Freescale Semiconductor
Application Note

AN2609
Rev. 1, 08/2005

Using the General Purpose Timer of the MPC5200

by: Charles Melear

1 Introduction

The purpose of this paper is to explain the usage of the General Purpose Timer of the MPC5200. The General Purpose Timer (GPT) module is part of the System Integration Unit and consists of eight timer channels that are essentially identical.

The General Purpose Timer module uses the internal IP bus clock, which is a derivative of the system clock, as its timing reference signal. Each Timer channel is associated with a single I/O pin on the external TIMER PORT. Each Timer channel has a 16-bit prescaler and 16-bit counter, thus achieving a 32-bit range (but only 16-bit resolution).

A variety of operating modes are available for each Timer Channel. These modes are Input Capture (including Wake-Up capability on GPT 6 and GPT 7), Output Compare, Pulse Width Modulation, Simple GPIO, Watchdog Timer (GPT 0) and Internal CPU Timer.

Each timer channel has an essentially identical set of four registers. These are the GPT Enable and Mode Select

Table of Contents

freescale™ semiconductor

Register, the GPT Counter Input Register, the GPT PWM Configuration Register, and the GPT Status Register.

This paper is divided into four sections. The first section describes the TIMER PORT and how the internal Timer Module is routed to the TIMER PORT. The second section covers the various modes of operation of the Timer Channels. The third section presents the registers that configure and control the timer channels. The fourth section is a set of examples, using the Timer channels.

1.1 Timer Port Usage

The General Purpose Timer (GPT), among other functions, is routed to the TIMER PORT through a set of multiplexers. The General Purpose Timer is actually an internal module inside the System Integration Unit of the MPC5200 that shares the external TIMER PORT Pins through a set of multiplexers with other internal modules. The 8-bit TIMER PORT can take five different configurations. These configurations are programmed using the GPS Port Configuration Register and are detailed in Figure 1. The entire TIMER PORT is configured as a single element. Individual pins cannot be programmed to other functions.

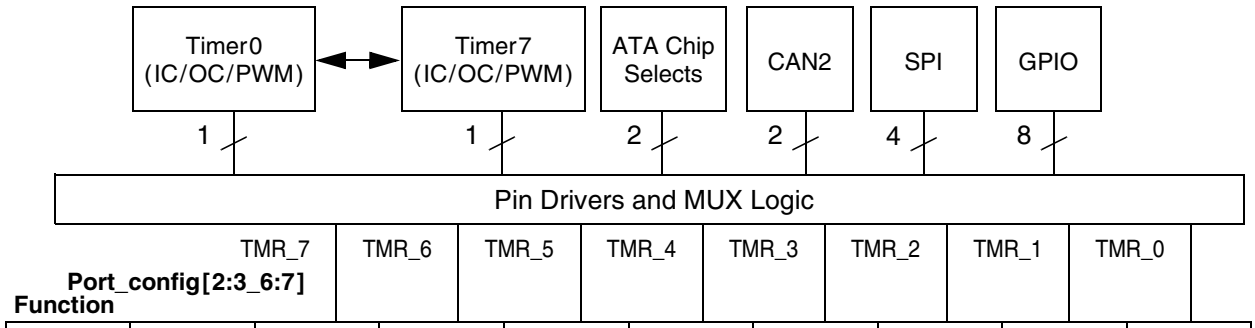
The available functions on the TIMER PORT are General Purpose Timer (GPT), Freescale Semiconductor, Inc. (formerly Motorola) Motorola Scalable Controller Area Network (MSCAN), Advanced Technology Attachment Chip Selects (ATA Chip Selects), and Serial Peripheral Interface (SPI). While all of these functions can appear on the TIMER PORT, they are not all available at the same time.

The configuration for the port can be changed at any time. There are no restrictions on changing the port configuration register other than that the external hardware connected to the TIMER PORT must be able to accommodate these changes.

The available configurations are:

1. Eight Timer Channels (which includes GPIO)
2. ATA_CS0/1 and six GPIO lines
3. MSCAN (channel 2) and six GPIO lines
4. SPI and four GPIO lines.
5. MSCAN (channel 2), SPI and two Wake-Up GPIO lines

It is important to note that the General Purpose Timer functions are always available regardless of whether the TIMER PORT pin is used or not for the General Purpose Timer. Even when a TIMER PORT Pin serves a function other than its respective timer channel, the timer channel function can still be used. For instance, if a channel is programmed to perform the Output Compare function but the Timer PORT Pin is configured for SPI operation, Timer Interrupts will be generated, but the TIMER PORT pin associated with that channel will not switch when an Output Compare event occurs.



Port_config[2:3_6:7] Function		TMR_7 TIMER7	TMR_6 TIMER6	TMR_5 TIMER5	TMR_4 TIMER4	TMR_3 TIMER3	TMR_2 TIMER32	TMR_1 TIMER1	TMR_0 TIMER0
TIMER	00_00 00_10	TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0
ATA_CS	00_11	TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	ATA_CS_1	ATA_CS_0
CAN2	01_00	TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	CAN2_RX	CAN2_TX
SPI	10_00	TIMER7	TIMER6	SPI_CLK	SPI_SS	SPI_MISO	SPI_MOSI	TIMER1	TIMER0
CAN2/SPI	11_00	TIMER7	TIMER6	SPI_CLK	SPI_SS	SPI_MISO	SPI_MOSI	CAN2_RX	CAN2_TX

Note:

1. Each Timer can be individually configured as General Purpose Input/Output (GPIO), Input Capture (IC), Output Compare (OC), or Pulse Width Modulator (PWM). If a timer pin is configured as a GPIO or some other function (SPI, ATA Chip Select, or MSCAN), the timer module can still be used internally by software.
2. Timers six and seven, when configured as input capture, contain Wake-Up functionally.
3. All Timer and GPIO function controls are within the Timer module register set.
4. MSCAN RX input supports WakeUp functionality.

Figure 1. Timer Port Functions And Configurations

The TIMER PORT, as shown in Figure 1, is configured through the GPS Port Configuration Register at memory location 0x(MBAR + 0B00), shown in Figure 2. Only bits 2:3 and 6:7 will be discussed for the present example because they are the only bits in this register that affect the operation of the TIMER PORT.

	msb 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	CS1	Rsvd	ALTs		Rsvd		ATA		IR_USB_CLK	IRDA			Rsvd	Ether		
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31 lsb
R	Rsvd			USB	PSC3				Rsvd	PSC2			Rsvd	PSC1		
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Description
2:3	ALTs	Alternatives, see Note 2 00=No Alternatives: CAN1/2 on PSC2 according to PSC2 setting. SPI on PSC3 according to PSC3 setting. 01=ALT CAN position: CAN1 on I2C1, CAN2 on TIMER PORT PINS 0/1, see Note 1 10=ALT SPI position: SPI on TIMER PORT PINS 2/3/4/5, see Note 2 11=Both on ALT
6:7	ATA	Advanced Technology Attachment 00=No ATA chip selects, $\overline{\text{LP\_CS4}}$/$\overline{\text{LP\_CS5}}$ used as normal chip select 01=ATA CS0/1 on $\overline{\text{LP\_CS4}}$/$\overline{\text{LP\_CS5}}$ 10=ATA CS0/1 on I2C_2/I2C_3 11=ATA CS0/1 on TIMER_0/TIMER_1, see Note 1

Notes:
1. ALT CAN cannot exist with ATA on Tmr0/1, nor with CAN on PSC2.
2. ALT SPI cannot exist with any SPI on PCS3.

