DATA SHEET

GPM6P1004B
GPM6P1001B

Remote Controller with
4KB/1KB OTP

Preliminary

Jun. 25, 2015

Version 0.2

Table of Contents

PAGE

REMOTE CONTROLLER WITH 4KB/1KB OTP

1. GENERAL DESCRIPTION

This document contains device-specific information for the following devices:

- GPM6P1004B
- GPM6P1001B

The GPM6P1004B/1001B is a special chip for remote control with 64 bytes built-in SRAM and 4K/1K bytes built-in ROM. It includes two Timers and up to 12 software selectable general I/Os. Additionally, it provides one frequency programmable and duty selectable Pulse Width Modulation (PWM) output for remote control, it operates over a wide voltage range of V_{LVR} - 3.6V. It also provides two SLEEP modes, normal sleep and key scan sleep mode for power saving. Both sleep modes maintain the RAM contents, stops the oscillator and causes all other chip functions to be inoperative. The differences between both sleep modes are their wakeup methods. In normal sleep mode, it can be released by using external key change. In the key-scan sleep mode, IO will be scanned continually and be released at key press.

There is a built-in amplifier for IR transmitter. Meanwhile, users can set the different driving current by themselves and design the application circuit without the external BJT and resister. The built-in IR transfer module can make IR control and use easily.

Especially, it has a high accuracy internal OSC, which can match the specification (4MHz ± 1.5% (0°C~60°C)@ 2.0V~3.6V) and can be used in most applications.

2. FEATURES

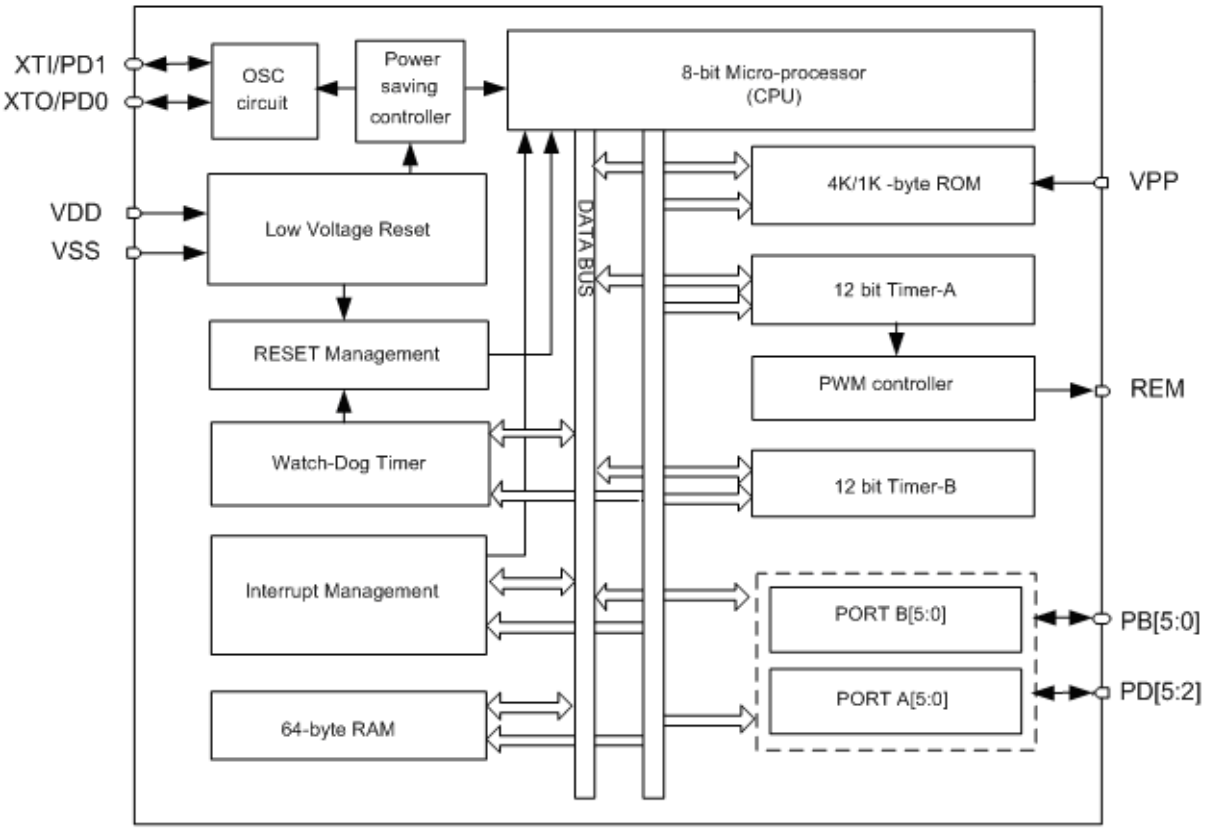
- **CPU**
  - 151 instructions
  - 13 addressing modes
  - 4MHz clock operation
- **Memories**
  - GPM6P1004B
    - 4K bytes program memory (ROM)
    - One time program capability
    - 64 bytes RAM including stack area
  - GPM6P1001B
    - 1K bytes program memory (ROM)
    - At least two times program capability
    - 64 bytes RAM including stack area
- **Reset Management**
  - Enhanced reset system
  - Power On Reset (POR)
  - Low Voltage Reset (LVR)
  - Watchdog Reset (WDR)
- **Interrupt Management**
  - Eight internal interrupts
- **I/O Ports**
  - Max. 12 multifunction bi-directional I/Os
  - Each incorporate with pull-up resistor, pull-down resistor or floating input, depending on programmer's settings on the corresponding registers
  - I/O ports with LED driving capability
  - 12 I/O ports with 16mA current sink
- **Clock Management**
  - Internal oscillator: 4MHz±1.5%(0°C~60°C), @2.0V~3.6V,
  - Crystal input: 4MHz @ 2.0V~3.6V
- **Power Management**
  - power saving modes: SLEEP mode
- **Analog Peripherals**
  - LVR：Low Voltage Reset 1.7V±0.1V
  - LVD : Low Voltage Detect
    LVDS0 = 0 , LVD = 2.0V ±0.1V
    LVDS0 = 1 , LVD = 2.4V ±0.1V
- **GPM6P1004B/1001B TimerA/B**
- **12-bit up count or 8bit down count selectable (by SFR) Timer (Timer A)**
  - –Timer mode with clock source selectable
  - –PWM output in carrier signal mode with duty and driver current programmable
  - –PWM output in no carrier signal mode with driver current programmable
- **12-bit up count or down count selectable (by SFR) Timer (Timer B)**
  - –Timer mode with clock source selectable
  - –Timer A's carry signal can be its clock source.
- **Watchdog Timer**
  - Frequency: 0.95Hz @4MHz(System Clock)
- **Key/Key-scan wake up**
  - Normal Key change wake-up from SLEEP mode
  - Key-scan & change wake-up from SLEEP mode
- **IR**
  - Built-in IR TX can drive IR LED with up to 150mA driving capability @ VDD=3.0V & V_{LED}=0.5V.

Table 2-1 GPM6P1004B /GPM6P1001B Configuration

Part NO.	ROM Type	Voltage (V)	Speed (MHz)	ROM (KByte)	RAM (Byte)	IR Tx	Timer		CPU OSC.		IO No.	PKG
							Mode0	Mode1	IOSC	XTAL		
GPM6P1004B	OTP	V_{LVR}~3.6	4	4	64	Tx	1	1	·	·	12	SOP16
	OTP	V_{LVR}~3.6	4	4	64	Tx	1	1			5	SOP8
GPM6P1001B	2TP	V_{LVR}~3.6	4	1	64	Tx	1	1	·	·	12	SOP16
	2TP	V_{LVR}~3.6	4	1	64	Tx	1	1			5	SOP8

Note: 2TP means at least two times program capacity.

3. BLOCK DIAGRAM



4. SIGNAL DESCRIPTIONS

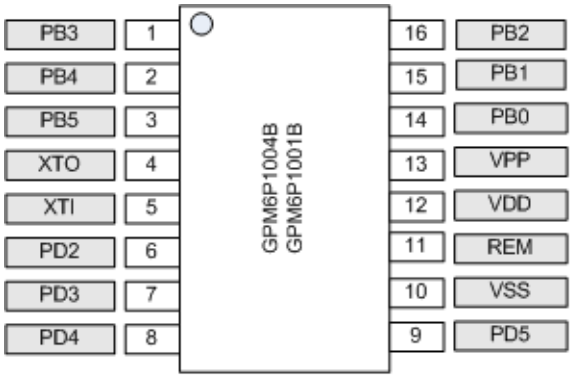
4.1. Pin Description

Type: I = Input, O = Output, S = Supply

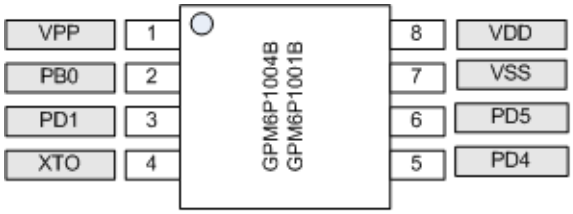
Pin Name	GPM6P1004B/1001B SOP16 pin no	GPM6P1004B/1001B SOP8 pin no	Type	Main Function	Alternate Function
PB5	3	NC	I/O	**PortB[5:0]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Normal wakeup; if a key is changed, the chip can be wakened from sleep mode. Key scan wakeup: if key scan changed is detected, the chip can be wakened up from sleep mode. **PortB[1:0]:** It can use external INT input.	
PB4	2	NC	I/O		
PB3	1	NC	I/O		
PB2	16	NC	I/O		
PB1	15	NC	I/O		
PB0	14	2	I/O		
PD5	9	6	I/O	**PortD[5:2]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Normal wakeup: N/A Key scan wakeup: if key scan changed is detected, the chip can be wakened from sleep mode.	
PD4	8	5	I/O		
PD3	7	NA	I/O		
PD2	6	NC	I/O		
XTI / PD1	5	3	I/O	**Crystal Input**: It is connected with external crystal for a crystal oscillation circuitry in crystal mode. **PortD[1]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Normal wakeup: N/A Key scan wakeup: if key scan changed is detected, the chip can be wakened from sleep mode.	
XTO / PD0	4	4	I/O	**Crystal Output**: It is connected with external crystal for crystal oscillation circuitry in crystal mode. **PortD[0]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Normal wakeup: N/A Key scan wakeup: if key scan changed is detected, the chip can be wakened from sleep mode.	
REM	11	NC	O	IR controller signal transmit pin.	
VPP	13	1	S	OTP Program power supply	
VDD	12	8	S	power supply	
VSS	10	7	S	Ground	

4.2. PIN Map (Top View)

4.2.1. SOP16 package for GPM6P1001B/1004B



4.2.2. SOP8 package for GPM6P1001B/1004B



5. FUNCTIONAL DESCRIPTIONS

5.1. Central Processing Unit

5.1.1. CPU Introduction

The CPU inside GPMP1004B/1001B is a high performance processor equipped with six internal registers: accumulator, program counter, X register, Y register, stack pointer, and processor status register. This CPU is a fully static CMOS design. The oscillation frequency could be varied up to 4.0MHz depending on the application needs.

5.1.2. CPU register

The CPU has six registers that are the Program Counter (PC), an Accumulator (A), two index registers (X, Y), the Stack Pointer (SP), and the Status register (P). The program counter consists of 16-bit register.

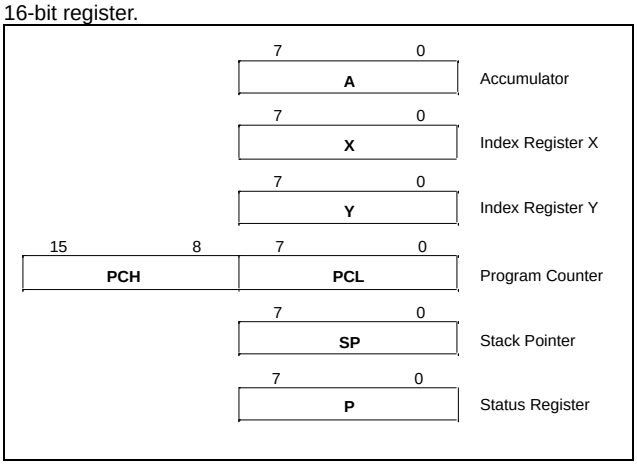


Figure 5-1 System registers

X, Y register

In address mode, X and Y registers can be used as index registers or buffer registers. These register contents are added to the specified address, which becomes the actual address. Some operations such as increment, decrement, comparison and data transfer function can be used in X and Y registers.

Accumulator

The Accumulation is the 8-bit general-purpose register, which can operate transfer, temporary saving, condition judgment, etc.

Stack pointer

The CPU has an 8-bit-wide register indicating the location in the stack to be accessed (push or pop) when a subroutine call or interrupt occurs.

When subroutine call is executed or an interrupt occurrence is accepted, the value of stack point is updated automatically.

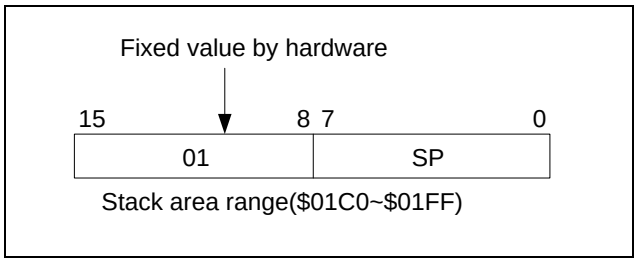


Figure 5-2 Stack point register

[Example] 5-1 Initialized stack point value

```
LDX   #C_STACK_BOTTOM    ; Initial stack pointer at $1FF
TXS                      ;Transfer to stack point
```

Program counter (PC)

The program counter is a 16-bit wide register. It consists of two 8-bit registers, PCH and PCL. This register indicates the address of next instruction to be executed. In Reset state, the content of program counter is stored with $FFFC.

Status registers (P)

The 8-bit status register contains the interrupt mask and 6 flags representative of the result of the instruction just executed. This register can also be handled by the PHP and PLP instructions. These bits can be individually controlled by specific instructions. The detailed description is shown in following description.

Note: Not all instructions affect status register. A detailed instruction description will be discussed in 6502 instruction manual.

❑ **Negative flag bit**

This flag indicates the bit7 status of the result of a data or arithmetic operation. Programmer can use this bit to perform operations, e.g. branch condition or bit operation.

❑ **Overflow flag bit**

This flag indicates whether the overflow occurs in arithmetic operation. When the result of an addition or subtraction is over +127 or less than –128, this overflow bit is set to '1'.

❑ **Decimal mode flag**

This flag indicates what mode is operated by arithmetic operation. The CPU has two operation modes, binary mode and decimal mode for arithmetic operation. Programmer can use the instruction to alternate them.

❑ **Interrupt disable flag**

This bit can enable or disable all interrupts except NMI interrupt source. If this bit is set to '1', CPU will ignore interrupt signal. On the contrary, if this bit is set to '0', CPU will accept interrupt signal.

❑ **Zero flag**

This flag indicates the result of a data or arithmetic operation. If the result is equal to zero, the zero flag is set to '1'. On the other hand, this bit is set to '0' by other values.

❑ **Carry flag**

This bit is set to "1" if the result of addition operation generates a carry, or if the result of subtraction doesn't generate a borrowing. In addition, some shift instructions or rotate instructions also change this bit.

