BASE+$1C            TIMER IC 4 OC 5 REG
33   00FF F91E         |TCTL1_2     EQU     BASE+$1E            TIMER CONTROL REGISTER 1 AND 2
34   00FF F920         |TMSK1_2     EQU     BASE+$20            TIMER INTERRUPT MASK REGISTER 1 AND 2
35   00FF F922         |TFLG1_2     EQU     BASE+$22            TIMER INTERRUPT FLAG REGISTER 1 AND 2
36   00FF F923         |TFLG2       EQU     BASE+$23            BYTE ADDRESS OF TFLG2
37   00FF F924         |CFORC_PWMC  EQU     BASE+$24            COMPARE FORCE AND PWM CONTROL REGISTER
38   00FF F925         |PWMC        EQU     BASE+$25            JUST THE PWMC REGISTER
39   00FF F926         |PWMA_B      EQU     BASE+$26            PWM REGISTER A AND B
40   00FF F928         |PWMCNT      EQU     BASE+$28            PWM COUNT REGISTER
41   00FF F92A         |PWMBUF_A_B  EQU     BASE+$2A            PWM BUFFER REGISTER A AND B (READ ONLY)
42   00FF F92C         |PRESCL      EQU     BASE+$2C            GPT PRESCALER (LOWER 9 BITS)
43                     |
44                     |
45                     |
46                     |*************** DEFAULT EQUATES **********************
47                     |
48                     |*       These equates are for setup of the basic operation of the
49                     |*       331 such as interrupt addresses and enabling certain interrupts.
50                     |*
51   0000 0088         |GMCR_ARB    EQU     $0088               GPT SUPER MODE, IARB = 8  ARBITRATION
52   0000 0540         |GICR_D      EQU     $0540               GPT PAB=0, IR=5, IVBA=$4X(VECTOR 64 DEC)
53   0000 0100         |VEC_BASE    EQU     $0100               VECTOR TABLE OFFSET
54   0000 0004         |V_ADDR_IC1  EQU     $01*04              IC1 VECTOR ADDRESS OFFSET
55   0000 0010         |V_ADDR_OC1  EQU     $04*04              0C1 VECTOR ADDRESS OFFSET
56   0000 0024         |V_ADDR_TOF  EQU     $09*04              TOF VECTOR ADDRESS OFFSET
57   0000 0080         |V_ADDR_TRP  EQU     32*04               TRAP #0 (VECTOR #32)VECTOR
58                     |*                                     ADDRESS OFFSET
59                     |
```

```
60   0000 0006       |TMSK_ID     EQU      $0006              DISABLE INT, SET TCNT PRESCALER TO /256
61   0000 0980       |TMSK_IE     EQU      $0980              OC1,TOI AND IC1 INTERRUPTS
62   0000 2400       |CPU_IM      EQU      $2400              SUPERVISOR MODE AND LEVEL 5 AND ABOVE
63                   |
64                   |*********************************************************************
65                   |*       The stack and the VBR will not be initialized because the
66                   |*       BCC init software takes care of this.
67                   |*********************************************************************
68                   |
69                   |
70                   |***************IC1_HAND EQUATES********************
71                   |*
72   0000 0040       |PWMC_DATA   EQU      $0040              SYS CLK /32 INPUT TO PWMCNT(524.29 KHZ)
73                   |*          FAST MODE 2.05 KHZ
74                   |
75                   |*************** OC1_HAND EQUATES ********************
76                   |
77   0000 2000       |DELAY1      EQU      $2000              THIS GIVES A xxx MSEC COUNT
78   0000 E351       |DELAY2      EQU      $E351              THIS GIVES A xxx MSEC COUNT
79                   |
80                   |****************MISC EQUATES**************************
81   0003 C000       |RPM_CONST   EQU      245760             CONVERSION CONSTANT FOR RPM
82   0000 03E8       |RPM_M       EQU      1000               SET MOTOR VELOCITY HERE FOR
83                   |*                                      TESTING
84                   |
85                   |
86   0000 3000       |RAM_ST      EQU      $3000              BEGINNING OF FREE RAM
87                   |
88   3000            |           ORG      RAM_ST
89                   |
90                   |
91                   |****************DATA AREA*****************************
92                   |
93                   |*       VARIABLES
94                   |
95   3000            |RPM_S:      DS.W     1                  SETPOINT PARAMETER STORAGE LOCATION
96   3002            |PREV_IC1:   DS.W     1                  PREVIOUS INPUT CAPTURE ONE VALUE
97   3004            |FIRST_FLG:  DS.W     1                  SET THE FIRST TIME IC1_HAND IS EXECUTED
98                   |
99                   |
100                  |
101                  |*       PROGRAM STARTS HERE
102                  |
103                  |*************************************
104                  |*       This portion of the code sets up vector ad
105                  |*       for the various interrupt handlers:
106                  |*       IC1_HAND, OC1_HAND, TOF_HAND
107                  |*************************************
108                  |
109  3006 203C 0000 |START:     MOVE.L   #VEC_BASE,D0       GET VECTOR BASE ADDRESS
     300A 0100       |
110  300C 7204       |           MOVE.L   #V_ADDR_IC1,D1
111  300E 8280       |           OR.L     D0,D1              CALCULATE INTERRUPT HANDLER ADDRESSES
112  3010 21BC 0000 |           MOVE.L   #IC1_HAND,(D1)     COMBINE THE TWO
                     |*                                     LOAD ADDRESS OF IC1 HANDLER
     3014 30CC 1190  |
113                  |*
114  3018 7210       |           MOVE.L   #V_ADDR_OC1,D1
115  301A 8280       |           OR.L     D0,D1              COMBINE THE TWO
116  301C 21BC 0000 |           MOVE.L   #OC1_HAND,(D1)     LOAD ADDRESS OF OC1 HANDLER
     3020 30A6 1190  |
117                  |*
118  3024 7224       |           MOVE.L   #V_ADDR_TOF,D1
119  3026 8280       |           OR.L     D0,D1
120  3028 21BC 0000 |           MOVE.L   #TOF_HAND,(D1)     LOAD ADDRESS OF TOF HANDLER
     302C 312A 1190  |
121                  |*
122  3030 21FC 0000 |           MOVE.L   #SPD_SET,V_ADDR_TRP LOAD ADDRESS OF TRAP #0 HANDLER
     3034 3128 0080  |
123                  |
```

```
124                 |*******************************************************
125                 |*       This section disables interrupts and disconnects output
126                 |*       compares while we are setting up. Then the operating
127                 |*       conditions for OC1, OC2, the prescaler and the PWM.
128                 |*       After this interrupts of interest are enabled.