Figure 2. GPS Port Configuration Register

To configure the TIMER PORT, user software writes to bits 2:3 - 6:7 of the GPS Port Configuration Register. The ALT field (bits 2:3) select the following. When the ALT field = 0b00, both MSCAN channels are routed to the PSC2 Port. When the ALT field = 0b01, TIMER PORT pins 0 and 1 are used for MSCAN 2. TIMER PORT pins 7 - 2 are used, respectively, for GPT Timer Channels 7 - 2. When the ALT field = 0b10, TIMER PORT pins 2 - 5 are used for the Serial Peripheral Interface. TIMER PORT pins 7, 6, 1 and 0 are used, respectively, for GPT Timer Channels 7, 6, 1 and 0. When the ALT field = 0b11, the Serial Peripheral Interface uses TIMER PORT pins 5, 4, 3 and 2 and the MSCAN 2 function uses TIMER PORT pins 1 and 0. TIMER PORT pins 7 - 6 are used, respectively, for GPT Timer Channels 7 - 6.

Bits 6:7 of the GPS Port Configuration Register determine whether ATA Chip Selects will be used and which port the ATA Chip Select function will appear. For the purposes of this application note, it is assumed that bits 6:7 will NOT be programmed to 0b11, which would cause TIMER PORT pins 1 and 0 to be ATA Chip Selects 1 and 0, respectively.

NOTE

If the SPI configuration is selected, TIMER PORT Pins 5 - 2 are under the control of the Serial Peripheral Interface and TIMER PORT Pins 7 - 6 and 1 - 0 are under the control of the General Purpose Timer. Likewise, if the CAN2 configuration is selected, TIMER PORT Pins 1 - 0 are under the control of the MSCAN Interface and TIMER PORT Pins 7 - 2 are under the control of the General Purpose Timer. If the CAN/SPI configuration is chosen, TIMER PORT pins 5 - 2 are under the control of the Serial Peripheral Interface and TIMER PORT pins 1 - 0 are under the control of the MSCAN interface. Only TIMER PORT pins 7 and 6 are under the control of the General Purpose Timer. If the ATA Chip Select configuration is selected, TIMER PORT Pins 1 - 0 are under the control of the ATA Interface and TIMER PORT Pins 7 - 2 are under the control of the General Purpose Timer.

2 Modes of Operation

2.1 Input Capture

In this mode, the TIMER PORT I/O pin associated with a particular TIMER channel is automatically configured as an input. Once enabled, the counter for the selected TIMER channel runs continuously. The “Capture Event” can be specified as a rising edge, falling edge, or either a rising or falling edge. The value of the counter will be latched into the CAPTURE field of the GPTx Status Register when the specified “capture event” occurs. If the “Capture Event” is a “pulse”, the width of the pulse, expressed in prescaled IP Bus clocks, is recorded in the CAPTURE field of the GPTx Status Register. If enabled, a capture event causes a CPU interrupt to be generated.

2.2 Output Compare

In this mode, the TIMER PORT I/O pin associated with a particular TIMER channel is automatically configured as an output. Once enabled, the counter for the selected TIMER channel runs continuously. Each time the counter increments by the number of counts stored in the COUNT field of the GPTx Counter Input Register, an Output Event will occur. That is, the associated TIMER PORT Pin will output a high active pulse, a low active pulse, or toggle from its current state. If enabled, an Output Compare event causes a CPU interrupt to be generated.

Once a TIMER channel is programmed and enabled in the Output Compare mode and the “toggle” function is selected, the TIMER PORT Pin will output a square wave, as long as the user software does not modify the COUNT field of the GPTx Counter Input Register. The width of the high time and low time is determined by the number of clock counts programmed into the COUNT field of the GPTx Counter Input Register.

2.3 PWM

In this mode, the TIMER PORT I/O pin associated with a particular Timer channel is an Output. The user can program “Period” and “Width” values to create an adjustable, repeating output waveform on the associated TIMER PORT I/O pin. A CPU interrupt can be generated at the beginning of each PWM Period, at which time a new Width value can be loaded. The new Width value, which represents “ON time,” is automatically applied at the beginning of the next period. This mode is suitable for PWM audio encoding.

2.4 Simple GPIO

In this mode, the TIMER PORT I/O pin associated with a particular Timer channel operates as a GPIO pin. It can be specified as Input or Output. The GPIO mode is mutually exclusive of the Input Capture Mode, Output Compare Mode, or PWM Mode. When a TIMER PORT Pin is configured as GPIO, it is still possible to use the CPU Timer Mode and the Watchdog Timer Mode.

2.5 CPU Timer

The TIMER PORT I/O pin associated with a particular Timer channel is not used in this mode. Once enabled, the counters run until they reach a programmed Terminal Count. When this occurs, an interrupt can be generated to the CPU. This Timer mode can be used simultaneously with the Simple GPIO mode.

2.6 Watchdog Timer

This is a special CPU Timer mode available only on Timer 0. The user must enable the Watchdog Timer mode, which is not active upon reset. The Terminal Count value is programmable. If the counter is allowed to expire, a full MPC5200 reset occurs. To prevent the Watchdog Timer from expiring, software must periodically write the value 0xA5 to the OCPW field of GPT0 Enable and Mode Select Register (in Timer 0). This causes the counter to be reloaded with the value in the COUNT field of the GPT0 Counter Input Register.

NOTE

A value of 0x0000 or 0x0001 cannot be used as a time-out value for the COUNT field. Both values are treated as an immediate time out, but no reset will occur. In effect, both of these values will essentially disable the watchdog function. For proper watchdog operation, the COUNT field must contain a value in the range of 0x0002 to 0xFFFF.

When a reset is caused by the expiration of the Watchdog Timer, the MPC5200 is reset. This means that the Timer Channels will become inactive. Therefore, from the time that the reset occurs until software sets up Timer Channel 0 to perform the watchdog function, the watchdog will be inactive.

3 GPT Register Set

3.1 GPT Registers—MBAR + 0x0600

There are four registers associated with each timer channel. These are the GPT (0 - 7) Enable and Mode Select Register, the GPT (0 - 7) Counter Input Register, the GPT (0 - 7) PWM Configuration Register, and the GPT (0 - 7) Status Register. These registers are presented here for reference and will be referred to while discussing actual examples.