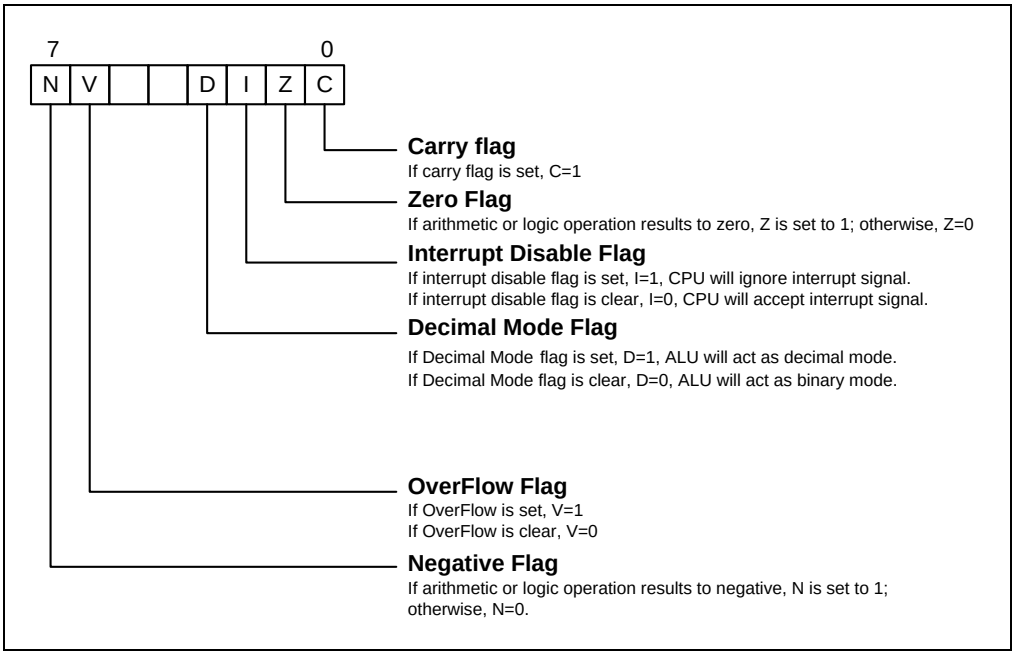


Figure 5-3 Status register

5.2. Memory Organization

5.2.1. Introduction

The GPM6P1004B/1001B separates address spaces into program memory and data memory. Program memory can be read only. It contains up to 4K/1K bytes of program memory. Data memory that contains 64 bytes of RAM including stack area can be read and written.

5.2.2. Memory Space

Memory address allocations on the GPM6P1004B are divided into several parts. Figure 5-5 shows GPM6P1004B memory map. Figure 5-6 is the GPM6P1001B memory map.

The first 128 addresses are allocated for special function registers, including function control registers and I/O control registers, which allow programmer to use the first page instruction in setting this register and help for programming size reduction.

The total RAM consists of 64 bytes (including Stack) on the locations from $00C0 through $00FF and double mapping to $01C0 ~ $01FF (see Figure 5-5).

GPM6P1004B supports 4K bytes of ROM. The address for ROM is located on $7000 ~ $7FFF double mapping to $F000 ~ $FFFF (see Figure 5-5).

GPM6P1001B supports 1K bytes of ROM. The address for ROM is located on $7C00 ~ $7FFF double mapping to $FC00 ~ $FFFF (see Figure 5-6).

The address of NMI, RESET and IRQ exception vectors are located from $FFFA to $FFFF. The exception vectors should be specified in the program for proper operation.

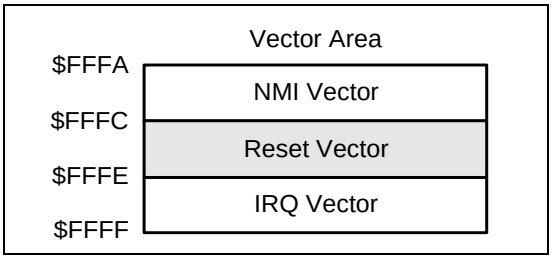


Figure 5-4 Interrupt vector area

[Example] 5-2 Interrupt vector table in software

```
VECTOR:     .SECTION
    DW  V_NMI
    DW  V_Reset
    DW  V_IRQ
```

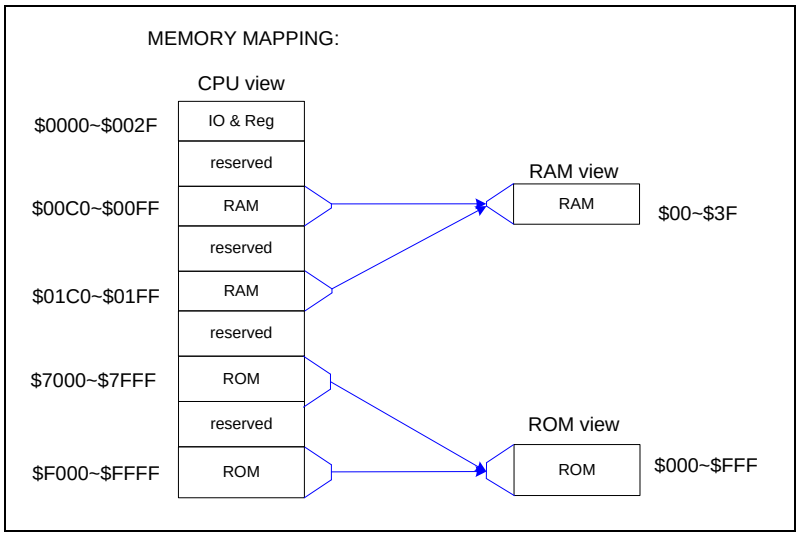


Figure 5-5 Memory map for GPM6P1004B

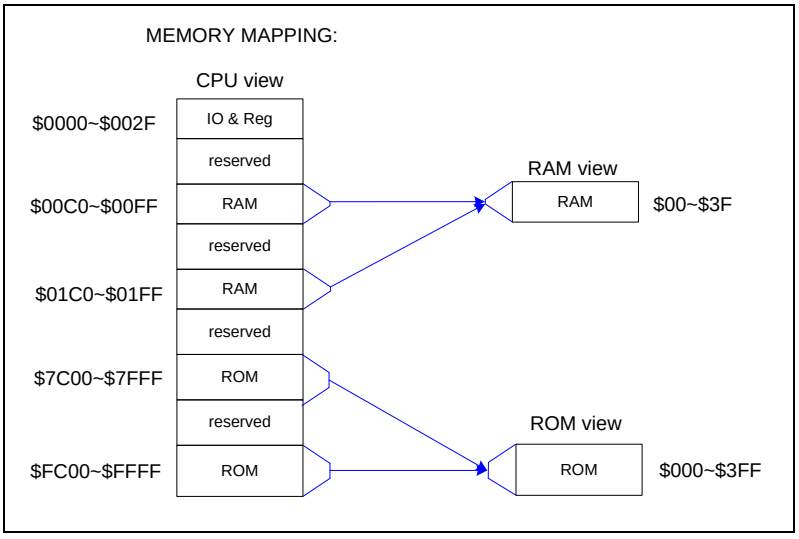


Figure 5-6 Memory map for GPM6P1001B

5.2.3. Configuration Option Register

The configuration option register is used to setup the operation condition. And its CPU view address is $FFF8.

The GPM6P1004B has the following configuration options.

- It supports crystal resonator or internal oscillator clock source.
- It supports LVR enable or disable option.
- It supports watchdog enable or disable option.
- It supports OTP ROM data protection.

Users can refer to the Device Configuration Register and set it in [Project/ Setting/ Mask Option] of Fortis IDE as Figure 5-7.

Device Configuration Register (OPCODE, $FFF8)

BIT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	OPTCHK3	OPTCHK2	OPTCHK1	OPTCHK0	Reserved	WDTENB	LVRENB	SYSCLKS
Access	R	R	R	R	R	R	R	R
Default	1	1	1	1	1	1	1	1

Bit [7:4] **OPTCHK [3:1]:** Configuration Option Check bits must be filled in 101.

OPTCHK0: disable/enable security protection. Whether to read data from OTPROM

1: Security disable

0: Security enable

Bit [3] Reserved

Bit [2] **WDTENB:** disable/enable watchdog

0= WDT is enabled

1= WDT is disabled

Bit [1] **LVRENB:** disable/enable LVR

0= LVR is enabled

1= LVR is disabled

Bit [0] **SYSCLKS:** IOSC (internal) / Crystal selection

0= IOSC

1= Crystal

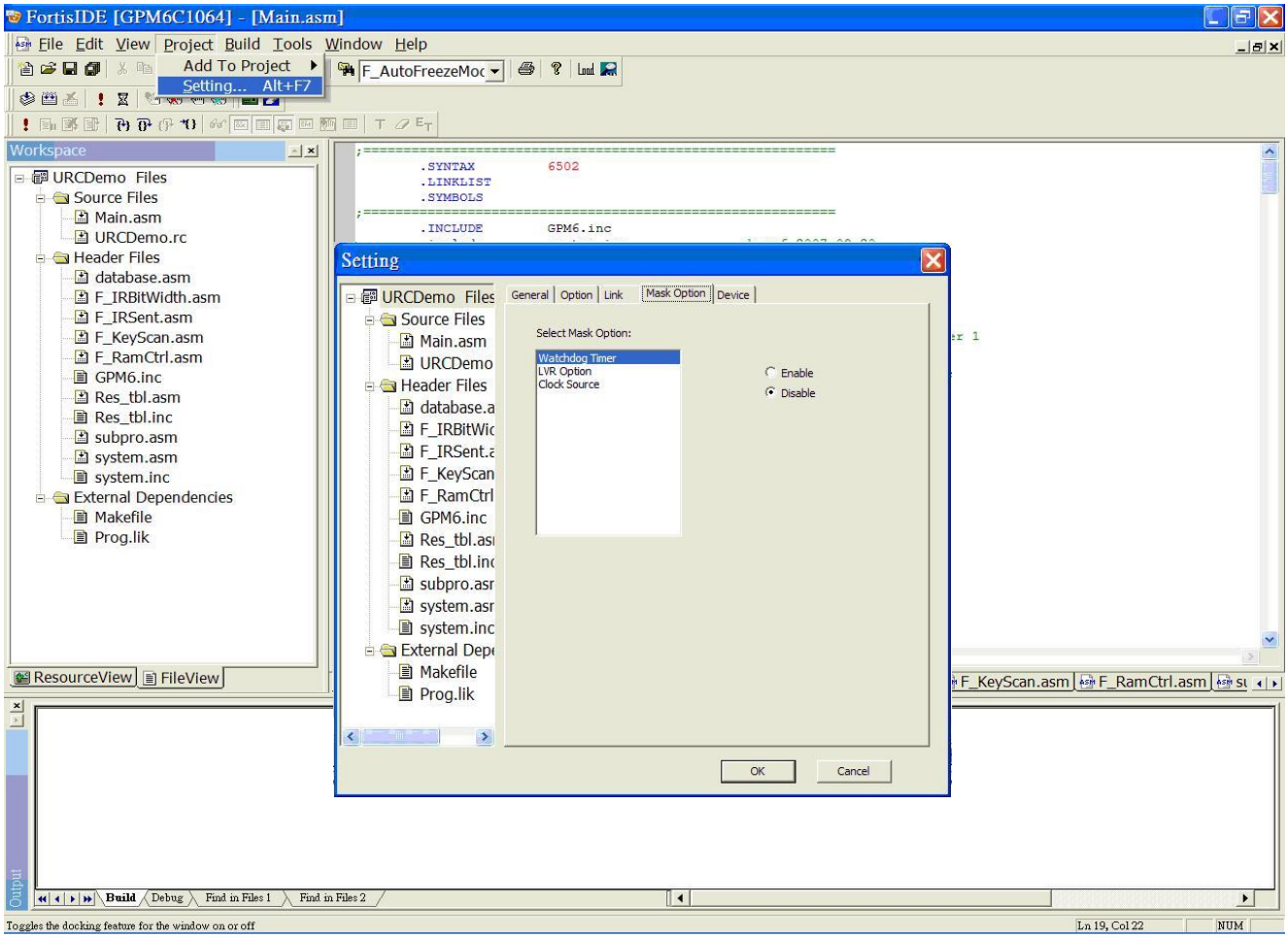


Figure 5-7 Device Configuration Register set in Fortis IDE

5.2.4. Special function Registers

GPM6P1004B/1001B device features up to nineteen control registers. All of the control registers are used by MCU and peripheral function block for needed operation controls of the device. Some of the control registers contain control and status bits for peripheral module such as Timer unit, Interrupt control unit, etc. Note that the reserved addresses are not implemented on the chip. Some of bits in control register are read only. When writing to them, there is no any effect on the corresponding bits. The following table shows the summary of the control registers. The detailed information of each control registers are explained in each peripheral section.

GPM6P1004B/1001B Special Function Registers Description

$0000~000A: I/O port

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$00	P_IOB_DIR	00h	R/W	R/0	R/0	Port B direction control					
$02	P_IOD_DIR	00h	R/W	R/0	R/0	Port D direction control					
$04	P_IOB_ATT	00h	R/W	R/0	R/0	Port B attribute register					
$06	P_IOD_ATT	00h	R/W	R/0	R/0	Port D attribute register					
$08	P_IOB_DAT	00h	R/W	R/0	R/0	Write data into the Port B data register and read data from the I/O pad.					
$0A	P_IOD_DAT	00h	R/W	R/0	R/0	Write data into the Port D data register and read data from the I/O pad.					

$0010~$0016: INT Flag & other special register

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$11	P_PWM_DRV	00h	R/W	R/0	R/0	R/0	R/0	PWMDRV0	R/0	R/0	R/0
$12	P_SYS_SLEEP	00h	W	C_SYS_SLEEP= AAH (Write other data to reset the system.)							
$13	P_INT_CTRL	00h	R/W	R/0	TMAOIE	R/0	TMBOIE	F1KIE	F4KIE	F32KIE	F2MIE
$14	P_INT_FLAG	00h	R/W	R/0	TMAOIF	R/0	TMBOIF	F1KIF	F4KIF	F32KIF	F2MIF
$17	P_TIM_SEL	00h	R/W	TIMSET	IRENB	R/0	R/0	R/0	R/0	R/0	R/0
$18	P_INT_IO	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	PBINT1	PBINT0
$19	P_INT_ FLAGIO	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	PBINTF1	PBINTF0
$1B	P_SC_IOB	00h	R/W	R/0	R/0	PBSE5	PBSE4	PBSE3	PBSE2	PBSE1	PBSE0
$1D	P_SC_IOD	00h	R/W	R/0	R/0	PDSE5	PDSE4	PDSE3	PDSE2	PDSE1	PDSE0

$0020~0026: Timer control

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$20	P_WDT_CTRL	00h	W	C_WDT_CLR= AAH (Write other data to reset the system.)							
$21	P_TMA_CTRL	00h	R/W	TMAES	R/0	TMACLK1	TMACLK0	TMADUT1	TMADUT0	R/0	TMAMOD0
$22	P_TMB_CTRL	00h	R/W	TMBES	R/0	TMBCLK1	TMBCLK0	R/0	R/0	R/0	R/0
$23	P_TMA_CNTL	00h	R	TMA Counter Low Byte 8-bit Pre-value.							
	P_TMA_PWML		W	TMA PWM Low Byte 8-bit Period Value.							
$24	P_TMA_CNTH	00h	R	R/0	R/0	R/0	R/0	TMA Counter High Byte 4-bit Pre-value.			
	P_TMA_PWMH		W	-	-	-	-	TMA PWM High Byte 4-bit Period Value.			
$25	P_TMB_CNTL	00h	R	TMB Counter Low Byte 8-bit Pre-value.							
	P_TMB_REGL		W	TMB Low Byte 8-bit Register							
$26	P_TMB_CNTH	00h	R	R/0	R/0	R/0	R/0	TMB Counter High Byte 4-bit Pre-value.			