129                 |*******************************************************
130                 |
131  3038 33FC 0006 |            MOVE.W   #TMSK_ID,TMSK1_2    DISABLE INTERRUPTS AND SET PRESCALER
     303C 00FF F920 |
132  3040 13FC 0000 |            MOVE.B   #$00,TCTL1_2        DISCONNECT OUTPUT COMPARES
     3044 00FF F91E |
133  3048 33FC 1010 |            MOVE.W   #$1010,OC1MD        OC2 AFFECTED BY OC1 SETS A 1
     304C 00FF F908 |
134  3050 33FC 0202 |            MOVE.W   #$0202,TCTL1_2      CONNECT OC2 AND OC1 FALLING EDGE
     3054 00FF F91E |
135  3058 13FC 0040 |            MOVE.B   #PWMC_DATA,PWMC     SETUP INPUT TO PWMCNT /32 AND
     305C 00FF F925 |
136                 |*                                       FAST MODE  2.05 KHZ
137  3060 31FC 0000 |            MOVE.W   #0,FIRST_FLG        INIT TO ZERO, FLAG TO SHOW
     3064 3004      |
138                 |*                                       FIRST TIME THRU IC1_HAND
139                 |*
140                 |*       Status flags are first read and then written to clear the flag.
141                 |*
142  3066 4A79 00FF |            TST      TFLG1_2
     306A F922      |
143  306C 33FC 0000 |            MOVE.W   #$0000,TFLG1_2      CLEAR ANY PENDING INTERRUPTS
     3070 00FF F922 |
144  3074 33FC 0980 |            MOVE.W   #TMSK_IE,TMSK1_2    ENABLE ANY INTERRUPTS WE WANT.
     3078 00FF F920 |
145                 |*
146                 |*******************************************************************
147                 |*       Here the Interrupt Arbitration ID is set.  The PAB and IRL
148                 |*       are also set up.  The Processor Status Register is written
149                 |*       so the CPU will respond to level 5 interrupts.
150                 |*******************************************************************
151                 |*
152  307C 33FC 0088 |            MOVE.W   #GMCR_ARB,GMCR      SETUP MCR REGISTER
     3080 00FF F900 |
153                 |*                                       THIS SETS THE IARB ID
154  3084 33FC 0540 |            MOVE.W   #GICR_D,GICR        SETUP VECTOR ADDRESS AND PRIORITY
     3088 00FF F904 |
155  308C 46FC 2400 |            MOVE.W   #CPU_IM,SR          SET CPU INTERRUPT MASK LEVEL
156                 |
157                 |*******************************************************************
158                 |*       Set up motor RPM. Now RPM_M is defined in MISC EQUATES.
159                 |*       This value after being converted to counts is passed to the Interrupt
160                 |*       Handler through the variable RPM_S. RPM_CONST is defined in IC1_HAND
161                 |*       EQUATES.  In the present example, the Trap instruction starts the moto
162                 |*
163                 |*******************************************************************
164                 |*
165  3090 223C 0003 |            MOVE.L   #RPM_CONST,D1
     3094 C000      |
166  3096 4C7C 1001 |            DIVU.L   #RPM_M,D1           DO CONVERSION
     309A 0000 03E8 |
167  309E 31C1 3000 |            MOVE.W   D1,RPM_S            PASS THE VALUE TO THE INT HANDLER
168                 |*                                       THROUGH VARIABLE RPM_S
169  30A2 4E40      |            TRAP     #0                  PRETEND THAT WE GOT AN INTERRUPT
170                 *
171  30A4 60FE      |            BRA      *                   GO AROUND UNTIL INTERRUPT.
172                 |*                                       THIS IS WHERE MAIN LOOP WOULD
173                 |*                                       BE IF WE HAD ONE.
174                 |
```

```
175                 |*********************************************************************
176                 |*       OC1_HAND
177                 |*
178                 |*       This is the interrupt handler entered when OC1 generates an
179                 |*       interrupt.  Its address is loaded into the vector table at
180                 |*       location $0100.  The only register modified is 0, and it is
181                 |*       saved on the stack on entry to the routine and restored before
182                 |*       returning from the interrupt.
183                 |*********************************************************************
184                 |
185                 |
186  30A6           |OC1_HAND:
187  30A6 3F00      |            MOVE.W   D0,-(SP)            SAVE D0
188  30A8 3039 00FF |            MOVE.W   TOC1,D0
     30AC F914      |
189  30AE 0640 2000 |            ADD.W    #DELAY1,D0          CALCULATE OC2 HIGH TIME
190  30B2 33C0 00FF |            MOVE.W   D0,TOC2
     30B6 F916      |
191  30B8 0679 E351 |            ADD.W    #DELAY2,TOC1        CALCULATE HOW LONG BEFORE NEXT OC1
     30BC 00FF F914 |
192  30C0 08B9 0003 |            BCLR     #3,TFLG1_2          CLEAR INTERRUPT STATUS FLAG
     30C4 00FF F922 |
193  30C8 301F      |            MOVE.W   (SP)+,D0            RESTORE D0
194  30CA 4E73      |            RTE                          GO BACK TO WHERE WE WERE
195                 |
196                 |
197                 |
198                 |*********************************************************************
199                 |*       IC1_HAND
200                 |
201                 |*       This interrupt handler is entered when IC1 generates an interrupt.
202                 |*       Its address is loaded into the vector table at location $0104.
203                 |*       In this example IC1 is used to capture the output of an optical
204                 |*       tachometer that provides velocity information from a motor. The
205                 |*       motor is driven by the PWMA signal.
206                 |*
207                 |*       This handler reads the input capture register and subtracts it from
208                 |*       the previous value.  If the result does not equal the first value,
209                 |*       the result of the calculation represents the quantity 245760/RMP_motor.
210                 |*
211                 |*       The tach provides 16 pulses per revolution of the motor shaft, or 16
212                 |*       input captures per revolution. Because the input to the TCNT is the
213                 |*       system clock divided by 256, the TCNT clock rate is 65536 Hz. Therefore,
214                 |*       the time between tach pulses is 1/65536 Hz, or 15.26 microseconds times
215                 |*       the count value between two successive input captures. This is what gives
216                 |*       the quantity 245760/RPM_motor.
217                 |*
218                 |*********************************************************************
219                 |
220  30CC 48E7 C000 |IC1_HAND:  MOVEM.L D0/D1,-(SP)           SAVE REGISTERS USED ON THE STACK
221  30D0 08F8 0007 |           BSET    #7,FIRST_FLG          CHECK IF FIRST TIME THRU
     30D4 3004      |
222  30D6 6700 003A |           BEQ     EXIT1                 IF IT IS EXIT
223  30DA 08B9 0000 |           BCLR    #0,TFLG1_2            CLEAR IC1 INTERRUPT
     30DE 00FF F922 |
224                 |
225  30E2 4280      |           CLR.L   D0
226  30E4 3039 00FF |           MOVE.W  TIC1,D0               READ INPUT CAPTURE REGISTER
     30E8 F90E      |
227  30EA 3238 3002 |           MOVE.W  PREV_IC1,D1           GET THE PREVIOUS VALUE OF IC1
228  30EE 31C0 3002 |           MOVE.W  D0,PREV_IC1           MAKE THE CURRENT ONE THE NEXT PREVIOUS
229  30F2 9041      |           SUB.W   D1,D0                 NUMBER OF COUNTS SINCE LAST TIME
230                 |*                                       D0 EQUALS THE 245,760/RPM_Motor
231  30F4 4281      |           CLR.L   D1
232  30F6 3238 3000 |           MOVE.W  RPM_S,D1              GET THE SPEED SETPOINT IN COUNTS
233  30FA B240      |           CMP     D0,D1                 IS SPEED THE SAME AS SETPOINT
234  30FC 6700 0024 |           BEQ     EXIT2                 SPEED IS THE SAME
235                 |*
236                 |       *   Code to calulate the ratio of the contents of D0
237                 |       *   (245,760/RPM_Motor) and D1 (245,760/RPM_Setpoint)
238                 |       *   is calulated here.  This gives an error term that
239                 |       *   represents the percentage that the motor is from
240                 |       *   its setpoint velocity.