The General Purpose Timer Module is located at an offset of 0x0600 from the value in the MBAR Register. GPT register addresses are relative to this offset. Therefore, the actual register address is: **MBAR + 0x0600 + register address**

3.2 GPT 0 Enable and Mode Select RegisterMBAR + 0x0600
GPT 1 Enable and Mode Select RegisterMBAR + 0x0610
GPT 2 Enable and Mode Select RegisterMBAR + 0x0620
GPT 3 Enable and Mode Select RegisterMBAR + 0x0630
GPT 4 Enable and Mode Select RegisterMBAR + 0x0640
GPT 5 Enable and Mode Select RegisterMBAR + 0x0650
GPT 6 Enable and Mode Select RegisterMBAR + 0x0660
GPT 7 Enable and Mode Select RegisterMBAR + 0x0670

	msb 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	OCPW								Reserved		OCT		Reserved		ICT	
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31 lsb
R W	WDen	Reserved		CE	Rsvd	Stop_Cont	Open_Drn	IntEn	Reserved		GPIO		Rsvd	Timer_MS		
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3. GPT (0 - 7) Enable and Mode Select Register

Bit	Name	Description
0:7	OCPW	Output Compare Pulse Width—Applies to OC Pulse types only. This field specifies the number of IP bus clocks (non-prescaled) to create a short output pulse at each Output Event. This pulse is generated at the end of the OC period and overlays the next OC period (rather than adding to the period). Note: This field is alternately used as the watchdog reset field if Watchdog Timer mode is \enabled.
8:9	—	Reserved

Bit	Name	Description
10:11	OCT	Output Compare Type—describes action to occur at each output compare event, as follows: 00=Special case, output is immediately forced low. 01=Output pulse highs, initial value is low (OCPW field applies). 10=Output pulses low, initial value is high (OCPW field applies). 11=Output toggles. GPIO modalities can be used to achieve an initial output state prior to enabling OC mode. It is important to move directly from GPIO output mode to OC mode and not to pass through the Timer_MS=000 state. To prevent the Internal Timer Mode from engaging during the GPIO state, CE bit should be held low during the configuration steps. GPIO initialization is needed when presetting the I/O to 1 in conjunction with a simple toggle OCT setting. Note: For Stop Mode operation (see Stop_Cont bit below) it is necessary to pass through the mode_sel = 0 state to restart the output compare counters with their programmed values. See prescale and count fields in Section 3.3, "GPT 0 Counter Input Register MBAR + 0x0604 GPT 1 Counter Input Register MBAR + 0x0614 GPT 2 Counter Input Register MBAR + 0x0624 GPT 3 Counter Input Register MBAR + 0x0634 GPT 4 Counter Input Register MBAR + 0x0644 GPT 5 Counter Input Register MBAR + 0x0654 GPT 6 Counter Input Register MBAR + 0x0664 GPT 7 Counter Input Register MBAR + 0x0674.
12:13	—	Reserved
14:15	ICT	Input Capture Type - describes the input transition type required to trigger an input capture event, as follows: 00=Any input transition causes an IC event. 01=IC event occurs at input rising edge. 10=IC event occurs at input falling edge. 11=IC event occurs at any input pulse (i.e., at 2nd input edge). **BE AWARE:** For ICT=11 (pulse capture), status register records only the pulse width.
16	WDen	Watchdog enable - bit enables watchdog operation. A timer expiration causes an internal MPC5200 reset. Watchdog operation requires the Timer_MS field be set for internal timer mode and the CE bit to be set high. In this mode the OCPW byte field operates as a watchdog reset field. Writing 0xA5 to the OCPW field resets the Watchdog Timer, preventing it from expiring. As long as the timer is properly configured, the watchdog operation continues. This bit (and functionality) is implemented only for Timer 0. 1 = enabled
17:18	—	Reserved
19	CE	Counter Enable - bit enables or resets the internal counter during Internal timer modes only. CE must be high to enable these modes. If low, counter is held in reset. This bit is secondary to the timer mode select bits (Timer_MS). If Timer_MS is1XX, internal timer modes are enabled. CE can then enable or reset the internal counter without changing the Timer_MS field. GPIO operation is also available in this mode. 1 = enabled
20	—	Reserved

Bit	Name	Description
21	Stop_Cont	Stop Continuous - Applies to multiple modes, as follows: 0 = Stop 1 = Continuous • IC mode Stop operation - At each IC event, counter is reset. Continuous operation - counter is not reset at each IC event. Effect is to create Status count values that are cumulative between Capture events. If the special Pulse Mode Capture type is specified, the Stop_Cont bit is not used, operation fixed as if it were Stop. • OC mode Stop operation - Counter resets and stops at first OC event. Continuous operation - counter resets and continues at each OC event. Effect to is create back-to-back periodic OC events. **BE AWARE -** In this mode the polarity of Stop_cont is reversed. Also, in Stop Mode, the output event falsely triggers at the expiration of the prescale count. This means the software has to service and output event prior to the prescale expiring. Service is defined as programming mode_sel field to 0, which causes the programmed prescale and count values to be reset. • PWM mode Bit not used, operation is always Continuous. • CPU Timer mode Stop operation - On counter expiration, Timer waits until Status bit is cleared before beginning a new cycle. Continuous operation - On counter expiration, Timer resets and immediately begin a new cycle. Effect is to generate fixed periodic time-outs. • WatchDog Timer and GPIO modes Bit not used.
22	Open_Drn	Open Drain 0 = Normal I/O 1 = Open Drain emulation - affects all modes that drive the I/O pin (GPIO, OC, & PWM). Any output “1” is converted to a tri-state at the I/O pin.
23	IntEn	Enable interrupt - enables interrupt generation to the CPU for all modes (IC, OC, PWM, and Internal Timer). IntEn is not required for watchdog expiration to create a reset. 1 = enabled
24:25	—	Reserved
26:27	GPIO	GPIO mode type. Simple GPIO functionality that can be used simultaneously with the Internal Timer mode. It is not compatible with IC, OC, or PWM modes, since these modes dictate the usage of the I/O pin. 0x=Timer enabled as simple GPIO input 10=Timer enabled as simple GPIO output, value=0 11=Timer enabled as simple GPIO output, value=1 (tri-state if Open_Drn=1) While in GPIO modes, internal timer mode is also available. To prevent undesired timer expiration, keep the CE bit low.

Bit	Name	Description
28	—	Reserved
29:31	Timer_MS	Timer Mode Select (and module enable). 000=Timer module not enabled. Associated I/O pin is in input state. All Timer operation is completely disabled. Control and status registers are still accessible. This mode should be entered when timer is to be re-configured, except where the user does not want the I/O pin to become an input. 001=Timer enabled for input capture. 010=Timer enabled for output compare. 011=Timer enabled for PWM. 1xx=timer enabled for simple GPIO. Internal timer modes available. CE bit controls timer counter.