	P_TMB_REGH		W	-	-	-	-	TMB High Byte 4-bit Register			
$33	P_LVD_CTRL	00h	R	LVDEN	R/0	LVDS0	PD0PWMS	-	R/0	R/0	LVD
			W	LVDEN	-	LVDS0	PD0PWMS	-	-	-	-

Note:

1. “R/0” means read data value always is “0”.
2. If the bits of the register are not defined, they will not be implemented in the real chip, and their readout value should be random. However, they are defined as 0 in this table for simplification.

5.3. Clock Source

The GPM6P1004B/1001B supports Crystal / Ceramic or Internal oscillator, as shown in the following diagram, Figure 5-8. They can be selected by device configuration option at address ($FFF8.0) and be set in Fortis IDE as indicated in Figure 5-7.

The detailed configuration option setting of device is given in section 5.2.3 configuration option register.

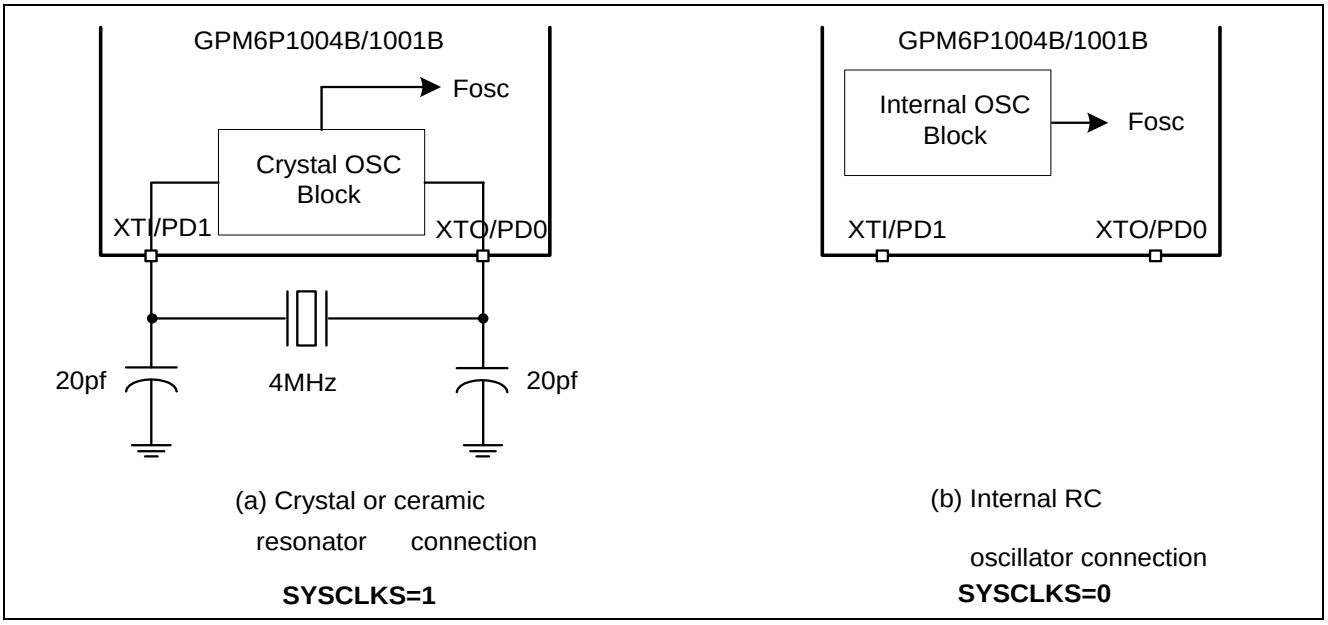


Figure 5-8 Two types of clock sources

5.4. Power Saving Mode

5.4.1. Introduction

To reduce the current consumption when system does not need to be active, SLEEP mode (normal sleep or key scan sleep) can be applied when system is not in active. These two modes are able to reduce power consumption to save more power. They also feature different wakeup time. Programmer must write the SLEEP Control Register with corresponding value to enter SLEEP mode. For more information about SLEEP mode, please refer to Figure 5-9 and they will be depicted in the next two sections.

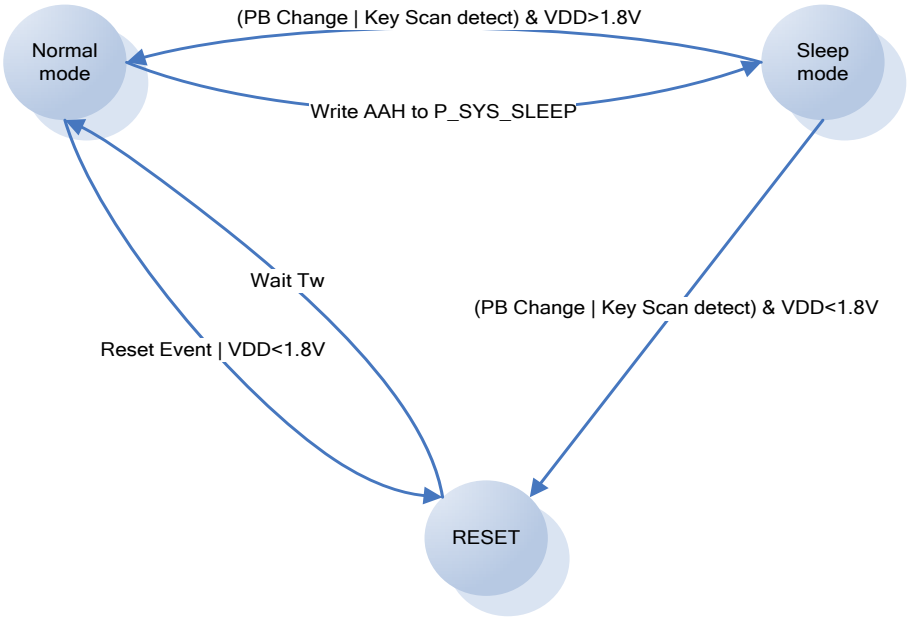


Figure 5-9 Power saving mode operation

5.4.2. SLEEP Mode

The SLEEP mode function will disable all system clocks, including the clock generation circuit. Once the system enters the SLEEP mode, two methods can be used to detect the key change. If key scan bits control registers (P_SC_IOB, P_SC_IOD) are disabled and normal wakeup is selected, the activated PortB change wakeup events (from I/Os) can recover the normal operation from SLEEP mode. If any key scan bits control registers (P_SC_IOB, P_SC_IOD) are enabled, key-scan wakeup is selected and the I/Os are connected as Figure 5-10, any key press detected can recover the normal operation from SLEEP mode.

In such mode, LVR function is disabled, RAM and I/Os will remain in their previous states until being awakened. The system will be wakened up by any change on port B or key press detection. After the GPM6P1004/1001B is awakened, the internal CPU will remain on previous State until $T_W \geqq 2048 \times T1$ (T_W = waiting time & T1 = system clock cycle); and then continue processing the program (see Figure 5-11).

$$T1 = 1 / (F_{CPU}),\ T_W \geqq 2048 \times T1$$

Entering SLEEP mode, programmer must write #C_SYS_SLEEP ($AA) to SLEEP control register (P_SYS_SLEEP).

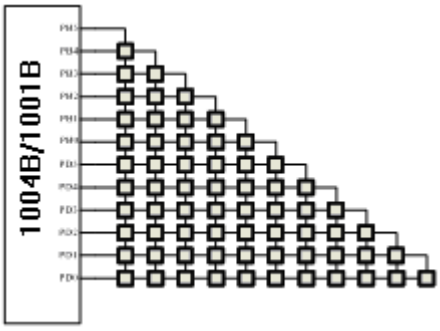


Figure 5-10 Key scan mode more key wakeup support

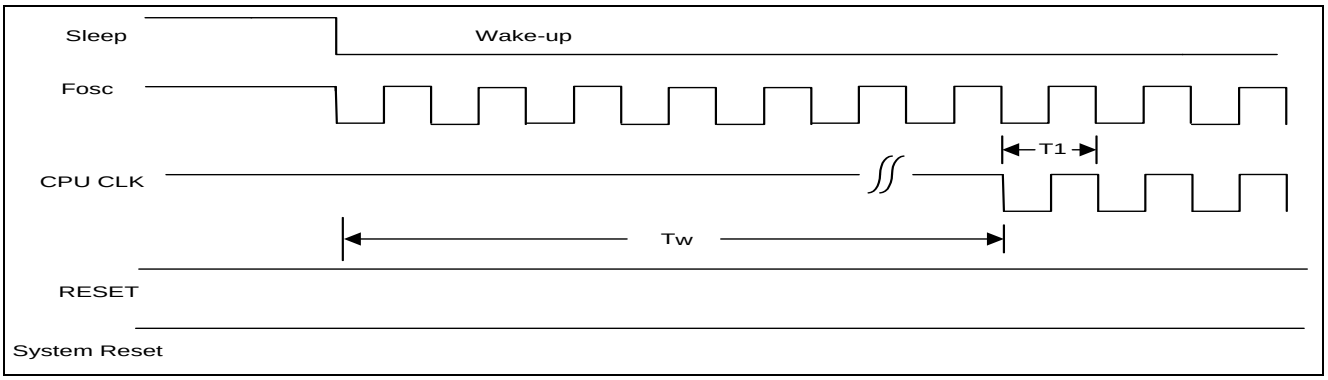


Figure 5-11 SLEEP mode

PB Key Scan Control Register (P_SC_IOB, $001B)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PBSE5	PBSE4	PBSE3	PBSE2	PBSE1	PBSE0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:6] Reserved

Bit [5:0] **PBSE** [5:0]: PortB key scan enabled bits

1 = enable, PB.x will be scanned under sleep mode.(x=0~5)

0 = disable, PB.x will act as normal IO (x=0~5)

PD Key Scan Control Register (P_SC_IOD, $001D)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PDSE5	PDSE4	PDSE3	PDSE2	PDSE1	PDSE0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:6] Reserved

Bit [5:0] **PDSE** [5:0]: PortD key scan enabled bits

1 = enable, PD.x will be scanned under sleep mode.(x=0~5)

0 = disable, PD.x will act as normal IO (x=0~5)

SLEEP Control Register (P_SYS_SLEEP, $0012)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	SLEEPCTRL7	SLEEPCTRL6	SLEEPCTRL5	SLEEPCTRL4	SLEEPCTRL3	SLEEPCTRL2	SLEEPCTRL1	SLEEPCTRL0
ACCESS	W	W	W	W	W	W	W	W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **SLEEPCTRL** [7:0]: Operation mode control register

$AA = write to enter SLEEP mode (C_SYS_SLEEP)

Other data = reset system

[Example] 5-3 Let MCU enter SLEEP mode

```
LDA     P_IOB_DAT                ; latch Port B
LDA     #C_SYS_SLEEP             ; SLEEP command $AA
STA     P_SYS_ SLEEP             ; go to sleep mode
NOP
NOP                              ; two temp command to avoid real command loss
```

[Example] 5-4 Let MCU enter Key Scan SLEEP mode

```
LDA     #FFH
STA     P_SC_IOB                ; Enable IOB key scan under sleep
STA     P_SC_IOD                ; Enable IOD key scan under sleep
LDA     #C_SYS_SLEEP            ; SLEEP command $AA
STA     P_SYS_ SLEEP            ; go to sleep mode
NOP
NOP                             ; two temp command to avoid real command loss
```

5.5. Interrupt

5.5.1. Introduction

The GPM6P1004B/1001B features eight types of interrupt sources with the same normal interrupt level and they are Timer A overflow interrupt, Timer B overflow interrupt, Fosc/1024 interrupt, Fosc/4096 interrupt, Fosc/32768 interrupt, port B INT1 interrupt, port B INT0 interrupt and Fosc/2097152 interrupt.

These interrupts have individual status (occur or not occur) and control (enable or not enable) registers. In general, once an interrupt event occurs, the corresponding flag bit will be set. If the related interrupt control bit is set to enable interrupt, an interrupt request signal will be generated and CPU will execute service routine. If the related interrupt control bit is disabled, programmer still can observe the corresponding flag bit, but no interrupt request signal will be generated. The interrupt flag bits must be cleared in the interrupt service routine to prevent program from deadlock in interrupt service routine. With any instruction, interrupts pending during the previous instruction is served.

Before entering interrupt service routine, the system saves the current PC address into bottom of the stack such as address $1FF and $1FE in Figure 5-12. And abstract the interrupt service routine first address from $FFFE and $FFFF. In a corresponding way, the system abstracts the returned PC address from the bottom of the stack when finishing the interrupt service (See Figure 5-13). These interrupt sources are listed as Table 5-1 and will be described in corresponding section.

Table 5-1 Interrupt source list

Source	Interrupt flag register	Interrupt control register	Source	Interrupt flag register	Interrupt control register
Timer A overflow	TMAOIF($0014.6)	TMAOIE($0013.6)	Time Fosc/32768	F32KIF($0014.1)	F32KIE($0013.1)
Timer B overflow	TMBOIF($0014.4)	TMBOIE($0013.4)	Time Fosc/2097152	F2MIF($0014.0)	F2MIE($0013.0)
Time Fosc/1024	F1KIF($0014.3)	F1KIE($0013.3)	PBINT1	PBINT1($0019.1)	PBINTF1($0018.1)
Time Fosc/4096	F4KIF($0014.2)	F4KIE($0013.2)	PBINT0	PBINT0($0019.0)	PBINTF0($0018.0)