241                 |*
242  3100 4280      |           CLR.L   D0
243  3102 1039 00FF |           MOVE.B  PWMBUF_A_B,D0         GET CURRENT VALUE OF PWMA
```

```
    3106 F92A       |
244                 |*
245                 |       *   Code to apply correction factor to PWMA
246                 |       *   would be inserted here.
247                 |       *
248                 |*
249 3108 13C1 00FF  |           MOVE.B  D1,PWMA_B          LOAD IT
    310C F926       |
250 310E 6000 0012  |           BRA     EXIT2              WE'RE DONE
251                 |
252 3112 31F9 00FF  |EXIT1:     MOVE.W  TIC1,PREV_IC1      MAKE THIS THE PREVIOUS VALUE
    3116 F90E 3002  |
253                 |*                                     FOR NEXT TIME
254 311A 13FC 00FF  |           MOVE.B  #$FF,PWMA_B        LOAD FULL SPEED TO START THE MOTOR
    311E 00FF F926  |
255 3122 4CDF 0003  |EXIT2:     MOVEM.L (SP)+,D1/D0        RESTORE REGISTERS
256 3126 4E73       |           RTE
257                 |
258                 |*************************************************************************
259                 |
260 3128 60A2       |SPD_SET:   BRA     IC1_HAND           THE TRAP CAUSES AN EXCEPTION TO HERE
261                 |*                                     AND THEN WE JUMP TO THE IC1 HANDLER
262                 |*                                     AND LET THE HANDLER RETURN FROM THE
263                 |*                                     EXCEPTION THROUGH ITS RTE INSTRUCTION
264                 |
265                 |
266                 |*************************************************************************
267                 |*       TOF_HAND
268                 |*
269                 |*       Here interrupts generated by a TCNT overflow would be
270                 |*       taken care of.
271                 |*************************************************************************
272                 |
273 312A 08B9 0007  |TOF_HAND:  BCLR    #7,TFLG2           FOR THIS EXAMPLE WE JUST CLEAR
    312E 00FF F923  |
274                 |*                                     THE FLAG
275 3132 4E73       |           RTE
276                 |
277                 |
278                 |           END
```

10

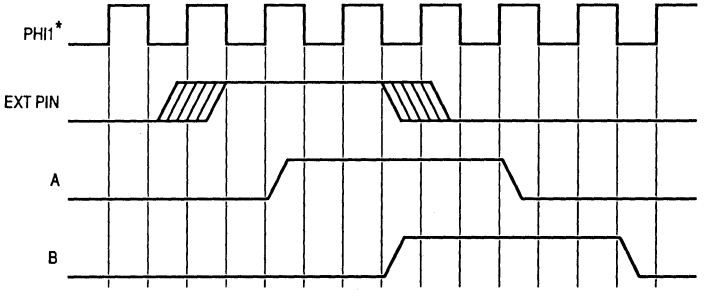
# SECTION 11
# ELECTRICAL CHARACTERISTICS

This section contains electrical characteristics and associated timing information for the General-Purpose Timer.

## 11.1 AC Characteristics

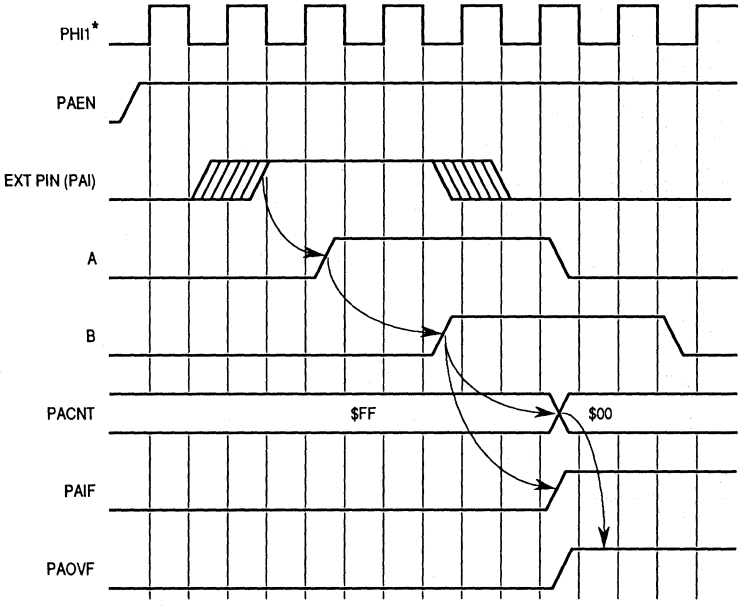
| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Operating Frequency | Fclock | 0 | | 16.77 | MHz |
| PCLK Frequency | Fpclk | 0 | | 1/4 Fclock | MHz |
| Pulse Width Input Capture | PWtim | 2/Fclock | | | |
| PWM Resolution | | 2/Fclock | | | |
| IC/OC Resolution | | 4/Fclock | | | |
| PCLK Width (PWM) | | 4/Fclock | | | |
| PCLK Width (IC/OC) | | 4/Fclock | | | |
| PAI Pulse Width | | 2/Fclock | | | |

## 11.2 Timing Specifications




NOTES:
A = Input signal after the synchronizer
B = "A" after the digital filter
*PHI1 is the same frequency as system clock; however, it does not have the same timing.

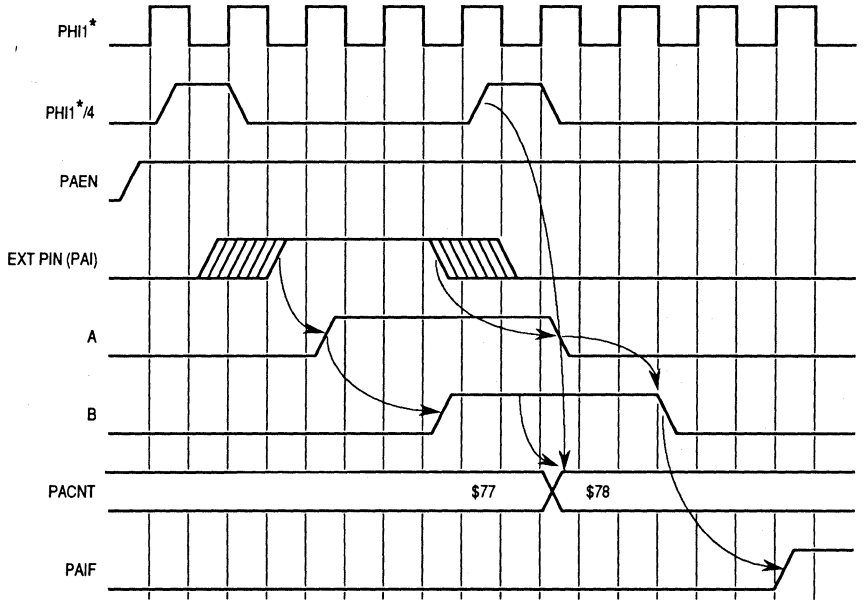
**Figure 11-1. Input Signal Conditioner Timing**




NOTES:
A — PAI signal after the synchronizer
B — "A" after the digital filter
*PHI1 is the same frequency as system clock; however, it does not have the same timing.