3.3 GPT 0 Counter Input RegisterMBAR + 0x0604
GPT 1 Counter Input RegisterMBAR + 0x0614
GPT 2 Counter Input RegisterMBAR + 0x0624
GPT 3 Counter Input RegisterMBAR + 0x0634
GPT 4 Counter Input RegisterMBAR + 0x0644
GPT 5 Counter Input RegisterMBAR + 0x0654
GPT 6 Counter Input RegisterMBAR + 0x0664
GPT 7 Counter Input RegisterMBAR + 0x0674

	msb 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	Prescale															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31 lsb
R W	Count															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. GPT (0-7) Counter Input Register

Bit	Name	Description
0:15	Prescale	Prescale amount applied to internal counter (in IP bus clocks). **BE AWARE** - The prescale field should be written prior to enabling any timer mode. A prescale of 0x0001 means one IP bus clock per count increment. If prescale is 0 when any timer mode is started, it results in an effective prescale of 64K. The counter will immediately begin and an output event will occur with the 64K prescale, rather than the desired value.
16:31	COUNT	Sets number of prescaled counts applied to reference events, as follows: IC - Field has no effect, internal counter starts at 0. OC - Number of prescaled counts counted before creating output event. PWM - Number of prescaled counts defining the PWM output period. Internal Timer - Number of prescaled counts counted before timer (or watchdog) expires. Note: Reading this register only returns the programmed value, intermediate values of the internal counter are not available to software.

3.4 GPT 0 PWM Configuration RegisterMBAR + 0x0608
GPT 1 PWM Configuration RegisterMBAR + 0x0618
GPT 2 PWM Configuration RegisterMBAR + 0x0628
GPT 3 PWM Configuration RegisterMBAR + 0x0638
GPT 4 PWM Configuration RegisterMBAR + 0x0648
GPT 5 PWM Configuration RegisterMBAR + 0x0658
GPT 6 PWM Configuration RegisterMBAR + 0x0668
GPT 7 PWM Configuration RegisterMBAR + 0x0678

	msb 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	WIDTH															
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31 lsb
R	Reserved							PWMOP	Reserved							LOAD
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5. GPT (0 - 7) PWM Configuration Register

Bit	Name	Description
0:15	WIDTH	PWM only. Defines ON time for output in prescaled counts. Similar to count value, which defines the period. ON time overlays the period time. If WIDTH = 0, output is always OFF. If WIDTH exceeds count value, output is always ON. ON and OFF polarity is set by the PWMOP bit.
16:22	—	Reserved
23	PWMOP	Pulse Width Mode Output Polarity—Defines PWM output polarity for OFF time. Opposite state is ON time polarity. PWM cycles begin with ON time.
24:30	—	Reserved
31	LOAD	Bit forces immediate period update. Bit auto clears itself. A new period begins immediately with the current count and width settings. If LOAD = 0, new count or width settings are not updated until end of current period. Note: Prescale setting is not part of this process. Changing prescale value while PWM is active causes unpredictable results for the period in which it was changed. The same is true for PWMOP bit.

3.5 GPT 0 Status RegisterMBAR + 0x060C
GPT 1 Status RegisterMBAR + 0x061C
GPT 2 Status RegisterMBAR + 0x062C
GPT 3 Status RegisterMBAR + 0x063C
GPT 4 Status RegisterMBAR + 0x064C
GPT 5 Status RegisterMBAR + 0x065C
GPT 6 Status RegisterMBAR + 0x066C
GPT 7 Status RegisterMBAR + 0x067C

This is a read-only register.

	msb 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	CAPTURE															
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31 lsb
R	Rsvd	OVF			Reserved			PIN	Reserved				TEXP	PWMP	COMP	CAFT
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6. GPT (0 - 7) Status Register

Bit	Name	Description
0:15	Capture	Read of internal counter latch at reference event. This is pertinent only in IC mode, in which case it represents the count value at the time the Input Event occurred. Capture status does not shadow the internal counter while an event is pending, it is updated only at the time the Input Event occurs. Note: If ICT is set to 11, which is Pulse Capture Mode, the Capture value records the width of the pulse. Also, the Stop_Cont bit is irrelevant in Pulse Capture Mode, operation is as if Stop_Cont were 0.
16	—	Reserved
17:19	OVF	Represents how many times internal counter has rolled over. This is pertinent only during IC mode and would represent an extremely long period of time between Input Events. However, if Stop_Cont = 1 (indicating cumulative reporting of Input Events), this field could come into play. This field is cleared by any “sticky bit” status write in bit positions 28, 29, 30, 31.
20:22	—	Reserved
23	PIN	Registered state of the I/O PIN (all modes). The IP Bus Clock registers the state of the I/O input. Valid, even if Timer is not enabled.
24:27	—	Reserved
28	TEXP	Timer Expired in Internal Timer mode. Cleared by writing 1 to this bit position. Also cleared if Timer_MS is 000 (i.e., Timer not enabled). See Note.
29	PWMP	PWM end of period occurred. Cleared by writing 1 to this bit position. Also cleared if Timer_MS is 000 (i.e., Timer not enabled). See Note.
30	COMP	OC reference event occurred. Cleared by writing 1 to this bit position. Also cleared if Timer_MS is 000 (i.e., Timer not enabled). See Note.
31	CAPT	IC reference event occurred. Cleared by writing 1 to this bit position. Also cleared if Timer_MS is 000 (i.e., Timer not enabled). See Note.
Note: To clear any of these bits, it is necessary to clear **all** of them. An F must be written to bits 28:31.		

4 General Purpose Timer Programming Examples

4.1 Timer Configuration Method

Use the following method to configure each timer:

1. Determine the Mode Select field (Timer_MS) value for the desired operation.
2. Program any other registers associated with this mode.
3. Program Interrupt enable as desired.
4. Enable the Timer by writing the Mode Select value into the Timer_MS field.

4.2 Programming Notes

Programmers should observe the following notes:

- Intermediate values of the Timer internal counters are **not** readable by software.

- The Stop_Cont bit operates differently for different modes. In general, this bit controls whether the Timer halts at the end of a current mode or resets and continues with a repetition of the mode. See Section 4.2 for the Bit Descriptions of the GPTx Enable & Mode Select Register for precise operation.
- The Timer_MS field operates somewhat as a Global Enable. If it is zero, then all Timer modes are disabled and internal counters are reset. See Section 4.2 for the Bit Descriptions of the GPTx Enable & Mode Select Register for precise operation.
- There is a CE (Counter Enable) bit that operates somewhat independently of the Timer_MS field. This bit controls the Counter for CPU Timer or Watchdog Timer modes only. See Section 4.2 for the Bit Descriptions of the GPTx Enable & Mode Select Register for precise operation for more details.

4.3 Simple GPIO Mode

To use a TIMER PORT pin as a GPIO pin, the pin must be configured as a TIMER pin. If the TIMER PORT pin is configured as “TIMER”, all 8 TIMER PORT pins are configured as TIMER pins. If the TIMER PORT is configured as ATA_CS (ATA Chip Selects), TIMER PORT pins 7 - 2 are configured as TIMER pins. Likewise, CAN2, SPI and CAN2/SPI configuration modes can be selected and certain TIMER PORT pins will be configured as TIMER pins according to Figure 1.