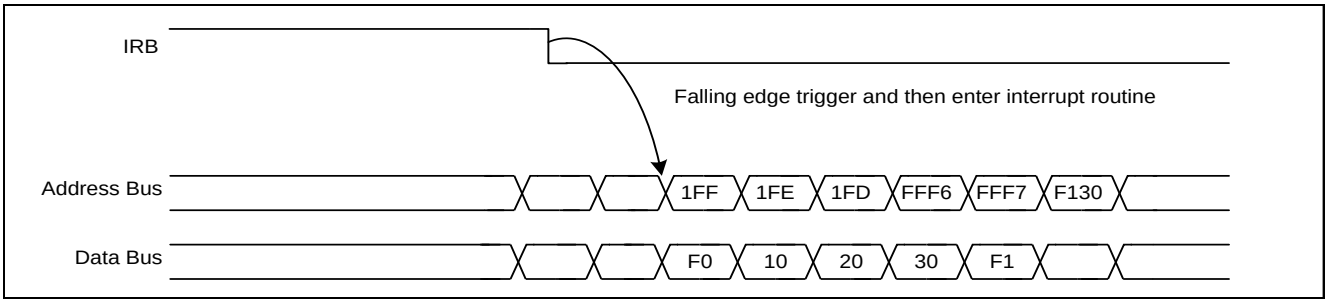


Figure 5-12 Interrupt triggered by IRB

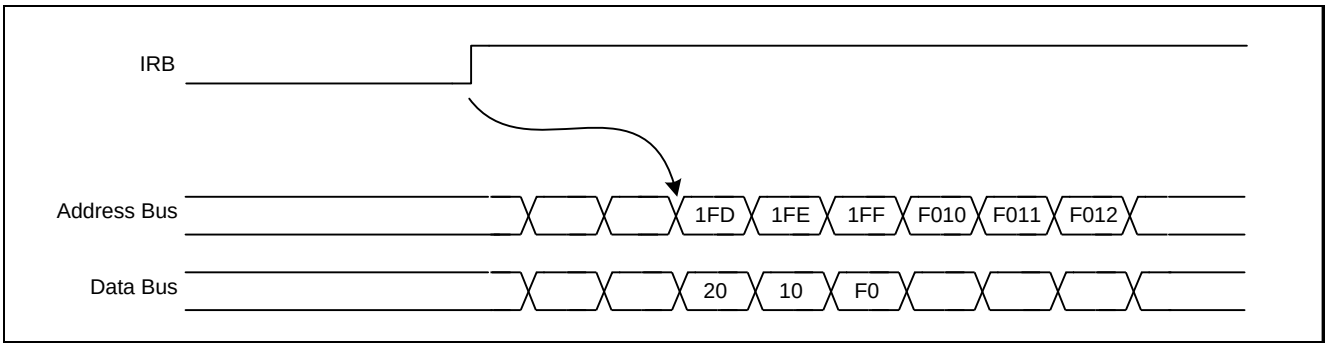


Figure 5-13 Leave interrupt routine

5.5.2. Interrupt register

Interrupt Control Register (P_INT_CTRL, $0013)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	TMAOIE	-	TMBOIE	F1KIE	F4KIE	F32KIE	F2MIE
ACCESS	-	R/W	-	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	0	-	0	0	0	0	0

Bit [7] Reserved

Bit [6] **TMAOIE**: Timer A overflow interrupt enable bit
0 = interrupt disable
1 = interrupt enable (C_INT_TMAOIE)

Bit [5] Reserved

Bit [4] **TMBOIE**: Timer B overflow interrupt enable bit
0 = interrupt disable
1 = interrupt enable (C_INT_TMBOIE)

Bit [3] **F1KIE**: Time Fosc/1024 interrupt enable bit
0 = interrupt disable
1 = interrupt enable (C_INT_F1KIE)

Bit [2] **F4KIE**: Time Fosc/4096 interrupt enable bit
0 = interrupt disable
1 = interrupt enable (C_INT_F4KIE)

Bit [1] **F32KIE**: Time Fosc/32768 interrupt enable bit
0 = interrupt disable
1 = interrupt enable (C_INT_F32KIE)

Bit [0] **F2MIE**: Time Fosc/2097152 interrupt enable bit
0 = interrupt disable
1 = interrupt enable (C_INT_F2MIE)

Interrupt Flag Register (P_INT_FLAG, $0014)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	TMAOIF	-	TMBOIF	F1KIF	F4KIF	F32KIF	F2MIF
ACCESS	-	R/W	-	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	0	-	0	0	0	0	0

This flag is cleared by writing the corresponding bit by “1”.

Bit [7] Reserved

Bit [6] **TMAOIF**: Timer A overflow interrupt flag
0 = no event
1 = event occurs (C_INT_TMAOIF).

Bit [5] Reserved

Bit [4] **TMBOIF**: Timer B overflow interrupt flag
0 = no event
1 = event has occurred (C_INT_TMBOIF).

Bit [3] **F1KIF**: Time Fosc/1024 interrupt flag
0 = no event
1 = event occurs (C_INT_F1KIF).

Bit [2] **F4KIF**: Time Fosc/4096 interrupt flag
0 = no event
1 = event has occurred (C_INT_F4KIF).

Bit [1] **F32KIF**: Time Fosc/32768 interrupt flag
0 = no event
1 = event has occurred (C_INT_F32KIF).

Bit [0] **F2MIF**: Time Fosc/2097152 interrupt flag
0 = no event
1 = event has occurred (C_INT_F2MIF).

[Example] 5-5 Enable Timer A overflow interrupt

```
;=========================;
; main loop routine
;=========================;
lda     #C_INT_TMAOIE
sta      P_INT_CTRL                    ; enable Timer A overflow INT
cli                                    ; enable INT
;=========================;
;IRQ interrupt service routine
;=========================;
lda     #C_INT_TMAOIF
sta     P_INT_FLAG                     ; clear INT request flag
sta     P_INT_CTRL                     ; enable Timer A overflow INT
```

IO Port Interrupt Flag Register (P_INT_FLAGIO, $0019)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	R/0	R/0	R/0	PBINTF1	PBINTF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

This flag is cleared by writing the corresponding bit by “1”.

Bit [1] **PBINTF1:** port B1 interrupt flag.
0 = no event
1 = event has occurred

Bit [0] **PBINTF0:** port B0 interrupt flag.
0 = no event
1 = event has occurred

IO Port interrupt Control Register (P_INT_IO, $0018)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	R/0	R/0	R/0	PBINT1	PBINT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit [1] **PBINT1:** port B1 interrupt
0 = interrupt disable
1 = interrupt enable

Bit [0] **PBINT0 :** port B0 interrupt
0 = interrupt disable
1 = interrupt enable

5.6. Reset Sources

5.6.1. Introduction

There are three types of reset sources for the system, Power-On Reset (POR), Low Voltage Reset (LVR), Watchdog Timer Reset (WDR). These reset sources can be concluded as external events and internal events. The external events come from power line or external trigger event. The internal events come from the program run away. Figure 5-14 shows the affected region for each reset source.

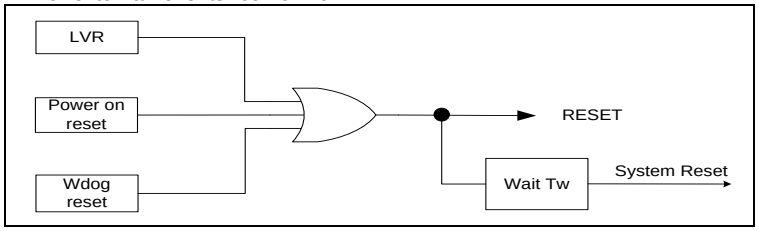


Figure 5-14 Reset sources

5.6.2. Power-On Reset (POR)

A POR is generated when VDD is rising from 0v. When VDD rises to an acceptable level (~1.45V), the power on reset circuit will start a power-on sequence. After that, the system will operate in target speed and start to activate.

5.6.3. Low Voltage Reset (LVR)

The on-chip Low Voltage Reset (LVR) circuitry forces the system entering reset status when the MCU voltage falls below the specific LVR trigger voltage. This function prevents MCU from working at an invalid operating voltage range.

A device configuration option at address ($FFF8.1, can be set in Fortis IDE as Figure 5-7) is used to enable or disable this function. If this function is enabled, the LVR circuit will monitor power level while chip is operating. If power is lower than the specific level for a specific period of time, the system will reset.

5.6.4. Watchdog Timer Reset (WDR)

On-chip watchdog circuitry makes the device entering reset when MCU goes into an unknown state and has no watchdog cleared information. This function is able to prevent the MCU away from abnormal conditions. The WDT can be disabled or enabled through configuration option address ($FFF8.2, can be set in Fortis IDE as Figure 5-7). The internal reset of WDT will be generated by a time-out event of the WDT automatically when watchdog is enabled.

These reset signals will reset the CPU and restart the program. To avoid a WDT time-out reset, programmer has to write # C_WDT_CLR (=$AA) to P_WDT_CTRL periodically. If a reset signal is generated, it will also clear the WDT counter and restart the WDT.

Watchdog Control Register (P_WDT_CTRL, $0020)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	WDTCTRL7	WDTCTRL6	WDTCTRL5	WDTCTRL4	WDTCTRL3	WDTCTRL2	WDTCTRL1	WDTCTRL0
ACCESS	W	W	W	W	W	W	W	W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **WDTCTRL [7:0]**: Operation mode control register

$AA = write to clear watch dog counter (C_WDT_CLR)

Other data = reset system

[Example] 5-6 Clear watch dog counter

```
lda     # C_WDT_CLR                    ; Clear watchdog command $AA
sta     P_WDT_ CTRL
```

5.7. I/O PORTS

5.7.1. Introduction

The GPM6P1004B/1001B has two ports: Port B and Port D. These port pins may be multiplexed with an alternate function for the peripheral feature on the device. In general, when an initial reset state occurs, all ports are used as a general purpose input port. The IO structures contain three major parts: data, direction and attribution registers. Each corresponding bit in these ports should be given a value.

The setting rules are as follows:

- The direction setting determines whether this pin is an input or an output.
- The data register is used to read the value on the port, which is different when programmer sets the port to input pull-high/ pull-low.

Please refer to the Table 5-2 for IOB and IOD settings.

Table 5-2 I/O configurations (for IOB, IOC and IOD)

Attribution (P_IOX_ATT)	Direction (P_IOX_DIR)	Data (P_IOX_DAT)	Function	Description
0	0	0	Floating	Input with float
0	0	1	Pull low	Input with pull-low
0	1	0	Driving low	Output Data
0	1	1	Driving High	Output Data
1	0	0	Floating	Input with float
1	0	1	Pull high	Input with pull-high
1	1	0	Driving High	Output Data
1	1	1	Driving low	Output Data

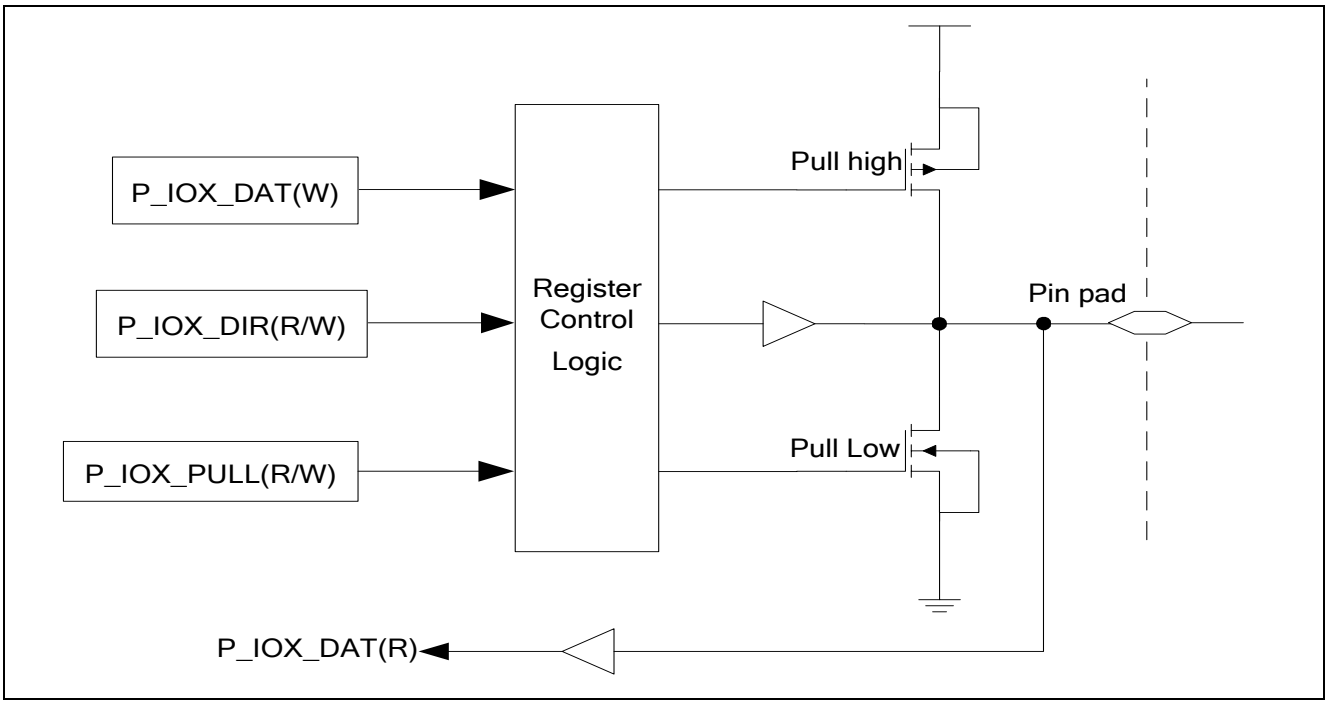


Figure 5-15 Block diagram of I/O port (IOB, IOC and IOD)

5.7.2. Port B

Port B is a 6-bit bi-directional I/O port. The I/O Port B has 6 programmable I/Os, controlled by direction control register P_IOB_DIR, and attribution register P_IOB_ATT. Reading P_IOB_DAT will get the real IO value.

Port B Direction Register (P_IOB_DIR, $0000)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	P_IOB_DIR					
ACCESS	-	-	R/W					
DEFAULT	00h							

Bit [7:0] **P_IOB_DIR**: Port B direction register.

0 = input

1 = output

Port B Attribution Register (P_IOB_ATT, $0004)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	P_IOB_ATT					
ACCESS	-	-	R/W					
DEFAULT	00h							

Bit [7:0] **P_IOB_ATT**: Port B attribution register

Port B Data Register (P_IOB_ DAT, $0008)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	P_IOB_ DAT					
ACCESS	-	-	R/W					
DEFAULT	00h							

Bit [7:0] **P_IOB_ DAT**: Port B Data value.
Read to get Port B value
Write to configure Port B register.

[Example] 5-7 Set Port B [3:0] as output with low data and Port B [7:4] as input with pulling low.

```
LDA   #$0F          ; store accumulator with $0F
STA   P_IOB_DIR     ; set direction
LDA   #$00          ; store accumulator with $00
STA   P_IOB_ATT     ; set attribute
LDA   #$F0          ; store accumulator with $F0
STA   P_IOB_DAT     ; set Port Data
```

[Example] 5-8 Set Port B [5:0] as input with floating.

```
LDA   #$00          ; store accumulator with $00
STA   P_IOB_ATT     ; set direction
STA   P_IOB_DIR     ; set attribute
STA   P_IOB_DAT     ; set Port Data
```

5.7.3. Port D

Port D is a 6-bit bi-directional I/O port. The I/O Port D has 6 programmable I/Os, controlled by direction control register P_IOD_DIR, and attribution register P_IOD_ATT. Reading P_IOD_DAT will get the real IO value. In addition, Port D is multiplexed with various special functions. After reset, the default setting for port D is used as general I/O ports.

Table 5-3 Port D function list

Port D Pin	BIT	Shared function
PD0	Bit0	Crystal output (XTO)
PD1	Bit1	Crystal input (XTI)

Port D Direction Register (P_IOD_DIR, $0002)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	P_IOD_DIR					
ACCESS	-	-	R/W					
DEFAULT	00h							

Bit [7:6] Reserved
Bit [5:0] **P_IOD_DIR**: Port D direction register.
0 = input
1 = output

Port D Attribution Register (P_IOD_ATT, $0006)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	P_IOD_ATT					
ACCESS	-	-	R/W					
DEFAULT	00h							

Bit [7:6] Reserved
Bit [5:0] **P_IOD_ATT**: Port D attribution register

Port D Data Register (P_IOD_ DAT, $000A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	P_IOD_ DAT					
ACCESS	-	-	R/W					
DEFAULT	00h							

Bit [7:6] Reserved

Bit [5:0] **P_IOD_ DAT**: Port D Data value.