The external leading edge causes the pulse accumulator to increment and the PAIF flag to be set.
The counter transition from $FF to $00 causes the PAOVF flag to be set.

**Figure 11-2. Pulse Accumulator — Event Counting Mode (Leading Edge)**

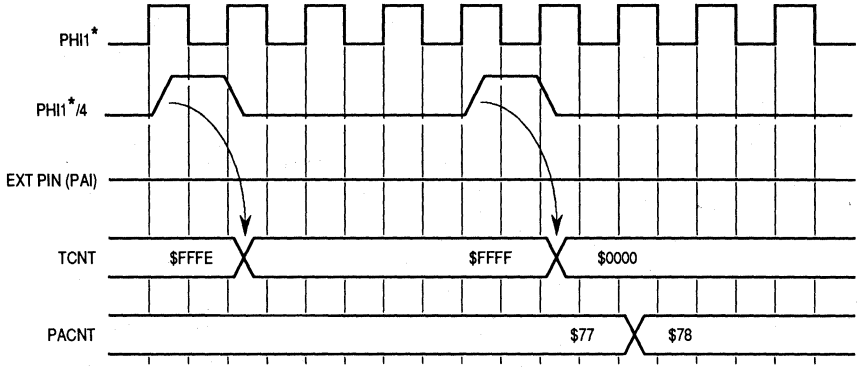
11




NOTES:
A — PAI signal after the synchronizer
B —"A" after digital the filter
PHI1*/4 clocks PACNT when GT–PAIF is asserted.
PAIF is asserted when PAI is negated.
*PHI1 has the same frequency as the system clock; however, it does not have the same timing.

**Figure 11-3. Pulse Accumulator — Gated Mode (Count while Pin High)**

11

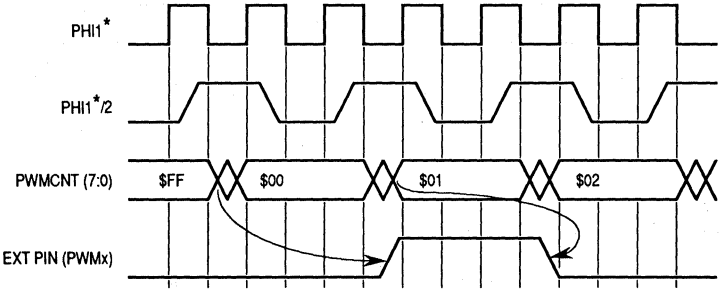



NOTES:

TCNT counts as a result of PHI1*/4; PACNT counts when TCNT overflows from $FFFF to $0000 and the conditioned PAI signal is asserted.

* PHI1 is the same frequency as the system clock; however, it does not have the same timing.

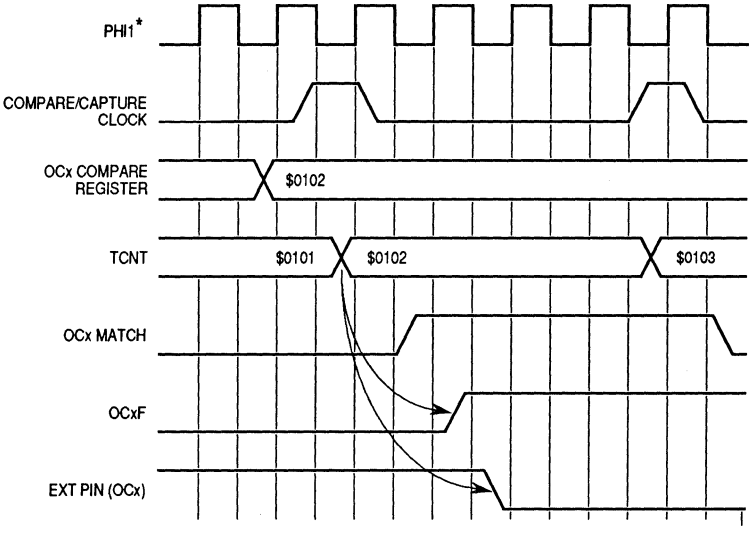
**Figure 11-4. Pulse Accumulator — Using TOF as Gated Mode Clock**




NOTES:

*PHI1 is the same frequency as the system clock; however, it does not have the same timing.
When the counter rolls over from $FF to $00, the PWM pin is set to a logic level one.
When the counter equals the PWM register, the PWM pin is cleared to a logic level zero.

**Figure 11-5. PWMx (PWMx Register = 01, Fast Mode)**

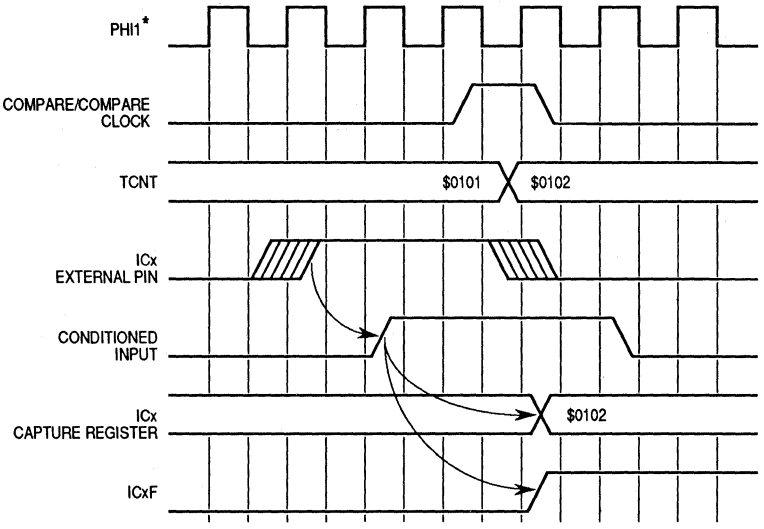



NOTES:

When the TCNT matches the OCx compare register, the OCxF flag is set followed by the OCx pin changing state.