There is no explicit setting of the GPS Port Configuration Register to configure the TIMER PORT as GPIO lines. Once a TIMER PORT pin is configured for GPT usage via the GPS Port Configuration Register, the TIMER PORT pin must be configured as GPIO using the GPIO field and the Timer_MS field of the GPT (0 -7) Enable and Mode Select Register. There is one GPT Enable and Mode Select Register for each timer channel.

The Timer_MS bits are used to configure a Timer for Input Capture, Output Compare, Pulse Width Modulation or General Purpose Input/Output. If the Timer_MS field is set to GPIO operation, the GPIO field is used to configure a TIMER PORT pin as a GPIO input, a GPIO output driving to a logic 1 or a GPIO output driving to a logic 0. The GPIO field only has meaning if the Timer_MS field is set for GPIO operation.

Regardless of how these fields are set, the internal functions of the Timer Channels can be used to generate internal timing events. For instance, if a Timer Channel is programmed for Output Compare operation via the Timer_MS field, an interrupt will occur at the specified time out period regardless of whether or not the TIMER PORT pin is configured for Timer operation or not via the GPS Port Configuration Register.

If the TIMER PORT is configured for SPI operation, TIMER PORT pins 5, 4, 3, and 2 are dedicated for the SPI module and these TIMER PORT pins will be unavailable for Timer functions, such as outputting a PWM signal. However, when the TIMER PORT is configured for SPI operation, TIMER PORT pins 7, 6, 1, and 0 are still available for use by their respective Timer channels.

NOTE

The code examples in the application note were developed on an Enchilada Validation Board (Tortilla V1.0) with the GPIO Arnold daughter card. The following GPIO channels are used on the Arnold daughter card: TIMER 0 - GPIO12, TIMER 1 - GPIO13, TIMER 2 - GPIO6, TIMER 3 - GPIO7, TIMER 4 - GPIO8, TIMER 5 - GPIO9, TIMER 6 - GPIO15, and TIMER 7 - GPIO16.

4.4 EXAMPLE 1 Configure TIMER PORT Pin 0 as GPIO and then Switch States in an infinite Loop.

To begin, the GPS Port Configuration Register bits 2:3 and 6:7 will be programmed to 0b00 and 0b00, respectively. This will cause all eight TIMER PORT Pins to be under the control of the General Purpose Timer Module.

The TIMER PORT is unique in that there is no Port Data Register or Port Data Direction Register associated with this port. The General Purpose Timer module is used to control the TIMER PORT Pins as GPIO. Essentially, all other port pins associated with the other MPC5200 ports, when configured as GPIO, use specific Data Direction Registers and Port Data Registers.

NOTE

If other configurations are selected for the TIMER PORT, some of the port pins will be reassigned to functions other than GPIO. However, for pins that retain GPIO functionality, this example will apply.

This program simply turns TIMER PORT Pin 0 on and off. Then the program loops on the on-off portion.

NOTE

The GPS Port Configuration Register bits 2:3 and 6:7 are programmed to 0b00 and 0b00, respectively. This will cause all eight TIMER PORT Pins to be under the control of the General Purpose Timer Module.

```
200000  70630000andi.   r3,r3,0x0000;r3 = 0x00000000, clear r3

200004  70840000andi.   r4,r4,0x0000;r4 = 0x00000000, clear r4

200008  70A50000andi.   r5,r5,0x0000;r5 = 0x00000000, clear r5

20000C  6463F000oris    r3,r3,0xF000;MBAR = 0xF000

200010  60630600ori     r3,r3,0x0600;r3 = 0xF0000600,address of GPT 0

                                ;Enable and Mode Select Register

200014  60840034ori     r4,r4,0x0034;r4 = 0x0034

200018  90830000stw     r4,0(r3);(EA: FF000600)

                                ;Write Timer MS field

                                ;Timer 0 port pin = 1
```

```
20001C  70840024andi.   r4,r4,0x0024;r4 =0x0024

200020  90830000stw     r4,0(r3);(EA: FF000600)
                                ;Write Timer MS field
                                ;Timer 0 port pin = 0

200024  4BFFFFF0b       0x00200014;loop forever
```

This program has the following flow:

1. Clear General Purpose Registers r3, r4, and r5
2. Load r3 with 0xF0000600, the base address of the General Purpose Timer
3. Load r4 with 0x34 and store the value to the Timer_MS field of GPT 0 Enable and Mode Select Register to make TIMER PORT Pin 0 = logic 1
4. Load r4 with 0x24 and store the value to the Timer_MS field of GPT 0 Enable and Mode Select Register to make TIMER PORT Pin 0 = logic 0
5. Branch to step 3

4.5 EXAMPLE 2 Program to Turn All the Timer Channels ON and then OFF, One at a Time

NOTE

The “ALT” field of the Port Configuration Register (MBAR + 0x0B00) must be set to 0b00, which makes the TIMER PORT pins function as general purpose I/O (GPIO).

Before the following program was run, the GPS Port Configuration Register at MBAR + 0x0B00 was set to 0x00000005. (The “5” in bits 28 - 31 turns on PSC1 and has no effect on the TIMER PORT.)

This program is very similar in function to example 1, except that all TIMER PORT Pins are configured as GPIO Channels. All TIMER PORT Pins are individually driven to a logic 1 and then driven to a logic 0. This is done by writing to the Timer_MS field of the General Purpose Timer (0 - 7) Enable and Mode Select Register.

```
200000  70630000andi.   r3,r3,0x0000;r3 = 0x00000000

200004  70840000andi.   r4,r4,0x0000;r4 = 0x00000000

200008  70A50000andi.   r5,r5,0x0000;r5 = 0x00000000

20000C  6463F000oris    r3,r3,0xF000

200010  60630600ori     r3,r3,0x0600put base address of GPT in R3

200014  60840034ori     r4,r4,0x0034put immediate value of 0x34 in R4

200018  90830000stw     r4,0(r3)(EA: FF000600) write Timer MS field
                                Timer 0 = output, logic 1

20001C  90830010stw     r4,16(r3) (EA: FF000610) write Timer MS field
```

```
                              Timer 1 = output, logic 1
200020  90830020stw     r4,32(r3)(EA: FF000620) write Timer MS field
                              Timer 2 = output, logic 1
200024  90830030stw     r4,48(r3)(EA: FF000630) write Timer MS field
                              Timer 3 = output, logic 1
200028  90830040stw     r4,64(r3)(EA: FF000640) write Timer MS field
                              Timer 4 = output, logic 1
20002C  90830050stw     r4,80(r3)(EA: FF000650) write Timer MS field
                              Timer 5 = output, logic 1
200030  90830060stw     r4,96(r3)(EA: FF000660) write Timer MS field
                              Timer 6 = output, logic 1
200034  90830070stw     r4,112(r3) (EA: FF000670) write Timer MS field
                              Timer 7 = output, logic 1
200038  70840000andi.   r4,r4,0x0000
20003C  60840024ori     r4,r4,0x0024r4 = 0x00000024
200040  90830000stw     r4,0(r3)(EA: FF000600) write Timer MS field
                              Timer 0 = output, logic 0
200044  90830010stw     r4,16(r3)(EA: FF000610) write Timer MS field
                              Timer 1 = output, logic 0
200048  90830020stw     r4,32(r3)(EA: FF000620) write Timer MS field
                              Timer 2 = output, logic 0
20004C  90830030stw     r4,48(r3)(EA: FF000630) write Timer MS field
                              Timer 3 = output, logic 0
200050  90830040stw     r4,64(r3)(EA: FF000640) write Timer MS field
                              Timer 4 = output, logic 0
200054  90830050stw     r4,80(r3)(EA: FF000650) write Timer MS field
                              Timer 5 = output, logic 0
200058  90830060stw     r4,96(r3)(EA: FF000660) write Timer MS field
                              Timer 6 = output, logic 0
20005C  90830070stw     r4,112(r3)(EA: FF000670) write Timer MS field
                              Timer 7 = output, logic 0
```