Read to get Port D value

Write to configure Port D register

[Example] 5-9 Set Port D[1:0] as output with low data.

```
LDA   #$03          ; store accumulator with $03
STA   P_IOD_DIR     ; set direction
LDA   #$00          ; store accumulator with $00
STA   P_IOD_ATT     ; set attribute
STA   P_IOD_DAT     ; set port data
```

5.8. Timer Module

5.8.1. Introduction

GPM6P1004B/1001B has two timers, Timer A and Timer B respectively. Timer A and timer B can be set as mode 0 (12-bit up count and 12-bit up count) or mode 1 (8-bit down count and 12-bit down count) timer by configure the timer select register (P_TIM_SEL.7). Timer A contains one powerful PWM function and is controlled by corresponding control registers. This function can be easily configured. Each timer's function summary is shown as [Table] 5-1.

[Table] 5-1 Summary of timer function for GPM6P1004B/1001B

	Timer Counter	PWM
Timer A	YES	YES
Timer B	YES	None

5.8.2. Mode 0 Timer A (12-bit up count timer)

When Timer A is selected as 12-bit up count timer via configuring the corresponding bits of the control register (P_TIM_SEL[7]), the Timer A is special for generating carrier signal in IR control application. The timer A's input clock is selectable (Fosc/1, Fosc/4, Fosc/16), which can be configured by control register P_TMA_CTRL[5:4]. Timer A provides with two PWM modes, and the PWM signal is send to IR TX (REM) pin. The driver current of these two kinds of PWM are programmable by configuring TX PWM driving current control source register (P_PWM_DRV [3]).

GPM6P1004B/1001B 12-bit up count Timer A module has all the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$FFF to #$000
- Supports PWM with carrier signal mode
- Supports PWM without carrier signal mode

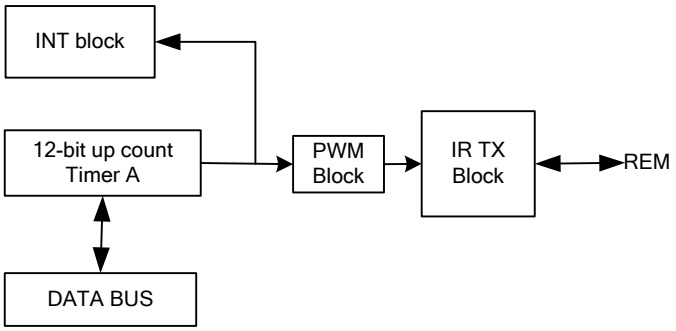


Figure 5-16 GPM6P1004B/1001B 12-bit up count Timer A block diagram

5.8.2.1. Mode 0 Timer A PWM with carrier signal mode

GPM6P1004B/1001B Timer A can be configured as PWM mode for generating carrier signal. In PWM with carrier signal mode, the 12-bit timer is an up counter with input clock selectable (Fosc/1, Fosc/4, Fosc/16). When Timer A is started, the value of 4-bit high-byte (low-nibble) register and 8-bit low-byte register will firstly be loaded into the 12-bit counter and then the counter starts count up from the loaded value. If an overflow occurs, the value of high-byte (low-nibble) register (P_TMA_CNTH) and low-byte register (P_TMA_CNTL) will be reloaded into the counter automatically and the counter starts count up again. So the carrier signal with frequency programmable can be generated by this PWM mode via configuring these two registers. Also users can select PWM duty cycle (1/3, 1/4, 1/5, 1/2) via configuring the corresponding bits of the control register (P_TMA_CTRL[3:2]). The carrier signal's enabled or disabled bit can be controlled by two methods depended on which clock source is selected by timer B. If timer B is selected one of the first three clock source (Fosc, Fosc/4 or Fosc/64) by P_TMB_CTRL [5:4] (TMBCLK [1:0]), the timer A's carrier signal on/ off is controlled by timer A's enable/ disable control bit (TMAES) directly.

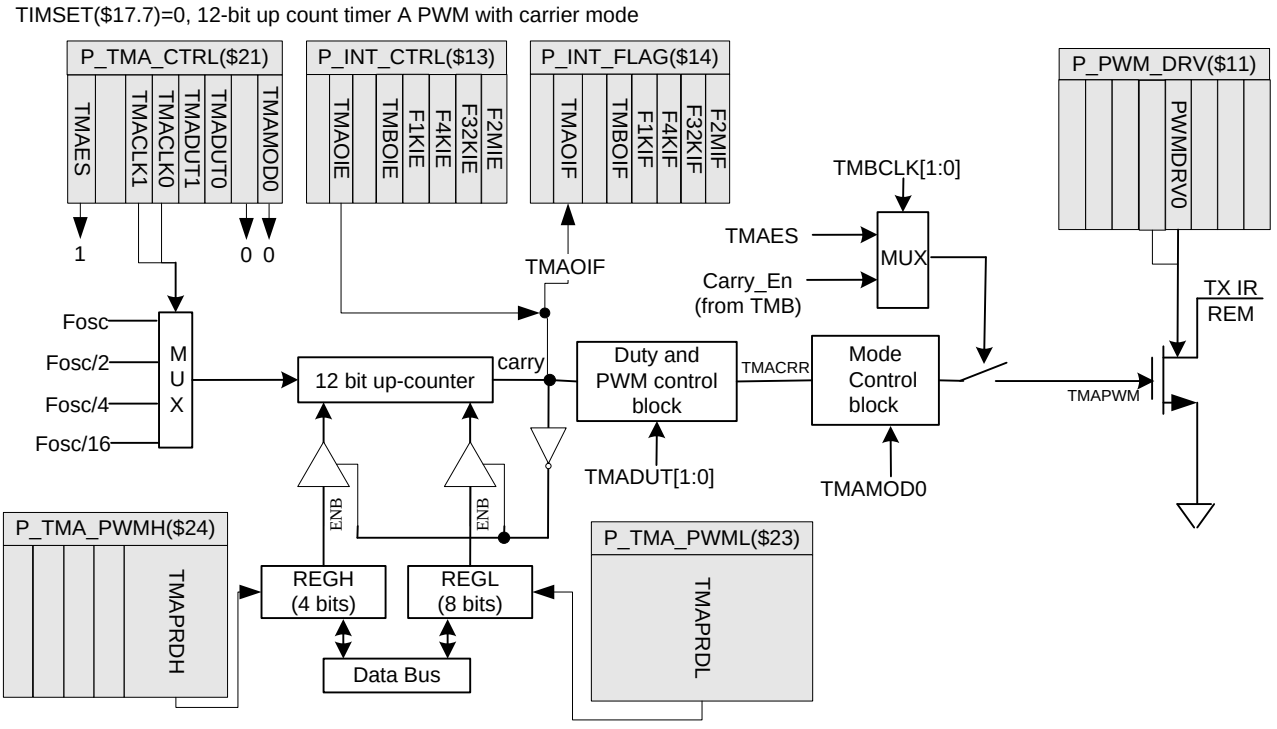


Figure 5-17 Mode 0 Timer A PWM mode diagram

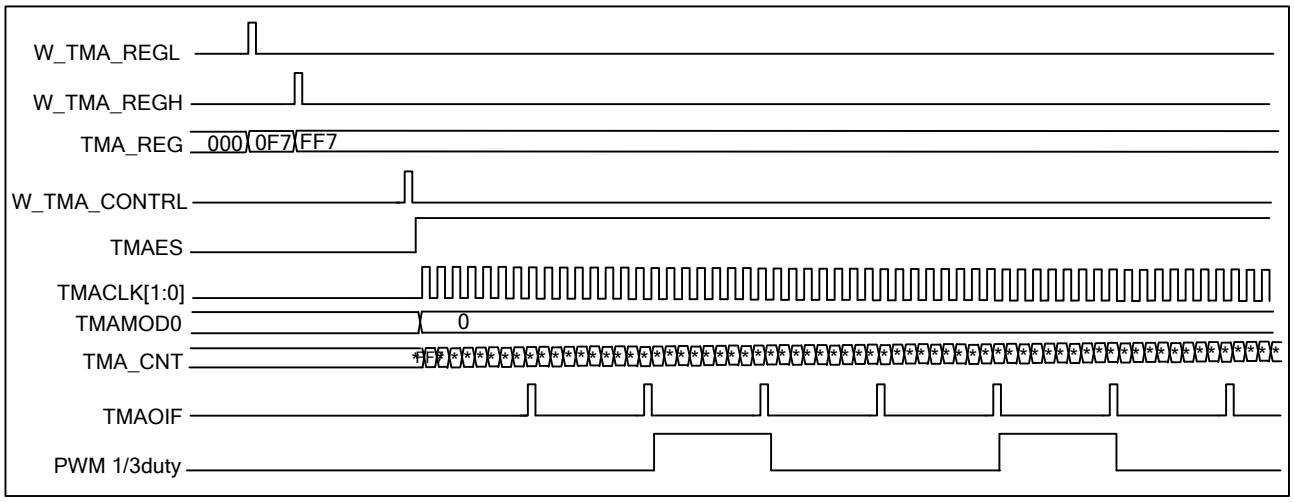


Figure 5-18 Mode 0 Timer A Normal PWM generation without envelop

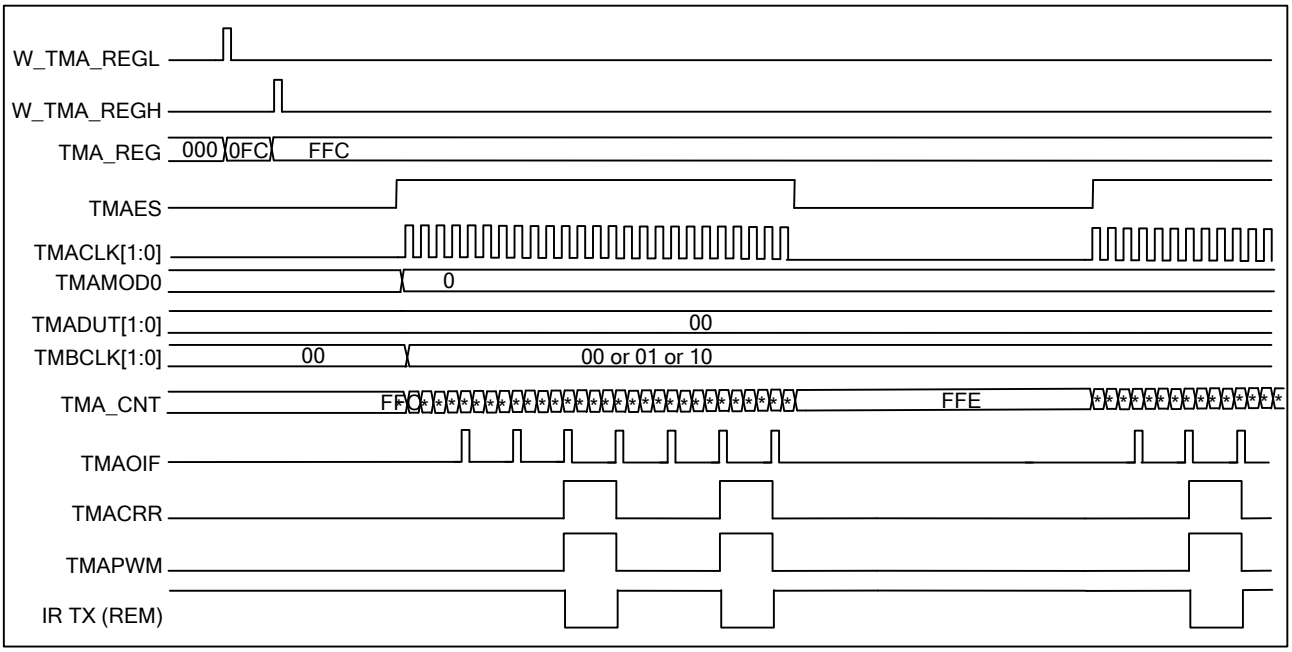


Figure 5-19 The Waveform of Mode 0 Timer A PWM with carrier signal mode (1/3 duty, on/off control by TMAES)

Another method to generate envelope PWM signal is that Timer A and Timer B must be used together. Timer A must first generate carry clock, which is same as normal PWM generation. Then enable Timer B and select Timer A carrier signal as its input clock. And Timer B register must be written in the right data, which represents the carry number. When TMBOVF occurs, another value must be written into Timer B register, which represents the no carry clock number. Envelop with carrier is on or off only when Timer B overflow events occur one by one. Then, the envelop PWM signal will be generated at REM port at last.