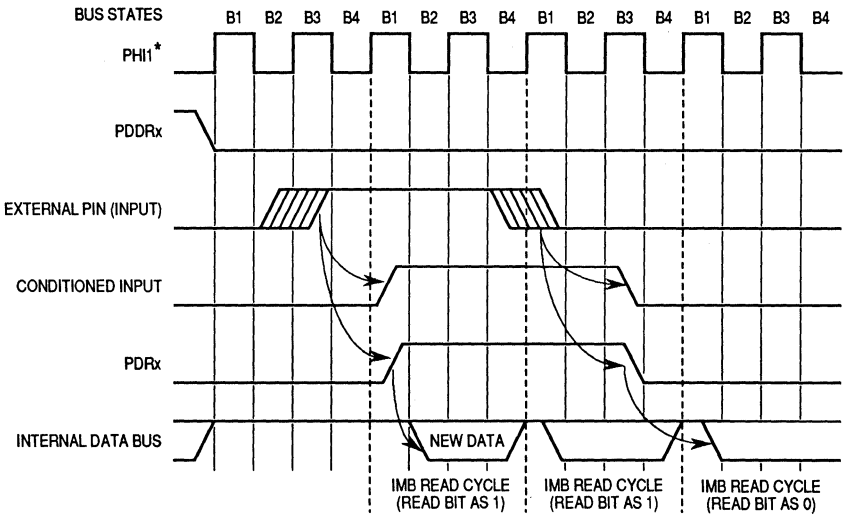
* PHI1 is the same frequency as the system clock; however, it does not have the same timing.

**Figure 11-6. Output Compare (Toggle Pin State)**



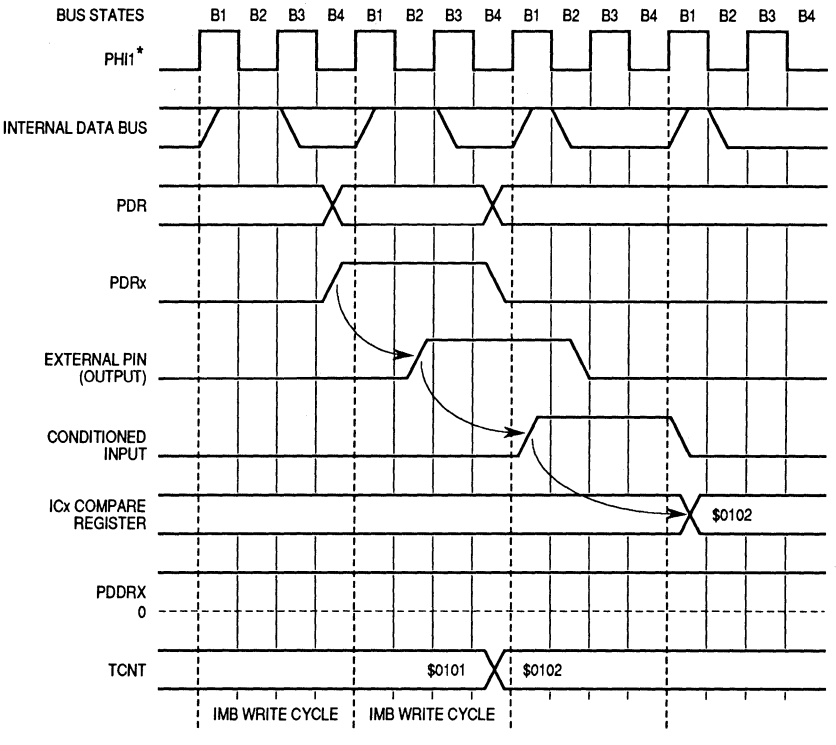

NOTES:
The conditioned input signal causes the current value of the TCNT to be latched by the ICx capture register. The ICxF flag is set at the same time.
*PHI1 is the same frequency as the system clock; however, it does not have the same timing.

**Figure 11-7. Input Capture (Capture on Rising Edge)**




* PHI1 is the same frequency as the system clock; however, it does not have the same timing.

**Figure 11-8. General-Purpose Input**



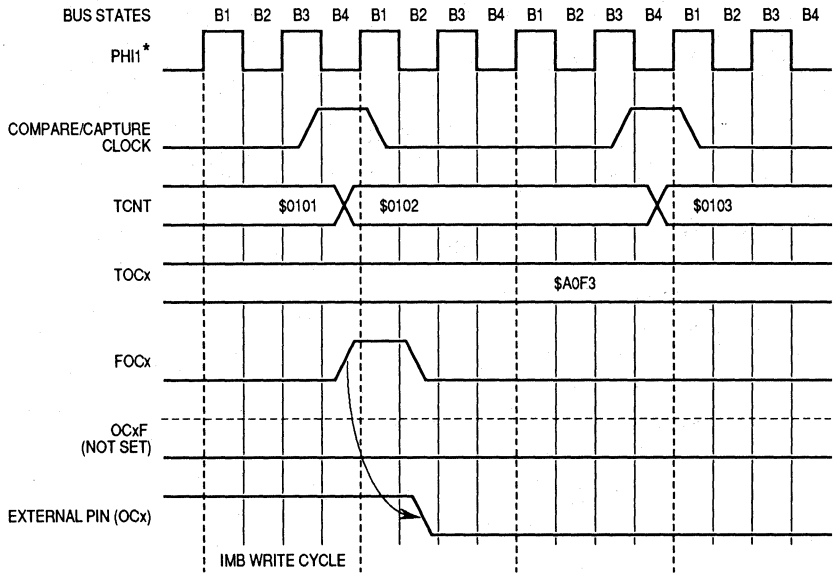

NOTES:

*PHI1 is the same frequency as the system clock; however, it does not have the same timing.

When the bit value is driven on the pin, the input circuit sees the signal. After it is conditioned it causes the contents of the TCNT to be latched into the ICx compare register.

**Figure 11-9. General-Purpose Output (Causes Input Capture)**

11




*PHI1 is the same frequency as the system clock; however, it does not have the same timing.

**Figure 11-10. Force Compare (CLEAR)**

# APPENDIX A
# MEMORY MAP AND REGISTERS

## A.1 GPT Register Map

| | | WORD | |
|---|---|---|---|
| | ADDRESS | 15 BYTE n 8 | 7 BYTE n + 1 0 |
| S | $YFF900 | MCR | |
| S | $YFF902 | RESERVED | |
| S | $YFF904 | ICR | |
| U | $YFFE06 | PDDR | PDR |
| U | $YFF908 | OC1M | OC1D |
| U | $YFF90A | TCNT | |
| U | $YFF90C | PACTL | PACNT |
| U | $YFF90E | TIC1 | |
| U | $YFF910 | TIC2 | |
| U | $YFF912 | TIC3 | |
| U | $YFF914 | TOC1 | |
| U | $YFF916 | TOC2 | |
| U | $YFF918 | TOC3 | |
| U | $YFF91A | TOC4 | |
| U | $YFF91C | TI4O5 | |
| U | $YFF91E | TCTL1 | TCTL2 |
| U | $YFF920 | TMSK1 | TMSK2 |
| U | $YFF922 | TFLG1 | TFLG2 |
| U | $YFF924 | CFORC/PWMC | |
| U | $YFF926 | PWMA REGISTER | PWMB REGISTER |
| U | $YFF928 | PWM COUNTER | |
| U | $YFF92A | PWMA BUFFER REGISTER | PWMB BUFFER REGISTER |
| U | $YFF92C | PRESCALER (Lower 9 bits) | |
| | $YFF92E<br>$YFF93F | RESERVED | |

S = Supervisor-accessible only

Y = m111 where m is the state of the modmap bit in the module configuration register of the system integration module (Y = $7 or $F)

A

**MCR** — GPT Module Configuration Register **$YFF900**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STOP | FRZ1 | FRZ0 | STOPP | INCP | 0 | 0 | 0 | SUPV | 0 | 0 | 0 | IARB3 | IARB2 | IARB1 | IARB0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

STOP — Stop Clocks

0 = Internal clocks are not shut down.