4.6 EXAMPLE 3 Configure TIMER PORT Pins 0 and 1 for Pulse Width Modulation.

In example 3, Timer 0 and Timer 1 will be configured to generate a Pulse Width Modulated signal with an initial period of 0xF000 prescaled timer clocks with an “ON” time of 0x8000 prescaled timer clocks. The prescaler is 0xFF00. This means that each “prescaled timer clock” will be 0xFF00 cycles of the IP Bus clock. The prescaler can range from 0x0001 to 0xFFFF. A prescaler of 0x0000 effectively acts as though a prescale value of 0xFFFF were used.

The waveform for this example is shown in Figure 7.

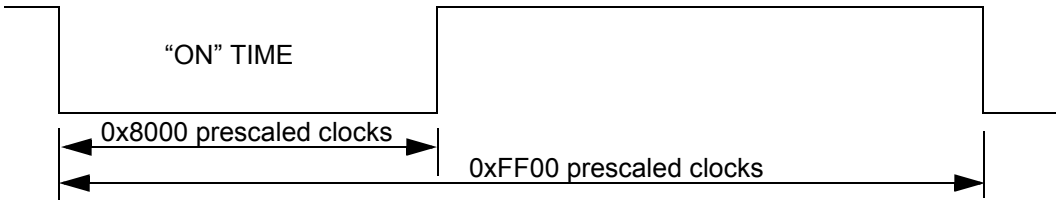


Figure 7. PWM Waveform

This program has the following program flow:

1. Clear General Purpose Registers r3, r4, and r5
2. Load r3 with 0xF0000600, the base address for the General Purpose Timer
3. Load r4 with 0xFF00F000 and store that value to GPT 0 Counter Input Register
4. This sets the Prescaler to 0xFF00 and the Period of the PWM to 0xF000 prescaled clocks
5. Load r4 with 0x80000100 and store to GPT 0 PWM Configuration Register
6. This sets the “ON” time of the PWM to 0x8000 prescaled clocks and defines the “OFF” time to be a logic 1.
7. Load r4 with 0x00000003 and store to GPT 0 Enable and Mode Select Register
8. This starts the PWM signal which begins with “ON” time.
9. Load r4 with 0xFF00F000 and store that value to GPT 1 Counter Input Register
10. This sets the Prescaler to 0xFF00 and the Period of the PWM to 0xF000 prescaled clocks
11. Load r4 with 0x80000100 and store to GPT 1 PWM Configuration Register
12. This sets the “ON” time of the PWM to 0x8000 prescaled clocks and then defines the “OFF” time to be a logic 1.
13. Load r4 with 0x00000003 and store to GPT 1 Enable and Mode Select Register
    This starts the PWM signal which begins with “ON” time.

NOTE

In this case, the PWM "OFF" time is a logic 1. While it is not done in this example, the system designer may first configure the TIMER PORT pins as GPIO, write the Timer_MS field to drive the Timer Pins high, and then run the following program, which configures the channel for PWM operation. This method will prevent an unexpected transition at the beginning of the first PWM period.

```
200000  70630000andi.   r3,r3,0x0000;clear r3
200004  70840000andi.   r4,r4,0x0000;clear r4
200008  70A50000andi.   r5,r5,0x0000;clear r5
20000C  6463F000oris    r3,r3,0xF000
200010  60630600ori     r3,r3,0x0600;load r3 with 0xF0000600
                                ;base address of GPT
200014  70840000andi.   r4,r4,0x0000;clear r4
200018  6484FF00oris    r4,r4,0xFF00
20001C  6084F000ori     r4,r4,0xF000;load r4 with 0xFF00F000
200020  90830004stw     r4,4(r3);prescale = 0xFF00, count = 0xF000
200024  70840000andi.   r4,r4,0x0000;clear r4
200028  64848000oris    r4,r4,0x8000
20002C  60840100ori     r4,r4,0x0100load r3 with 0x80000100
200030  90830008stw     r4,8(r3);width = 0x8000, PWMOP bit = 1
200034  70840000andi.   r4,r4,0x0000
200038  64840000oris    r4,r4,0x0000
20003C  60841004ori     r4,r4,0x1004
200040  9083000Cstw     r4,12(r3);
200034  70840000andi.   r4,r4,0x0000
200038  64840000oris    r4,r4,0x0000
20003C  60840003ori     r4,r4,0x0003;load r4 with 0x00000003
200040  90830000stw     r4,0(r3);set Timer_MS field to
                                ;0b011 - PWM
                                ;start PWM
200044  70840000andi.   r4,r4,0x0000;clear r4
```

```
200048  6484FF00oris    r4,r4,0xFF00
20004C  6084F000ori     r4,r4,0xF000;load r4 with 0xFF00F000
200050  90830014stw     r4,20(r3);prescale = 0xFF00, count = 0xF000
200054  70840000andi.   r4,r4,0x0000
200058  64848000oris    r4,r4,0x8000
20005C  60840100ori     r4,r4,0x0100load r3 with 0x80000100
200060  90830018stw     r4,24(r3);width = 0x8000, PWMOP bit = 1
200064  70840000andi.   r4,r4,0x000;clear r4
200068  64840000oris    r4,r4,0x0000
20006C  60841003ori     r4,r4,0x0003;load r4 with 0x00000003
200070  90830010stw     r4,16(r3);set Timer_MS field to
                                  ;0b011 - PWM
                                  ;start PWM
```

5 Output Compare Mode

The Output Compare mode works in the following manner. The desired Timer Channel is configured, using GPIO mode, to be a logic 1 or a logic 0 as dictated by system needs. The selected Timer Channel can be programmed to immediately drive to a logic 0, output a high active pulse, output a low active pulse, or toggle from its current state. When a high active pulse is desired, it is the responsibility of the system software to ensure that the initial value of the Timer Channel is at a logic 0. Likewise, when a low active pulse is desired, it is also the responsibility of the system software to ensure that the initial value of the Timer Channel is at a logic 1. Depending upon system needs, when using the “toggle” mode, there may be some need to have the Timer Channel at a specified initial value. This can be accomplished by using the GPIO mode and then setting up and enabling the Timer Channel for the Output Compare mode.