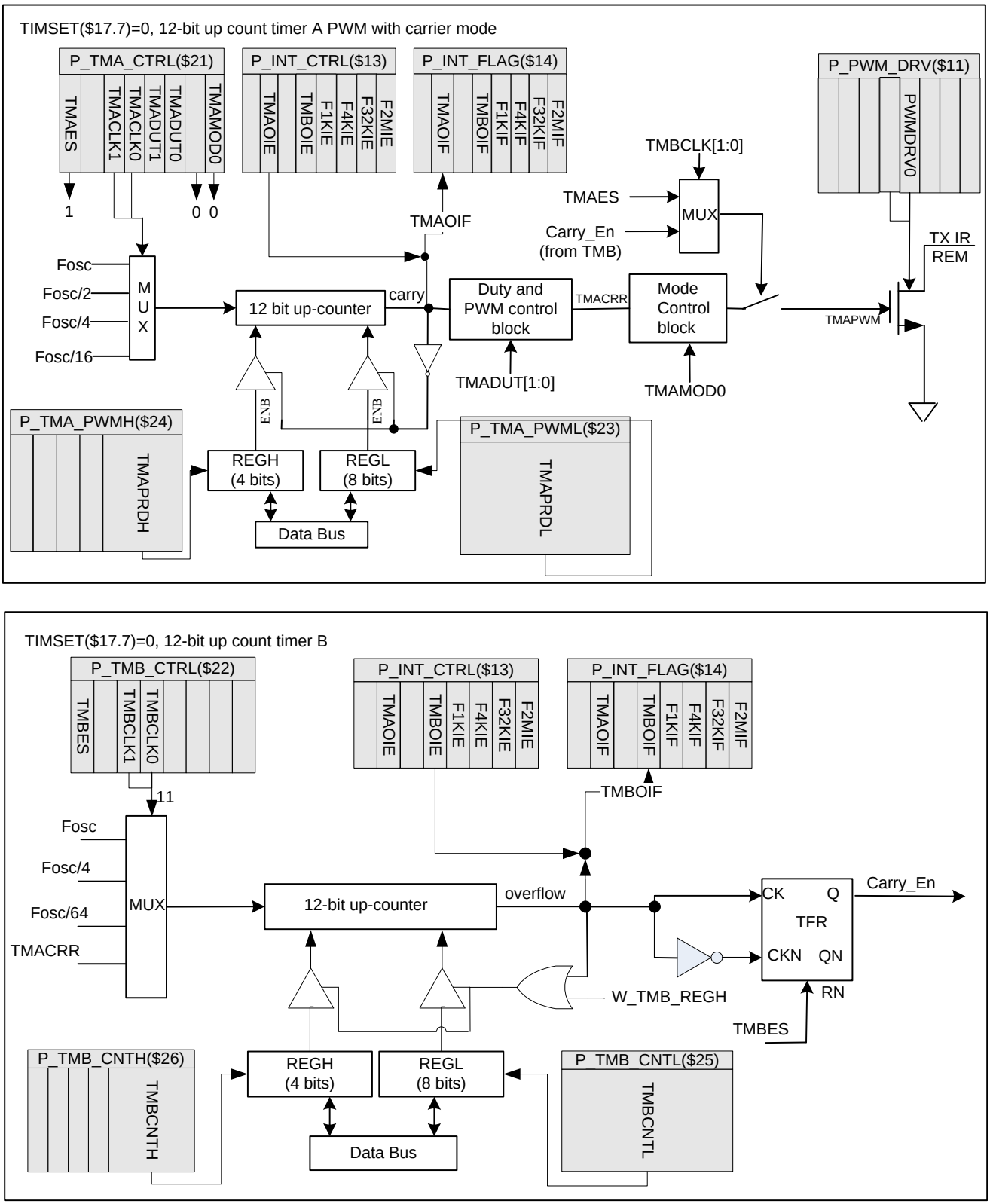


Figure 5-20 Envelope PWM Generated by Mode 0 Timer A & Mode 0 Timer B diagram

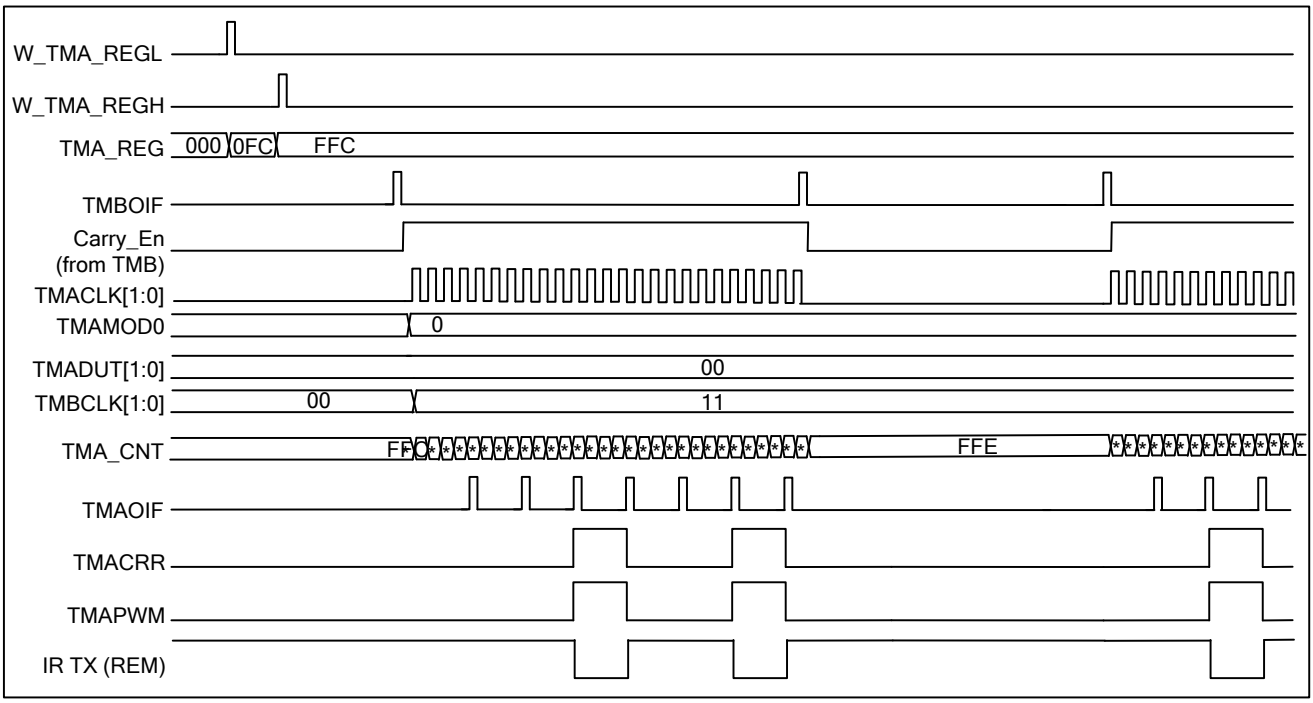


Figure 5-21 The Waveform of Mode 0 Timer A PWM with carrier signal mode (1/3 duty, on/off control by Mode 0 Timer B overflow events)

5.8.2.2. Mode 0 Timer A PWM without carrier signal mode

GPM6P1004B/1001B PWM without carrier signal mode is used to generate envelop PWM signal without carrier signal. In this mode, IR TX (REM) pin just output high or low, and is controlled by Timer A's enable or disable control bit or TimerB's overflow events in turn. The same as PWM with carrier signal mode, the 12-bit timer is an up counter with input clock selectable (Fosc/1, Fosc/4, Fosc/16). When the Timer A is started, the value of high-byte (low-nibble) Register and low-byte Register will firstly be loaded into the 12-bit counter and then the counter starts to count up from the loaded value. If an overflow occurs, the value of high-byte (low-nibble) register and low-byte register will be reloaded into the counter automatically and the counter starts to count up again. The internal carrier signal is generated but is not sent to IR TX pin.

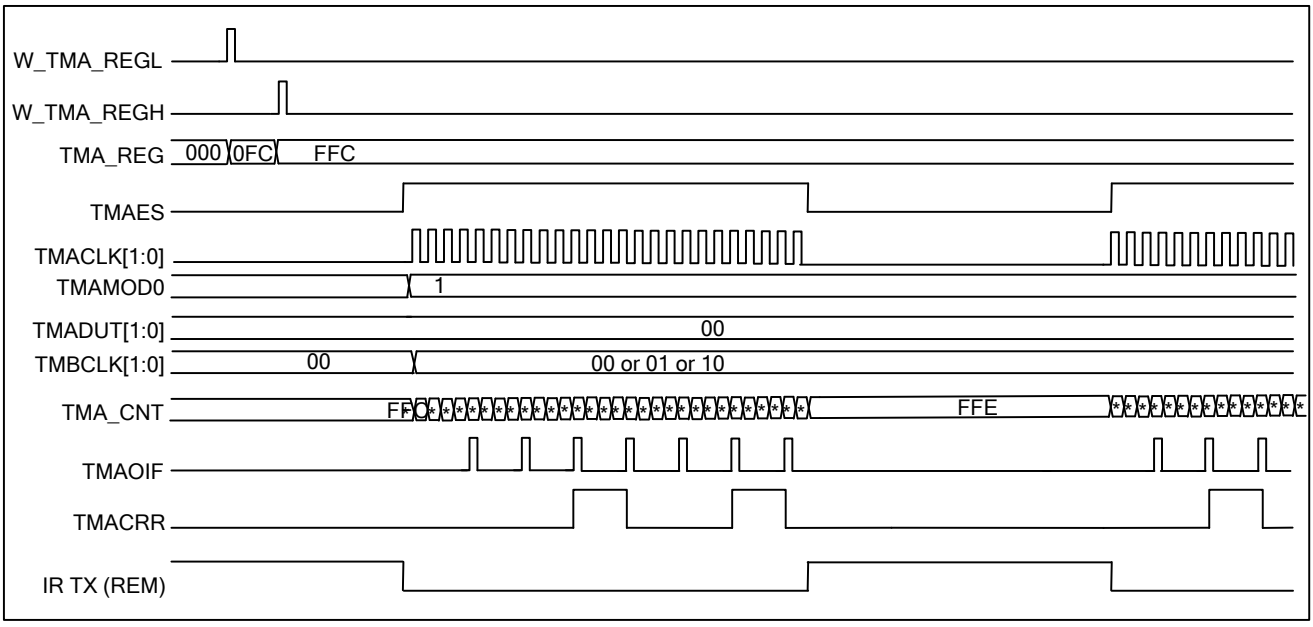


Figure 5-22 The Waveform of Mode 0 Timer A PWM without carrier signal mode (on/off control by TMAES)

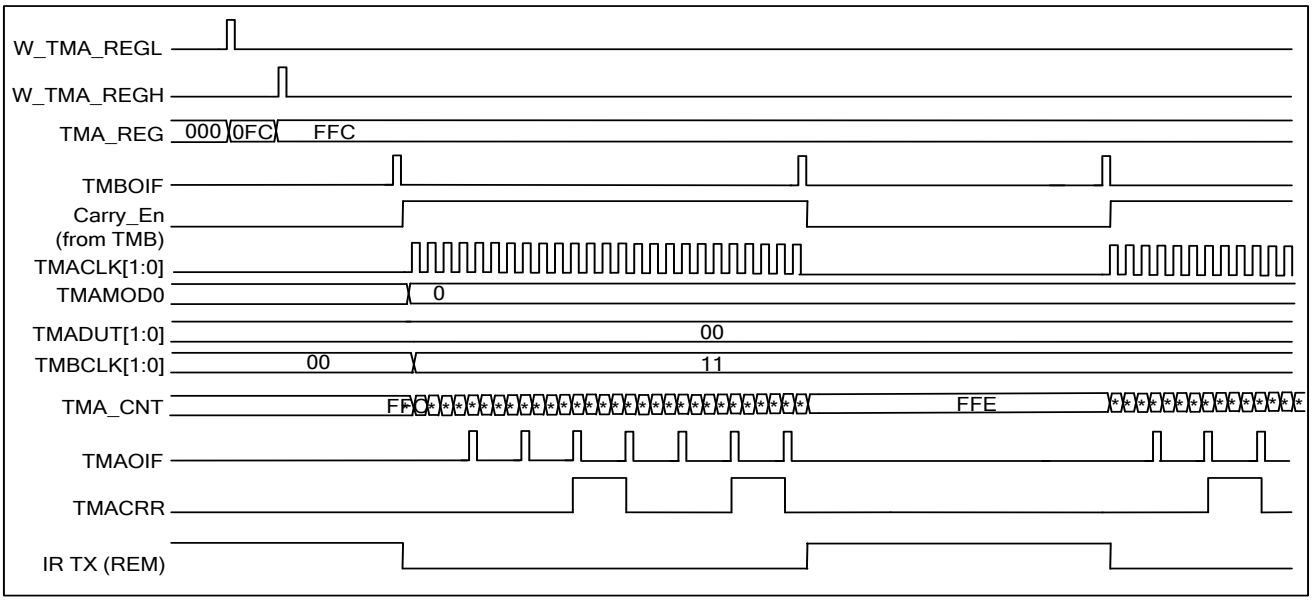


Figure 5-23 The Waveform of Mode 0 Timer A PWM without carrier signal mode (on/off control by Mode 0 Timer B overflow events)

Timer Special Configure Register (P_TIM_SEL, $0017)

BIT	7	6	5	4	3	2	1	0
Name	TIMSET	IRENB	-	-	-	-	-	-
Access	R/W	R/W	-	-	-	-	-	-
Default	0	0	-	-	-	-	-	-

Bit [7] **TIMSET :** Timer A/B up or down count select.

0 : Timer A 12-bit up count; Timer B 12-bit up count.(C_TIMAB_UP)

1 : Timer A 8-bit down count; Timer B 12-bit down count. (C_TIMAB_DN)

Bit [6] **IRENB:** PWM output function enable/disable. Active when TIMONS=1

0 : PWM output function enable; (C_PWM_EN)

1 : PWM output function disable. (C_PWM_DIS)

Bit [5:0] **Reserved**

Mode 0 Timer A Control Register (P_TMA_CTRL, $0021)

BIT	7	6	5	4	3	2	1	0
Name	TMAES		TMACLK1	TMACLK0	TMADUT1	TMADUT0		TMAMOD0
Access	R/W		R/W	R/W	R/W	R/W		R/W
Default	0		0	0	0	0		0

Bit [7] **TMAES:** Timer A enable/disable control.

0 : disable; (C_TMAES_DIS)

1 : enable. (C_TMAES_EN)

Bit [6] **Reserved**

Bit [5:4] **TMACLK[1:0]:** Timer A clock source select bits

00 : Fosc (C_TMACLK_1)

01 : Fosc/2 (C_TMACLK_2)

10 : Fosc/4 (C_TMACLK_4)

11 : Fosc/16 (C_TMACLK_16)

Bit [3:2] **TMADUT[1:0]:** Timer A PWM duty selection

00 : 1/3 (C_TMADUT_3)

01 : 1/4 (C_TMADUT_4)

10 : 1/5 (C_TMADUT_5)

11 : 1/2 (C_TMADUT_2)

Bit[1] **Reserved**

Bit [0] **TMAMOD0:** Timer A mode setting

0 : PWM (C_TMAMOD_WTC)

1 : PWM1 (enter the mode, PWM out always high) (C_TMAMOD_WOC)

Mode 0 Timer A Count Low Byte Register (P_TMA_CNTL, $23)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMACNTL7	TMACNTL6	TMACNTL5	TMACNTL4	TMACNTL3	TMACNTL2	TMACNTL1	TMACNTL0
Access	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMACNTL**[7：0]: Timer A low byte 8-bit pre-value for the counter.

Read : Timer A Count Low Byte Value(R)

Write : Timer A Pre-Load Count Low Byte Value (W)

Mode 0 Timer A PWM Low Byte Period Register (P_TMA_PWML, $23)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMAPRDL7	TMAPRDL6	TMAPRDL5	TMAPRDL4	TMAPRDL3	TMAPRDL2	TMAPRDL1	TMAPRDL0
Access	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMAPRDL**[7：0]: Timer A low byte 8-bit period value for the PWM.

Read : Timer A Count Low Byte Value(R)

Write : PWM signal carrier signal Pre-load Period Low Byte Value (W)

Mode 0 Timer A Count High Byte Register (P_TMA_CNTH, $24)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	TMACNTH3	TMACNTH2	TMACNTH1	TMACNTH0
Access	-	-	-	-	R	R	R	R
Default	-	-	-	-	0	0	0	0

Bit [7:4] **Reserved**

Bit [3:0] **TMACNTH**[3：0]: Timer A high byte 4-bit pre-value for the counter.

Read : Timer A Count High Byte Value (R)

Write : Timer A Pre-Load Count High Byte Value (W)

Mode 0 Timer A PWM High Byte Period Register (P_TMA_PWMH, $24)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	TMAPRDH3	TMAPRDH2	TMAPRDH1	TMAPRDH0
Access	-	-	-	-	W	W	RW	W
Default	-	-	-	-	0	0	0	0

Bit [7:4] **Reserved**

Bit [3:0] **TMAPRDH**[3：0]: Timer A high byte 4-bit period value for the PWM.

Read : Timer A Count High Byte Value(R)

Write : PWM signal carrier signal Pre-load Period High Byte Value (W)

[Example] 5-10 Set Timer A as PWM with carrier signal mode.

```
LDA     #C_TIMAB_UP
STA     P_TIM_SEL                                   ;set timer as up count
LDA     #$FC                                        ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_PWML                                   ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMA_PWMH                                   ;set high 4-bit pre-value
LDA     #C_TMAES_EN + #C_TMACLK _4 +#C_TMADUT_3 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                  ;Set clock source Fosc/4, 1/3duty, PWM with carrier signal mode
```

5.8.2.3. PWM Carrier Signal Algorithm

The frequency of PWM carrier signal (F_{PWM}) generated by Timer A depends on three factors.

- The initial value (V_{REG}=12-bit Preload PREIOD) is filled into high-byte (low-nibble) register (P_TMA_PWMH [3:0]) and low-byte register (P_TMA_PWML [7:0])
- The duty of the carrier signal (DUT= PWM DUTY).
- The frequency of timer A clock source (F_{timer})

V_{REG} = P_TMA_PWMH[4:0]+P_TMA_PWML[7:0]
DUT = one of (1/3, 1/4, 1/5, 1/6), defined by P_TMA_CTRL[3:2]
If
F_{timer} = F_{osc}/1 or F_{osc}/2, defined by P_TMA_CTRL[5:4]
Then
V_{REG} = 4097- F_{timer} / F_{PWM} * DUT

For example, if user needs to generate 38 KHz 1/3 duty PWM carrier frequency and TIMER clock source is 4MHz/1 (system clock is 4MHz).