1 = Internal clocks are shut down.

FRZ1, FRZ0 — FREEZE Response

0 = Ignore FREEZE.
1 = Freeze the current state of the GPT when FREEZE is asserted.

STOPP— Stop Prescaler

0 = Normal operation.
1 = Stop the prescaler and pulse accumulator from incrementing; ignore changes to input pins.

INCP — Increment Prescaler

0 = Has no meaning.
1 = If STOPP is asserted, increment the prescaler once and clock the input synchronizers once.

SUPV — Supervisor/Unrestricted Data Space

0 = Registers with access controlled by the SUPV bit are unrestricted (FC2 is a don't care).
1 = Registers with access controlled by the SUPV bit are restricted to supervisor access.

IARB3–0 — Interrupt Arbitration ID

The interrupt structure of the IMB supports a total of 15 internal interrupt sources. External interrupts are grouped as one of the internal sources. Each source supports seven interrupts levels. A level 7 interrupt has the highest priority. Each of the 15 internal sources is assigned a unique ID, the Interrupt Arbitration ID, which is used to determine which interrupt will be serviced if two or more interrupts of the same priority occur at the same time.

A

These sources must arbitrate for the interrupt during the IACK cycle. The module with the higher ID wins the arbitration. System software must set this field to $F–$1, $F being the highest priority. This field is initialized to zero during reset, which disables arbitration and causes interrupts generated by the module to be treated as spurious.

**MTR** — GPT Module Test Register **$YFF902**

No module test register is implemented on the GPT. However, this address location is reserved for future needs.

**ICR** — GPT Interrupt Configuration Register **$YFF904**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAB | | | | 0 | IRL | | | VBA | | | | 0 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

PAB — Priority Adjust Bits

The GPT has 11 sources capable of generating an interrupt. These bits specify the single interrupt source which is to be advanced to the highest priority level. All of the other interrupt sources maintain their relative priority.

| Interrupt Source | | Priority Level | Vector Address |
|---|---|---|---|
| **Name** | **Function** | | |
| — | Adjusted Channel | 0 (Highest) | $X0 |
| IC1 | Input Capture 1 | 1 | $X1 |
| IC2 | Input Capture 2 | 2 | $X2 |
| IC3 | Input Capture 3 | 3 | $X3 |
| OC1 | Output Compare 1 | 4 | $X4 |
| OC2 | Output Compare 2 | 5 | $X5 |
| OC3 | Output Compare 3 | 6 | $X6 |
| OC4 | Output Compare 4 | 7 | $X7 |
| IC4/OC5 | Input Capture 4/Output Compare 5 | 8 | $X8 |
| TOF | Timer Overflow Flag | 9 | $X9 |
| PAOVF | Pulse Accumulator Overflow Flag | 10 | $XA |
| PAIF | Pulse Accumulator Input Flag | 11 (Lowest) | $XB |

X = 4-bit Vector Base Address (VBA)

X = 4-bit Vector Base Address (VBA)

IRL — Interrupt Request Level

These bits specify the priority level of the GPT interrupts. The GPT can have any of eight priority levels, level 7 being the highest and level 0 disabling interrupts. The interrupt request level specifies the priority level presented to the CPU. The interrupt request level is initialized to zero during reset.

All GPT internal interrupts are prioritized as shown in the above table. The interrupt with the highest priority generates the interrupt level to the IMB specified by this field.

A

Vector Base Address —

This field specifies the most significant nibble of all the vector numbers that can be generated by the different interrupt sources of the GPT module. This value concatenated with the vector address shown in the above table is the module interrupt vector.

**PDDR/PDR** — Parallel Data Direction Register/Parallel Data Register **$YFF906**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DDRI4O5 | DDRO4 | DDRO3 | DDRO2 | DDRO1 | DDRI3 | DDRI2 | DDRI1 | IC4/OC5 | OC4 | OC3 | OC2 | OC1 | IC3 | IC2 | IC1 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DDRx — Data Direction for Input Capture/Output Compare Pins

0 = Input only

1 = Output

IC4/OC5, OC4–1, IC3–1 — Parallel Data Port

**OC1M/OC1D** — OC1 Action Mask Register/OC1 Action Data Register **$YFF908**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC1M7 | OC1M6 | OC1M5 | OC1M4 | OC1M3 | 0 | 0 | 0 | OC1D7 | OC1D6 | OC1D5 | OC1D4 | OC1D3 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

OC1Mx — OC1 Mask Bits

0 = Corresponding bit in the parallel data port is not affected by OC1 compare.

1 = Corresponding bit in the parallel data port is affected by OC1 compare.

OC1Dx — OC1 Data Bits

0 = If OC1Mx is set, store 0 on the corresponding parallel data port bit on OC1 match.

1 = If OC1Mx is set, store 1 on the corresponding parallel data port bit on OC1 match.

**TCNT** — Timer Counter Register **$YFF90A**

TCNT is the 16-bit free-running counter associated with the input capture, output compare, and pulse accumulator functions of the GPT module.

A

**PACTL/PACNT** — Pulse Accumulator Control Register/Pulse Accumulator Counter **$YFF90C**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAIS | PAEN | PAMOD | PEDGE | PCLKS | I4/O5 | PACLK1 | PACLK0 | PULSE ACCUMULATOR COUNTER | | | | | | | |
| RESET: | | | | | | | | | | | | | | | |
| U | 0 | 0 | 0 | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

PAIS — PAI Pin State (Read-Only)

PAEN — Pulse Accumulator System Enable
- 0 = Pulse accumulator disabled
- 1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode
- 0 = External event counting
- 1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control

This bit has different meanings based on the state of the PAMOD bit.

| PAMOD | PEDGE | Action on Clock |
|---|---|---|
| 0 | 0 | PAI falling edge increments counter. |
| 0 | 1 | PAI rising edge increments counter. |
| 1 | 0 | A zero on PAI inhibits counting (gated mode). |
| 1 | 1 | A one on PAI inhibits counting (gated mode). |

PCLKS — PCLK Pin State (Read-Only)

I4/O5 — Input Capture 4/Output Compare 5
- 0 = Output compare 5 function enabled
- 1 = Input capture 4 function enabled

PACLK1–0 — Pulse Accumulator Clock Select (Gated Mode)

| PACLK1 | PACLK0 | Pulse Accumulator Clock Selected |
|---|---|---|
| 0 | 0 | System Clock Divided by 512 |
| 0 | 1 | Same Clock Used to Increment TCNT |
| 1 | 0 | TOF Flag from TCNT |
| 1 | 1 | External Clock, PCLK |

Pulse Accumulator Counter

This is an 8-bit read/write counter used for external event counting or gated time accumulation.