5.1 EXAMPLE 4 Configure TIMER PORT Pin 7 for Immediate Force Low Mode of the Output Compare Function

When a General Purpose Timer channel is configured in the Output Compare function, the “immediate force low” mode can be used to drive the TIMER PORT pin to a logic 0. To implement this mode, only the GPT (0-7) Enable and Mode Select Register must be modified. Specifically, the OCT field and the Timer_MS field must be written. The OCT field is written to 0b00, and the Timer_MS field is written to 0b010. These fields, respectively, enable the “immediate force low” mode and put the timer channel in the Output Compare function.

```
2000A0  70A50000andi.   r5,r5,0x0000
2000A4  64A50000oris    r5,r5,0x0000
```

```
2000A8  60A50002ori     r5,r5,0x0002;load r5 with 0x00000002

2000AC  70840000andi.   r4,r4,0x0000

2000B0  6484F000oris    r4,r4,0xF000

2000B4  60840600ori     r4,r4,0x0600;load r4 with 0xF0000600

2000B8  90A40070stw     r5,112(r4);write GPT 7 Enable and Mode Select
                                   ;Register
                                   ;OCT field = 0b00 and
                                   ;Timer_MS field = 0b01
                                   ; to force Timer 7 to a logic 0
```

5.2 EXAMPLE 5 Configure TIMER PORT Pin 7 for Toggle Mode of the Output Compare function

The Toggle Mode can be used to switch the state of a Timer Channel from its current state to the opposite state. When the STOP_CONT bit is set to enable continuous operation, the Toggle Mode can be used to generate a square wave.

In order to set up the Toggle Mode, the GPT Enable and Mode Select Register and the GPT Counter Input Register are employed. The program flow is as follows:

1. The GPT Counter Input Register is written to specify the Prescale and Count values. This will determine the period of the “toggle” on and off time.
2. The GPT Enable and Mode Select Register is written, so that the OCT field specifies toggle mode, the Stop_Cont field specifies “continuous”.

NOTE

Once the “TOGGLE MODE” is started and the CONT mode is used, the Timer Channel will toggle continuously. The High Time and the Low Time will be equal. Both High and Low Time will have a duration of IP Bus Clock Period x Prescale X Count. Software can modify the GPTx Counter Input Register at any time to change the current period.

Program to create a toggle function:

```
200000   70630000andi.  r3,r3,0x0000;clear r3

200004   70840000andi.  r4,r4,0x0000;clear r4

200008   70A50000andi.  r5,r5,0x0000;clear r5

20000C   6463F000oris   r3,r3,0xF000

200010   60630600ori    r3,r3,0x0600

200014   70840000andi.  r4,r4,0x0000;load r3 with 0xF0000600
                                   ;base address for GPT
```

```
                                ;module
200018   64840008oris   r4,r4,0x0008
20001C   608400FFori    r4,r4,0x00FF;load r4 with 0x000800FF
                                ;prescale = $08, count = $FF
200020   90830074stw    r4,116(r3) ;write GPT 7 PWM Config.
                                ;Reg.
                                (EA: 00000074)
200024   70840000andi.  r4,r4,0x0000
200028   64840830oris   r4,r4,0x0830
20002C   60840402 ori   r4,r4,0x0402;value for “toggle mode” and
                                ;”output compare” function.
200030   90830070stw    r4,112(r3);write GPT 7 Enable and
                                ;Mode
                                ;Select Register
                                ; (EA: 00000070)
200034   48000000b      0x00200034;branch forever
```

In this example, the IP Bus Clock is 66 MHz, the prescale value is 0x8, and the count is 0xFF. The high time of the “toggle” waveform, shown in Figure 8, is given by the equation:

low time = high time = prescale x count x IP Bus clock period
low time = 8 x 256 x 15.1 ns = 30924.8 ns

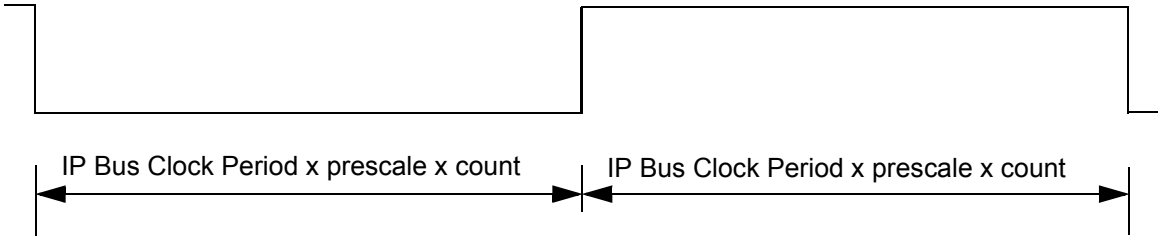


Figure 8. TOGGLE MODE WAVEFORM

5.3 Example 6 Configure Timer 7 for Output Pulse High Mode of Output Compare

The Pulse High Mode is used to create an active high pulse or an active low pulse. This mode can be programmed to produce one pulse, or it can be programmed to output a periodic pattern. The active pulse width is specified in IP Bus Clock cycles and can have values ranging from 0x1 to 0xFF. (Note: There is

no prescaling of the IP Bus clock in calculating the “active pulse” time.) The period of the Pulse High (or Low) Mode is specified in “prescaled” IP Bus clock cycles. A typical wave form for the Pulse High Mode is shown in Figure 9

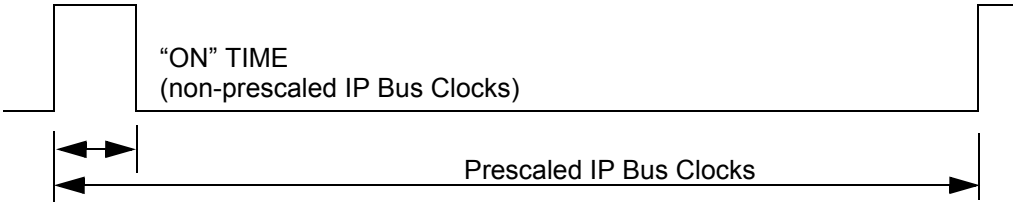


Figure 9. Pulse High Mode Waveform

The program flow to obtain the above wave form is as follows:

1. Write the GPT 7 Counter Input Register with the Prescale and Counter values to obtain the desired period. The period is calculated by the following formula:
   Period = IP Bus Clock Period x Prescaler x Count
2. Write the GPT 7 Enable and Mode Select Register. The OCPW field specifies the Pulse High (or Low) Time in IP Bus clock cycles. The OCT field specifies the Output Compare Type to be Output Pulse High and the Timer_MS field to Output Compare.