Condition: F_{PWM} = 38 KHz, F_{timer} =4MHz, DUT=1/3
V_{REG} = 4097 – (4M/38K)*1/3 = 4062 =FDEH

Then the result FDEH can be written into the PWM high/low register, and the 38 KHz PWM signal is generated.

V_{REG} = P_TMA_PWMH[4:0]+P_TMA_PWML[7:0]
DUT = one of (1/3, 1/4, 1/5, 1/2), defined by P_TMA_CTRL[3:2]
If
F_{timer} = F_{osc}/4, F_{osc}/16 , defined by P_TMA_CTRL[5:4]
Then
V_{REG} = 4096- F_{timer} / F_{PWM} * DUT

For example, if user needs to generate 38 KHz 1/3 duty PWM carrier frequency, and system frequency is 4MHz. and F_{OSC}/4 is selected as timer clock.

Condition: F_{PWM} = 38 KHz, F_{timer} = 4MHz/4, DUT=1/3
V_{REG} = 4096 – (1M/38K)*1/3 = 4087 =FF7H

Then the result FF7H can be written into the PWM high/low register, and the 38 KHz PWM signal is generated.

[Example] 5-11 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38 KHz with 1/3 duty (clock source=Fosc/1).

```
LDA     #C_TIMAB_UP
STA     P_TIM_SEL                                   ;set timer as up count and enable PWM output function
LDA     #$DE                                        ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_ PWML                                  ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMA_ PWMH                                  ;set high 4-bit pre-value
LDA     #C_TMAES_EN + #C_TMACLK _1 +#C_TMADUT_3 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                  ;Set clock source Fosc/1, 1/3duty, PWM with carrier signal mode
```

[Example] 5-12 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38 KHz with 1/3 duty (clock source=Fosc/4).

```
LDA     #C_TIMAB_UP
STA     P_TIM_SEL                                   ;set timer as up count
LDA     #$F7                                        ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_ PWML                                 ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMA_ PWMH                                 ;set high 4-bit pre-value
LDA     #C_TMAES_EN + #C_TMACLK _4 +#C_TMADUT_3 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                  ;Set clock source Fosc/4, 1/3 duty, PWM with carrier signal mode
```

5.8.3. Mode 0 Timer B (12-bit up count timer)

When Timer A is selected as 12-bit up count timer via configuring the corresponding bit of register (P_TIM_SEL[7]), the Timer B is selected as 12-bit up count timer too. The Timer B is special for envelope signal generation in IR controller application. The 12-bit timer is an up counter with input clock selectable (Fosc/1, Fosc/4, Fosc/64, TMACAR) via configuring the control register P_TMB_CTRL [5:4] (TMBCLK [1:0]). And the value of low-byte register (P_TMB_CNTL) and high-byte (low-nibble) register (P_TMB_CNTH) will be reloaded into the 12-bit up counter and an interrupt (TMBOIF) will be generated whenever an overflow occurs. The interrupt frequency can be freely selected by selecting different clock source and configuring the low-byte register and high-byte (low-nibble) register with different values.

The Timer B module has the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$FFF to #$000

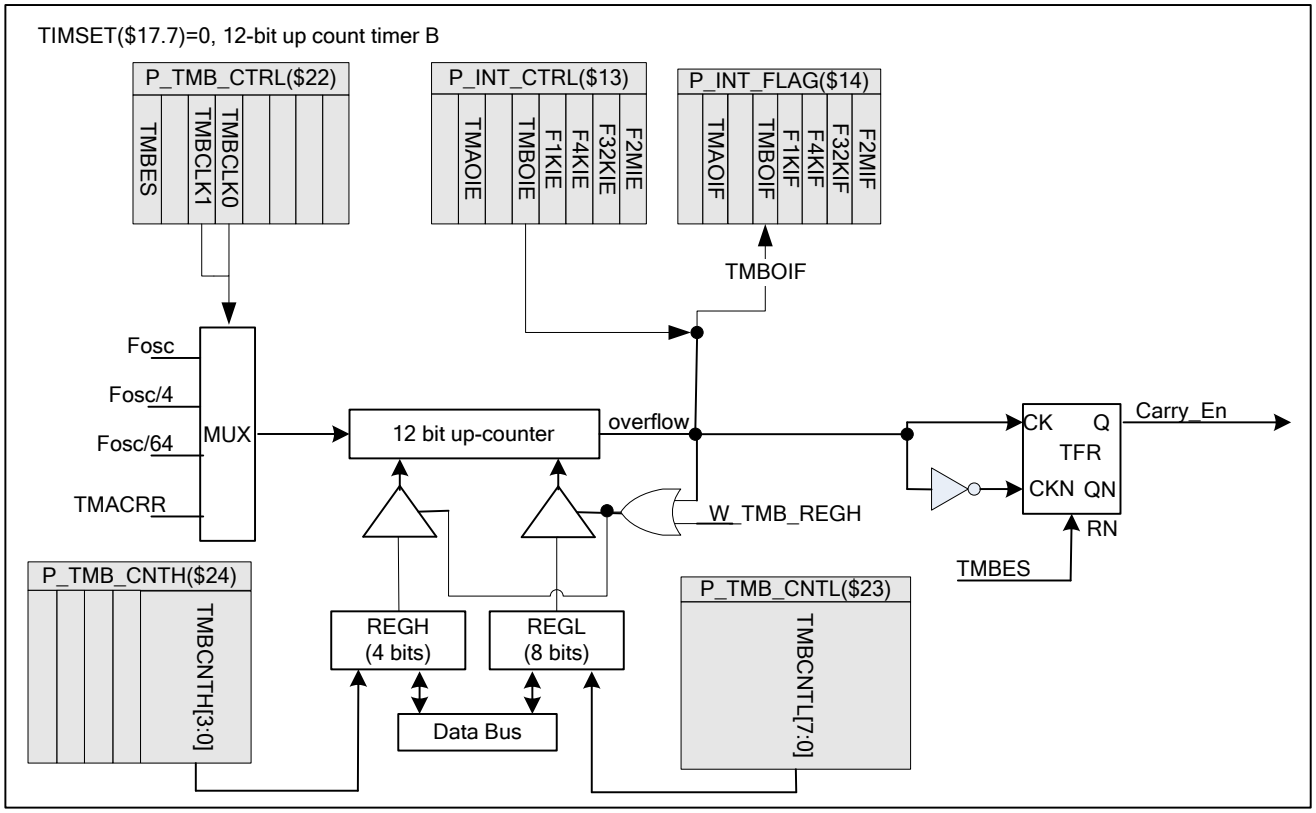


Figure 5-24 Mode 0 Timer B block diagram

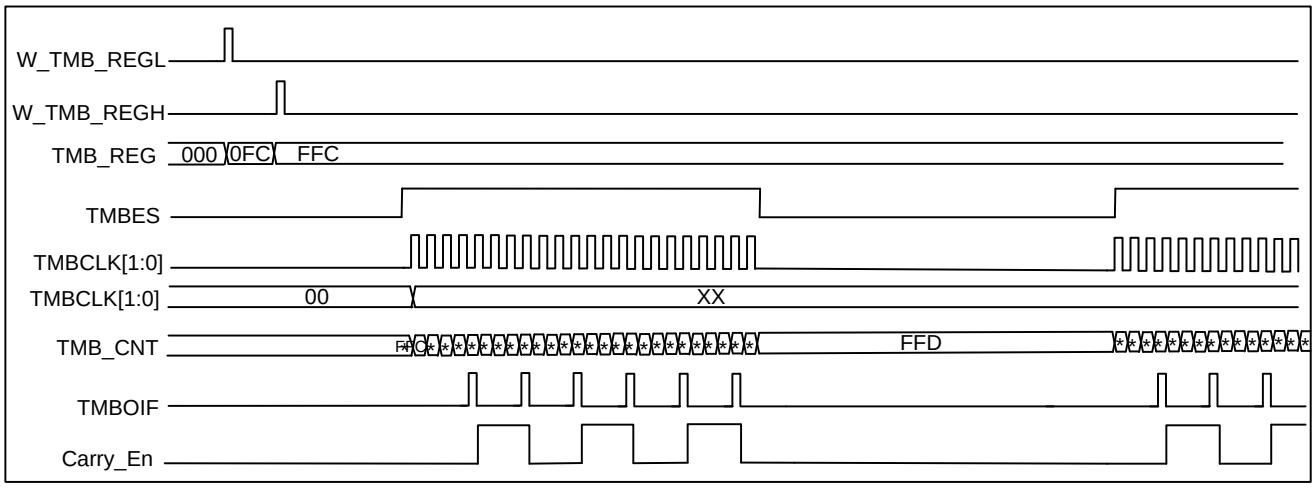


Figure 5-25 The Waveform of mode 0 Timer B

Mode 0 Timer B Control Register (P_TMB_CTRL, $0022)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBES	-	TMBCLK1	TMBCLK0	-	-	-	-
ACCESS	R/W	-	R/W	R/W	-	-	-	-
DEFAULT	0	-	0	0	-	-	-	-

Bit [7] **TMBES**: Timer B enable/disable control selected bit.
0 : disable (C_TMBES_DIS)
1 : enable (C_TMBES_EN)

Bit [6] **Reserved**

Bit [5:4] **TMBCLK[1：0]**: Timer B clock source selected bits
00: Fosc (C_TMBCLK_1)
01: Fosc/4 (C_TMBCLK_4)
10: Fosc/64 (C_TMBCLK_64)
11: TMACRR (C_TMBCLK_TMACRR)

Bit [3:0] **Reserved**

Mode 0 Timer B Count low 8-bit data Register (P_TMB_CNTL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBCNTL7	TMBCNTL6	TMBCNTL5	TMBCNTL4	TMBCNTL3	TMBCNTL2	TMBCNTL1	TMBCNTL0
ACCESS	R	R	R	R	R	R	R	R
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **TMBCNTL**[7：0]: Timer B count low byte 8-bit value for the counter.

Mode 0 Timer B low 8-bit data Register (P_TMB_REGL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBREGL7	TMBREGL6	TMBREGL5	TMBREGL4	TMBREGL3	TMBREGL2	TMBREGL1	TMBREGL0
ACCESS	W	W	W	W	W	W	W	W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **TMBREGL**[7：0]: Timer B low byte 8-bit pre-value for the counter.

Mode 0 Timer B Count high 4-bit Register (P_TMB_REGL, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBCNTH3	TMBCNTH2	TMBCNTH1	TMBCNTH0
ACCESS	-	-	-	-	R	R	R	R
DEFAULT	-	-	-	-	0	0	0	0

Bit [7:4] **Reserved**

Bit [3:0] **TMBCNTH**[3：0]: Timer B Count High byte 4-bit value for the counter

Mode 0 Timer B high 4-bit data Register (P_TMB_REGH, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBREGH3	TMBREGH2	TMBREGH1	TMBREGH0
ACCESS	-	-	-	-	W	W	W	W
DEFAULT	-	-	-	-	0	0	0	0

Bit [7:4] **Reserved**

Bit [3:0] **TMBREGH**[3：0]: Timer B High byte 4-bit pre-value for the counter.

[Example] 5-13 Set Timer B selects timer A carrier signal as counter clock.

```
LDA     #C_TIMAB_UP
STA     P_TIM_SEL                       ; set Timer A/B as 12-bit up count timers
LDA     #$FC                            ; Before starting timer, set Timer B counter initial value first
STA     P_TMB_CNTL                      ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMB_CNTH                      ; set high 4-bit pre-value
LDA     #C_TMBES_EN + #C_TMBCLK_TMACRR
STA     P_TMB_CTRL                      ;Set clock source for TMA_Carrier
```

5.8.4. Mode 1 Timer A (8-bit down count timer)

When Timer A is selected as 8-bit down count timer via configuring the corresponding bits of the control register (P_TIM_SEL[7]), the Timer A is special for generating carrier signal in IR control application. The timer A's input clock is selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16), which can be configured by control register P_TMA_CTRL[5:4]. Timer A provides with two PWM modes, and the PWM signal is send to IR TX (REM) pin. The driver current of these two kinds of PWM are programmable by configuring TX PWM driving current control source register (P_PWM_DRV [3]).

GPM6P1004B/1001B 8-bit down count Timer A module has all of the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$00 to #$FF
- Support PWM with carrier signal mode
- Support PWM without carrier signal mode

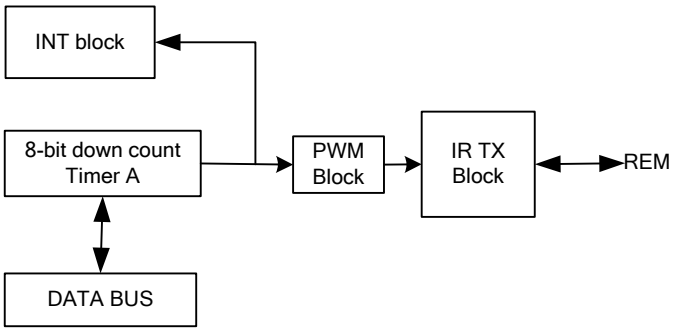


Figure 5-26 GPM6P1004B/1001B Mode 1 timer A block diagram

5.8.4.1. Mode 1 timer A PWM with carrier signal mode

The Timer A can be configured as PWM mode for generating carrier signal. In PWM with carrier signal mode, the 8-bit timer is an down counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When Timer A is started, the value of 8-bit cycle width (frequency) set register will firstly be loaded into the 8-bit counter and the value of 8-bit high pulse width (duty) set register will be loaded into the compare unit. And then the counter starts count down from the loaded value. PWM initial output low, and if the counter value is same as the value in compare unit, the PWM will switch to high. If an overflow occurs, the PWM switch to low once again, and the value of frequency register (P_TMA_PWMF, $23) and duty register (P_TMA_PWMD,

$24) will be reloaded into the counter and the compare unit automatically and the counter starts count down again. So the carrier signal with frequency and duty programmable can be generated by this PWM mode via configuring these two registers. The carrier signal's enabled or disabled bit can be controlled by two methods depended on which clock source is selected by timer B. If timer B is selected one of the first three clock source (Fosc, Fosc/4 or Fosc/64) by P_TMB_CTRL [5:4] (TMBCLK [1:0]), the timer A's carrier signal on/ off is controlled by timer A's enable/ disable control bit (TMAES) directly. In addition, PWM output function also can be disabled by writing 1 to register IRENB($17.6).