A

**TIC1–TIC3** — Input Capture Registers 1–3 **$YFF90E, $YFF910, $YFF912**

The input capture registers are 16-bit read-only registers which are used to latch the value of TCNT when a specified transition is detected on the corresponding input capture pin. They are reset to $FFFF.

**TOC1–TOC4** — Output Compare Registers 1–4 **$YFF914, $YFF916, $YFF918, $YFF91A**

The output compare registers are 16-bit read/write registers which can be used as output waveform controls or as elapsed time indicators. For output compare functions, they are written to a desired match value and compared against TCNT to control specified pin actions. They are reset to $FFFF.

**TI4O5** — Input Capture 4/Output Compare 5 Register **$YFF91C**

This register serves either as input capture register 4 or output compare register 5, depending on the function chosen for the I4/O5 pin.

**TCTL1/TCTL2** — Timer Control Registers 1–2 **$YFF91E**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OM5 | OL5 | OM4 | OL4 | OM3 | OL3 | OM2 | OL2 | EDGE4B | EDGE4A | EDGE3B | EDGE3A | EDGE2B | EDGE2A | EDGE1B | EDGE1A |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

OMx — Output Compare Mode Bits

OLx — Output Compare Level Bits

These bits are encoded to specify the output action to be taken as a result of a successful OCx compare.

| OMx–OLx | Action Taken on Successful Compare |
|---|---|
| 00 | Timer Disconnected from Output Logic |
| 01 | Toggle OCx Output Line |
| 10 | Clear OCx Output LIne to 0 |
| 11 | Set OCx Output Line to 1 |

EDGExA, EDGExB — Input Capture Edge Control Bits

These bits are encoded to configure the input sensing logic for the corresponding input capture.

| EDGExB–A | Configuration |
|---|---|
| 00 | Capture Disabled |
| 01 | Capture on Rising Edge Only |
| 10 | Capture on Falling Edge Only |
| 11 | Capture on Any (Rising or Falling) Edge |

**TMSK1/TMSK2** — Timer Interrupt Mask Registers 1–2 **$YFF920**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I4O5I | OC4I | OC3I | OC2I | OC1I | IC3I | IC2I | IC1I | TOI | 0 | PAOVI | PAII | CPROUT | CPR2 | CPR1 | CPR0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

OCxI — Output Compare x Interrupt Enable
- 0 = OCx interrupts disabled
- 1 = OCx Interrupts requested when OCxF flag is set

ICxI — Input Capture x Interrupt Enable
- 0 = ICx interrupts disabled
- 1 = ICx interrupts requested when ICxF flag is set

I4O5I — Input Capture 4/Output Compare 5 Interrupt Enable
- 0 = IC4 or OC5 interrupt disabled (depending on I4/O5 pin function)
- 1 = IC4 or OC5 interrupt requested when I4O5I flag is set (depending on I4/O5 pin function)

TOI — Timer Overflow Interrupt Enable
- 0 = Timer overflow interrupts disabled
- 1 = Interrupts requested when TOF flag is set

PAOVI — Pulse Accumulator Overflow Interrupt Enable
- 0 = Pulse accumulator overflow interrupts disabled
- 1 = Interrupts requested when PAOVF flag is set

PAII — Pulse Accumulator Interrupt Enable
- 0 = Pulse accumulator interrupts disabled
- 1 = Interrupts requested when PAIF flag is set

CPROUT — Compare/Capture Unit Clock Output Enable
- 0 = Normal operation for OC1 pin.
- 1 = Output of prescaler mux for compare/capture unit (TCNT clock) driven out on OC1 pin.

CPR2–0 — Timer Prescaler Select Bits

These bits select a prescaler tap or the external clock, PCLK, to be the input to TCNT.

| CPR2–0 | System Clock Divide-By Factor |
|---|---|
| 000 | 4 |
| 001 | 8 |
| 010 | 16 |
| 011 | 32 |
| 100 | 64 |
| 101 | 128 |
| 110 | 256 |
| 111 | PCLK* |

* PCLK is an external clock input pin.

**TFLG1/TFLG2** — Timer Interrupt Flag Registers 1–2 **$YFF922**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I4O5F | OC4F | OC3F | OC2F | OC1F | IC3F | IC2F | IC1F | TOF | 0 | PAOVF | PAIF | 0 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

OCxF — Output Compare x Flag

This flag is set each time TCNT matches the value in output compare register x.

ICxF — Input Capture x Flag

This flag is set each time a selected edge is detected at the input capture x pin.

I4O5F — Input Capture 4/Output Compare 5 Flag

If the I4/O5 pin is configured as input capture 4, this flag is set each time a selected edge is detected at the I4/O5 pin. If the I4/O5 pin is configured as output compare 5, this flag is set each time TCNT matches the value in output compare register 5.

TOF — Timer Overflow Flag

This flag is set each time TCNT advances from a value of $FFFF to $0000.

PAOVF — Pulse Accumulator Overflow Flag

This flag is set each time the pulse accumulator counter advances from a value of $FF to $00.

A

PAIF — Pulse Accumulator Flag

In event counting mode, this bit is set when an active edge is detected on the PAI pin. In gated time accumulation mode, this bit is set at the end of the timed period (when going from the active (counting) state to the inactive (no longer counting) state).

**CFORC/PWMC** — Compare Force Register/PWM Control Register **$YFF924**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FOC5 | FOC4 | FOC3 | FOC2 | FOC1 | 0 | FPWMA | FPWMB | PPROUT | PPR2 | PPR1 | PPR0 | SFA | SFB | F1A | F1B |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

FOCx — Force Output Compare Bits

0 = Has no meaning

1 = Causes pin action programmed for OCx, except that the OCxF flag is not set

FPWMx — Force PWM Value

0 = PWM pin x is used for PWM functions; normal operation.

1 = PWM pin x is used for discrete output. The value of the F1x bit will be driven out on the PWMx pin. This is true for PWMA regardless of the state of the PPROUT bit.

PPROUT — PWM Prescaler Clock Output Enable

0 = Normal PWM operation on PWMA

1 = Output of prescaler mux for PWM counter driven out on the PWMA pin

PPR2–0 — PWM Prescaler

These bits select a prescaler tap or the external clock, PCLK, to be the input to PWMCNT.

| PPR2–0 | System Clock Divide-By Factor |
|---|---|
| 000 | 2 |
| 001 | 4 |
| 010 | 8 |
| 011 | 16 |
| 100 | 32 |
| 101 | 64 |
| 110 | 128 |
| 111 | PCLK* |

* PCLK is an external clock input pin.

SFx — Slow/Fast Bits

0 = The higher speed of PWMx is selected. (The PWMx period is 256 PWMCNT increments long.)

1 = The slower speed of PWMx is selected. (The PWMx period is 32,768 PWMCNT increments long.)

A

F1x — Force Logic One on PWMx

0 = Normal PWMx operation or force a zero on the PWMx pin for discrete output.

1 = Force one on the PWMx pin; a 100% duty cycle PWM or discrete output.