NOTE

It is the responsibility of the system programmer to ensure that the Timer Channel’s starting logic level is programmed to the desired initial value if it is important to the particular application. The channel can be configured using the GPIO mode via the Timer_MS field and then changed to the Output Compare Mode.

The Pulse Low mode is identical to this example, except that the OCT field is programmed to 0b10 instead of 0b01.

With an IP Bus clock frequency of 66 MHz, a Prescale value of 0x0F, a Count value of 0xFF, and an OCPW field of 0x20, the Pulse High Time is 0.48 microseconds, and the waveform period is 57.6 microseconds. Changing the OCPW field to 0x40 yields a Pulse High Time of 0.96 microseconds. The waveform period is not changed. Changing the Prescale Value or Count Value in the GPT 7 Counter Input Register changes the waveform period but does not change the Pulse High Time.

5.4 Example 7 Configure Timer 0 for Input Capture Mode Using the Measure Pulse Width Function

This example is a continuation of Example 6. In Example 6, GPT 7 is programmed to Output Compare Pulse High Mode. Now, GPT 0 is programmed to measure a pulse width, using the Input Capture Mode. GPT 7 (output) is connected to GPT 0 (input).

NOTE

GPT7 is assumed to be programmed and operating as specified in Example 6 with an output high time of 60 IP Bus Clock periods.

To configure GPT 0 in the Input Capture Mode, the Timer_MS field is written to 0b001 (input capture) and the ICT field is written to 0b11 (pulse mode). The Prescaler Field in the GPT 0 Counter Input Register is programmed, in this example, to 0x0001. This causes the number of clocks between active edges on GPT 0 to be divided by 0x1. The result is placed in the OCPW Field of the GPT 0 Enable and Mode Select Register. The value written to the OCPW field is the pulse width of the pulse being measured expressed in "prescaled" IP Bus Clock counts. The following program is used to configure GPT 0.

```
200000   70630000andi.  r3,r3,0x0000;clear r3
200004   70840000andi.  r4,r4,0x0000;clear r4
200008   70A50000andi.  r5,r5,0x0000;clear r5
20000C   6463F000oris   r3,r3,0xF000
200010   60630600ori    r3,r3,0x0600
200014   70840000andi.  r4,r4,0x0000;load r4 with GPT Base Address
200018   64840004oris   r4,r4,0x0001
20001C   60840000ori    r4,r4,0x0000
200020   90830074stw    r4,116(r3);store 0x00010000 in GPT 0 Counter
                                   ;Input Register at (EA: F0000674)
                                   ;Prescale value = 0x1
200024   70840000andi.  r4,r4,0x0000
200028   64840003oris   r4,r4,0x0003
20002C   60841401ori    r4,r4,0x1401;store 0x0003 1401 to GPT 0 Enable
                                   ;and
                                   ;Mode Select Reg. This selects Input
                                   ;Capture Function and Pulse Width
                                   ;Mode
200030   90830070 stw   r4,112(r3);write GPT 0 enable and Mode
                                   ;Select
                                   ;Reg. (EA: F0000670)
200034   48000000b      0x00200034;branch forever
```

The following memory dump shows the values programmed into the registers for GPT 0 and GPT 7, respectively.

```
F0000600: GPT 0 Enable and Mode Select Register: 00030401

F0000604: GPT 0 Counter Input Register: 00010001

F0000608: GPT 0 PWM Configuration Register: 00000000

F000060C: GPT 0 Status Register:        00600001
```

NOTE

The value of 0x0060 in the GPT 0 Status Register's OCPW field reflects a measured value of 60 IP Bus Clocks for the incoming pulse.

```
F0000670: GPT 7 Enable and Mode Select Register: 60100402

F0000674: GPT 7 Counter Input Register: 000F00FF

F0000678: GPT 7 PWM Configuration Register: 00000000

F000067C: GPT 7 Status Register:        00000102
```

NOTE

The values for GPT7 reflect that the Output Pulse High time is 60 IP Bus Clocks in width, and the Output Pulse Period is 0x000F x 0x00FF IP Bus Clocks in width.

In this case, the OCPW field of the GPT 7 Enable and Mode Select Register specifies a value of 0x60 IP Bus clock counts for the GPT 7 Output High Time. The waveform period is the product of the Prescale Value and the Count Value specified in GPT7 Counter Input Register. A Prescale Value of 0x0F and a Count Value of 0xFF in the GPT7 Counter Input Register yields a pulse period of 0x0F x 0xFF IP Bus Clock Periods. With an IP Bus Clock frequency of 66 MHz, these values yield a period of 57.57 microseconds.

If the Prescale Value in GPT 0 is changed from 0x1 to 0x2, the number of "prescaled" clock counts in the GPT 0 Status Register's Capture field changes from 0x60 to 0x30 (0x60 / 0x2 = 0x30).

```
F0000600: GPT 0 Enable and Mode Select Register: 00030401

F0000604: GPT 0 Counter Input Register: 00020000

F0000608: GPT 0 PWM Configuration Register: 00000000

F000060C: GPT 0 Status Register:        00300001
```

NOTE

The value of 0x0030 in the GPT 0 Status Register's OCPW field reflects a measured value of 0x30 "prescaled" IP Bus Clocks for the incoming pulse.

```
F0000670: GPT 7 Enable and Mode Select Register: 60100402

F0000674: GPT 7 Counter Input Register: 000F00FF

F0000678: GPT 7 PWM Configuration Register: 00000000

F000067C: GPT 7 Status Register:        00003002
```

If the Prescale Value in GPT 0 is changed from 0x1 to 0x4, the number of “prescaled” clock counts in the GPT 0 Status Register’s Capture field changes from 0x60 to 0x18 (0x60 / 0x4 = 0x18).

```
F0000600: GPT 0 Enable and Mode Select Register: 00030401

F0000604: GPT 0 Counter Input Register: 00040000

F0000608: GPT 0 PWM Configuration Register: 00000000

F000060C: GPT 0 Status Register:        00180001
```

NOTE

The value of 0x0018 in the GPT 0 Status Register’s OCPW field reflects a measured value of 0x18 “prescaled” IP Bus Clocks for the incoming pulse.

```
F0000670: GPT 7 Enable and Mode Select Register: 6010040

F0000674: GPT 7 Counter Input Register: 000F00FF

F0000678: GPT 7 PWM Configuration Register: 00000000

F000067C: GPT 7 Status Register:        00006002
```

6 Conclusion

The General Purpose Timer provides eight timer channels that are software driven to do input capture, output compare, pulse width modulation signals, and other specialized functions. The granularity for the GPT is controlled by the IP Bus Clock. That is, incoming edges can be located within a window the size of one IP Bus Clock. Likewise, output edges can be located within the same window size.

Because the eight timer channels are completely independent, each timer can “prescale” the incoming IP Bus clock with different values, so that one timer can have a very fast counting rate for better accuracy, and another timer can have a very slow counting rate for increased range.

This page intentionally left blank.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN2609
Rev. 1, 08/2005

freescale™
semiconductor