**PWMA/PWMB** — PWM Registers A/B **$YFFF926, $YFFF927**

These registers are associated with the pulse-width value of the PWM output on the corresponding PWM pin. A value of $00 loaded into one of these registers results in a continuously low output on the corresponding pin. A value of $80 results in a 50% duty cycle output to the maximum value of $FF. This maximum value corresponds to an output which is high for 255/256 of the period.

**PWMCNT** — PWM Count Register **$YFF928**

PWMCNT is the 16-bit free-running counter associated with the PWM functions of the GPT module.

**PWMBUFA/B** — PWM Buffer Registers A/B **$YFF92A, $YFF92B**

These read-only registers contain the values associated with the duty cycles of the corresponding PWMs in progress.

**PRESCL** — GPT Prescaler **$YFF92C**

The value of the 9-bit prescaler can be read at this address. The value of the prescaler will be reflected in bits 8–0; whereas, bits 15–9 are unimplemented and will always read as zeros.

# APPENDIX B
# PIN SUMMARY

## B.1 GPT Pin Summary

The following tables summarize the GPT pins, the pin configuration control bits, and the data on the external pins. Reads of pin data (PDR, PAIS and PCLKS) always return the actual state of the pin. In freeze mode, the pin data reflects the state of the pin at the time of the freeze.

Table Legend:

| | |
|---|---|
| OC1 | Pin is used as output compare 1. |
| (1) | Pin data is changed only on OC1 match. |
| OCx | Pin is used as output compare x. |
| (2) | Pin data is changed only on OCx match. |
| OCx/OC1 | Pin is used as output compare x and output compare 1. |
| (3) | Pin is changed on both OC1 and OCx match.<br>Pin data is OC1Dx on OC1 match and the other on OCx match. |
| ICx | Pin is used as input capture. |
| CLK_OUT | Pin is used to output the clock. |
| GPO | Pin is used as general-purpose output. |
| GPI | Pin is used as general-purpose input. |
| PWM | Pin is used for PWM function. |
| PAI | Pin is used as pulse accumulator input. |
| PCLK | Pin is used as external clock input. |

**Table B-1. OC1 Pin**

| Function | OC1M3 | DDRO1 | CPROUT | Pin Direction | Pin Data | Read PDR |
|---|---|---|---|---|---|---|
| OC1 | 1 | x | x | Output | (1) OC1Dx | External Pin |
| GPI | 0 | 0 | 0 | Input | | |
| GPO | 0 | 1 | 0 | Output | | |
| CLK_OUT | 0 | x | 1 | Output | | |

## Table B-2. OC2–OC4 Pins

| Function | OC1Mx | OMx | OLx | PDDRx | Pin Direction | Pin Data | Read PDR |
|---|---|---|---|---|---|---|---|
| OC1 | 1 | 0 | 0 | x | Output | (1) OC1Dx | External Pin |
| OCx/OC1 | 1 | 0 | 01 | x | Output | (3) OC1Dx/Toggle | External Pin |
| OCx/OC1 | 1 | 1 | 00 | x | Output | (3) OC1Dx/0 | External Pin |
| OCx/OC1 | 1 | 1 | 11 | x | Output | (3) OC1Dx/1 | External Pin |
| OCx | 0 | 0 | 1 | x | Output | (2) Toggle | External Pin |
| OCx | 0 | 1 | 0 | x | Output | (2) 0 | External Pin |
| OCx | 0 | 1 | 1 | x | Output | (2) 1 | External Pin |
| GPI | 0 | 0 | 0 | 0 | Input | External | External Pin |
| GPO | 0 | 0 | 0 | 1 | Output | PDRx | External Pin |

## Table B-3. IC40C5 Pin

| Function | IC4OC5 | OC1M7 | OM5/ OL5 | EDG4B-A | DDRI4O5 | Pin Direction | Pin Data | Read PDR |
|---|---|---|---|---|---|---|---|---|
| OC1 | 0 | 1 | 00 | xx | x | Output | (1) OC1D7 | External Pin |
| OC5/ OC1 | 0 | 1 | 01 | xx | x | Output | (3) OC1D7/Toggle | External Pin |
| OC5/ OC1 | 0 | 1 | 10 | xx | x | Output | (3) OC1D7/0 | External Pin |
| OC5/ OC1 | 0 | 1 | 11 | xx | x | Output | (3) OC1D7/1 | External Pin |
| OC5 | 0 | 0 | 01 | xx | x | Output | (2) Toggle | External Pin |
| OC5 | 0 | 0 | 10 | xx | x | Output | (2) 0 | External Pin |
| OC5 | 0 | 0 | 11 | xx | x | Output | (2) 1 | External Pin |
| IC4 | 1 | x | xx | xx | 0 | Input | External Pin | External Pin |
| GPI | 0 | 0 | 00 | xx | 0 | Input | External Pin | External Pin |
| GPO | 0 | 0 | 00 | xx | 1 | Output | PDR7 | External Pin |
| GPO | 1 | x | xx | xx | 1 | Output | PDR7 | External Pin |

## Table B-4. IC1–IC3 Pins

| Function | DDRx | EDGxB/A | Pin Direction | Pin Data | Read PDR |
|---|---|---|---|---|---|
| OCx | 0 | xx | Input | External Pin | External Pin |
| GPO | 1 | xx | Output | PDRx | External Pin |

## Table B-5. PWMA Pin

| Function | FPWMA | FIA | PPROUT | Pin Direction | Pin Data | Read F1A |
|---|---|---|---|---|---|---|
| PWM | 0 | 0 | 0 | Output | PWM | External Pin |
| PWM | 0 | 1 | 0 | Input | 1 (100% PWM) | External Pin |
| CLK_OUT | 0 | x | 1 | Output | PWM_CLK | External Pin |
| GPO | 1 | 0 | x | Output | 0 | External Pin |
| GPO | 1 | 1 | x | Output | 1 | External Pin |

## Table B-6. PWMB Pin

| Function | FPWMA | FIA | PPROUT | Pin Direction | Pin Data | Read F1B |
|---|---|---|---|---|---|---|
| PWM | 0 | 0 | 0 | Output | PWM | External Pin |
| PWM | 0 | 1 | 0 | Input | 1 (100% PWM) | External Pin |
| GPO | 1 | 0 | x | Output | 0 | External Pin |
| GPO | 1 | 1 | x | Output | 1 | External Pin |

## Table B-7. PAI, PCLK Pins

| Function | Pin Direction | Pin Data | Read PAIS, PCLKS |
|---|---|---|---|
| PAI | Input | External Pin | External Pin (PAI) |
| PCLK | Input | External Pin | External Pin (PCLK) |

# Index

## — MN —



## — O —



## — P —



## — S —

## — T —



## — U —



## — V —



## — Z —

A30025-0 PRINTED IN USA 5/91 GTE DIRECTORIES #14047 25,000 MCU YGACAA

**MOTOROLA**

**Literature Distribution Centers:**

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands, Milton Keynes, MK145 BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
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.