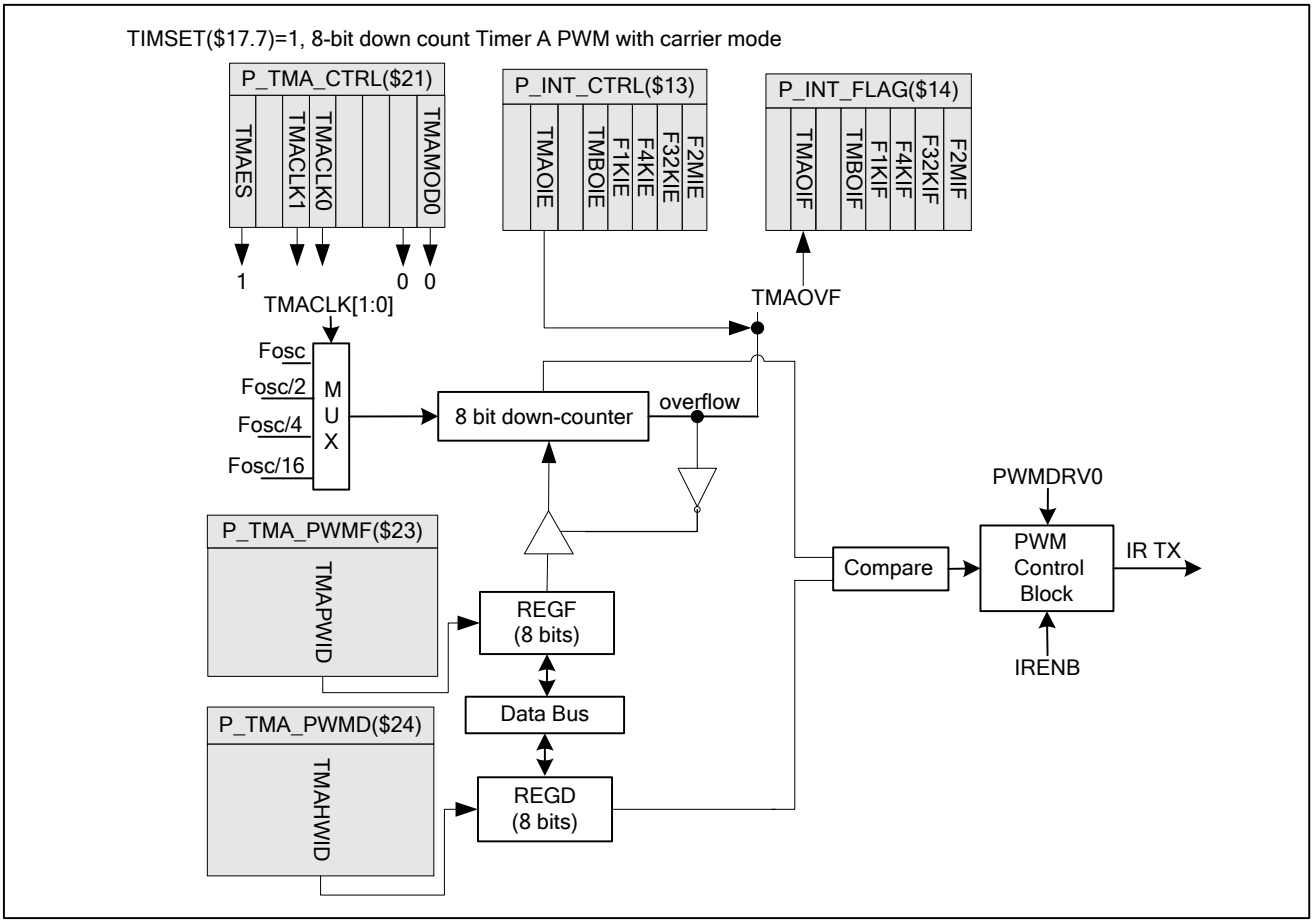


Figure 5-27 Mode 1 Timer A PWM mode diagram

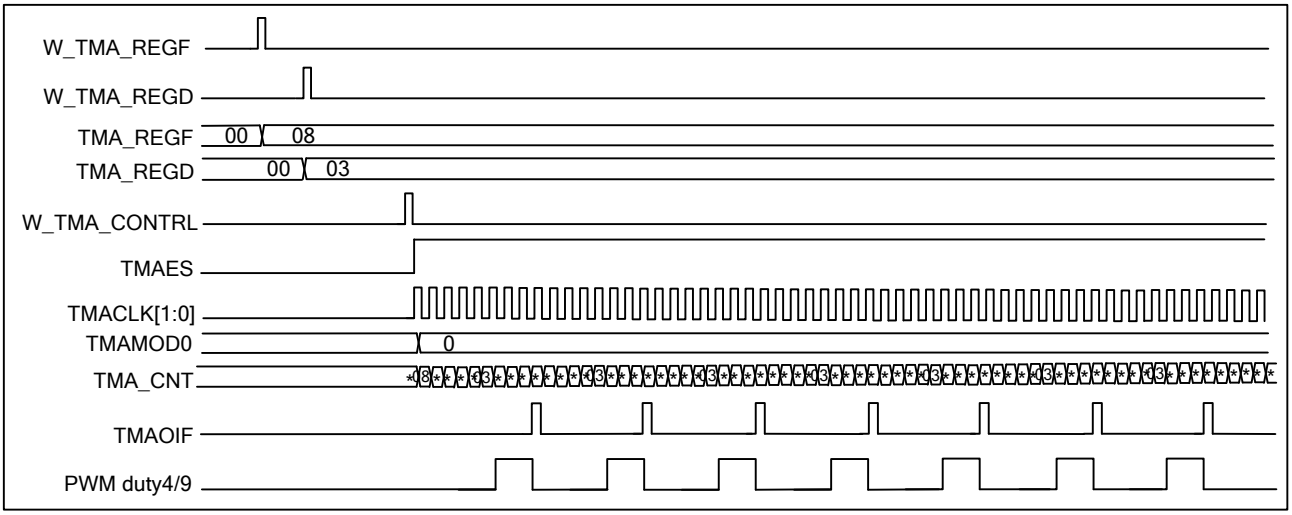


Figure 5-28 Mode 1 Timer A Normal PWM generation without envelop

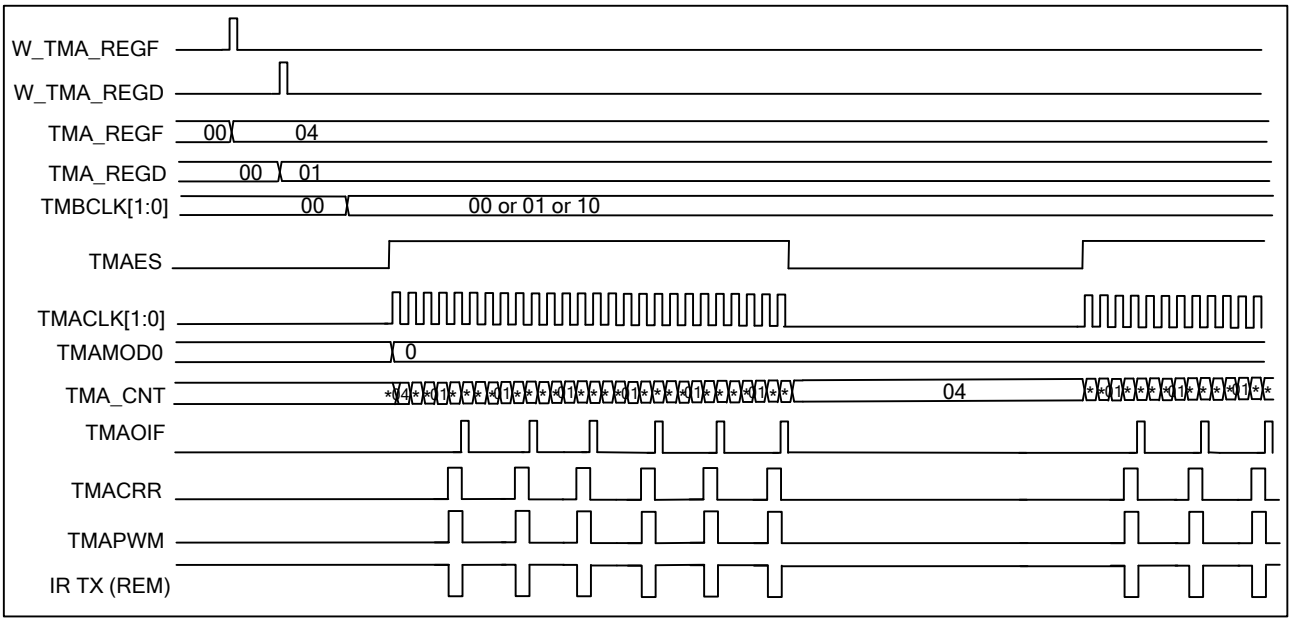


Figure 5-29 The Waveform of mode 1 timer A PWM with carrier signal mode (1/5 duty, on/off control by TMAES)

Another method to generate envelope PWM signal is that Timer A and Timer B must be used together. Timer A must generate carry clock at first, which is same as normal PWM generation. Then enable Timer B and select Timer A's carrier signal as its input clock. And Timer B register must be written in the right data, which represents the carry number. When TMBOVF happens, another value must be written into Timer B register, which represents the no carry clock number. Envelop with carrier is on or off only when Timer B overflow events occur one by one. Then, the envelop PWM signal will be generated at REM port at last.

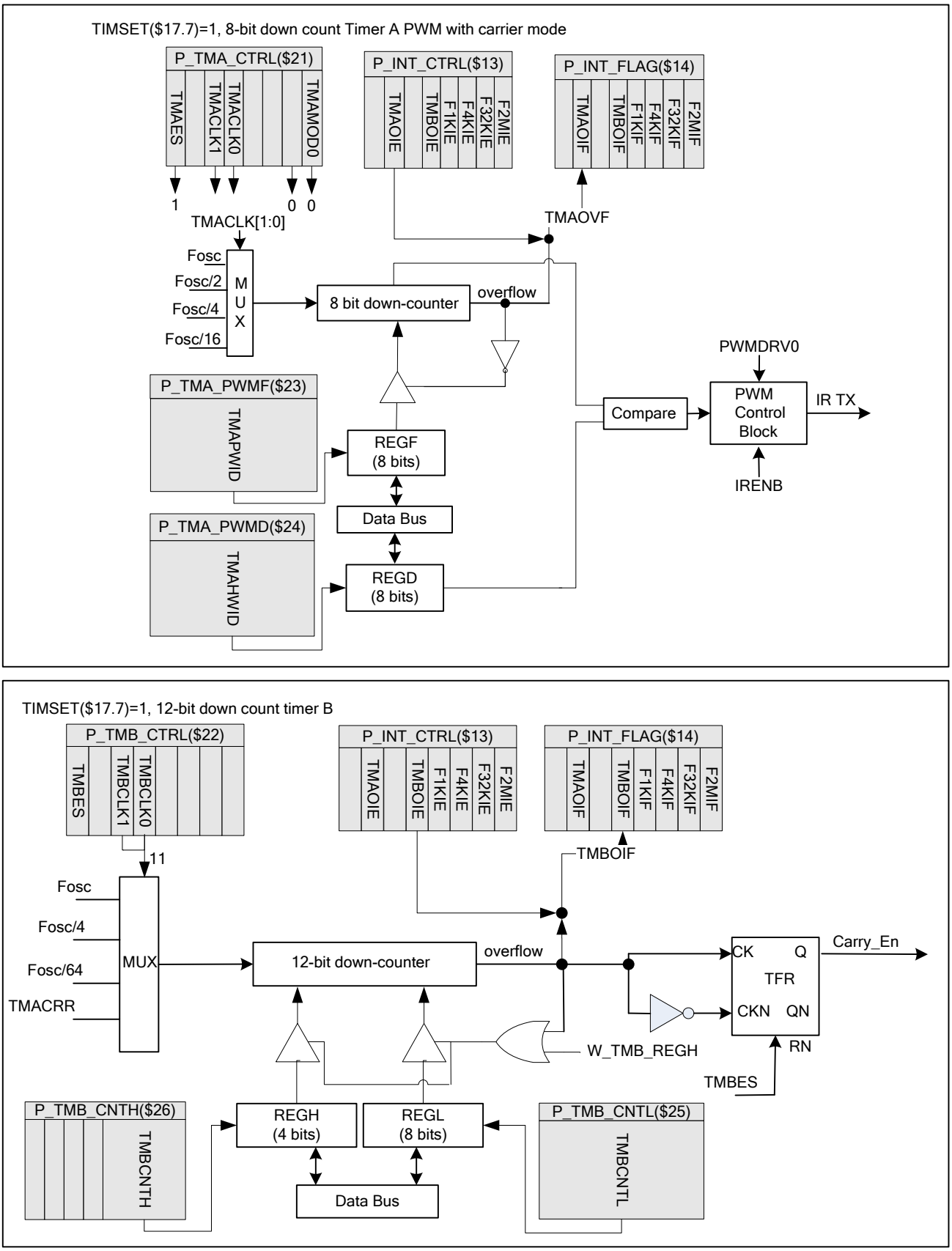


Figure 5-30 Envelope PWM Generated by mode 1 Timer A & mode 1 Timer B diagram

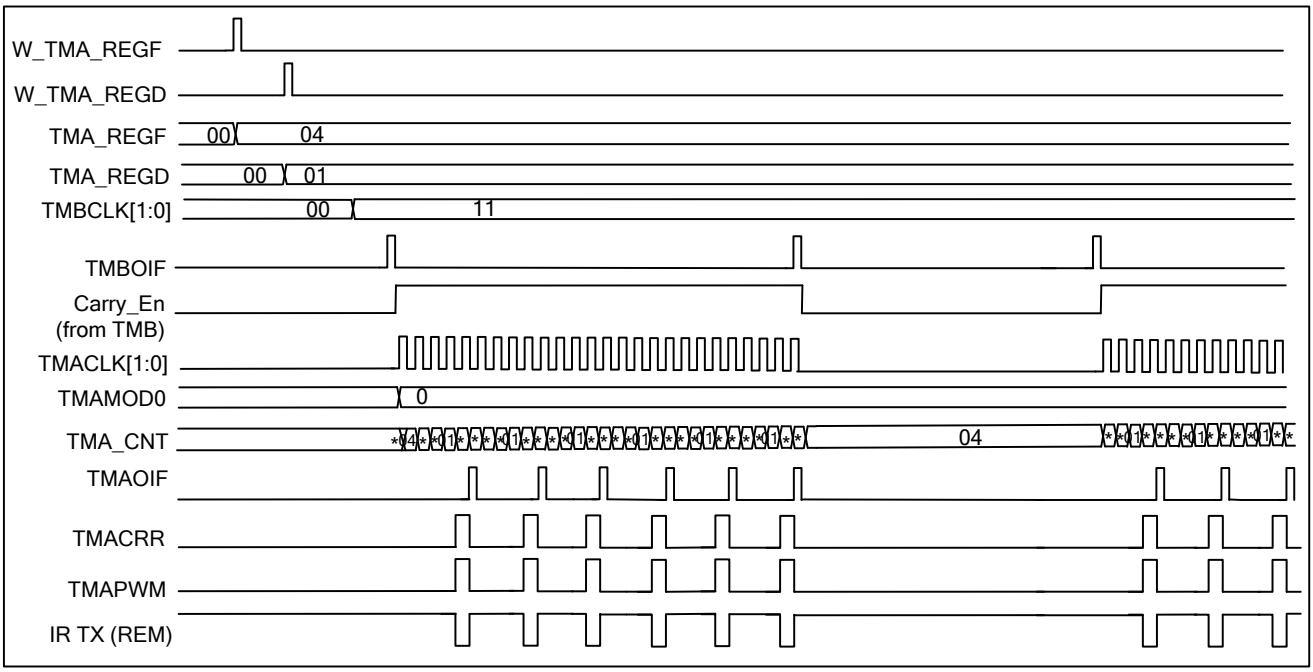


Figure 5-31 The Waveform of mode 1 timer A PWM with carrier signal mode (1/5 duty, on/off control by mode 1Timer B overflow events)

5.8.4.2. PWM without carrier signal mode

PWM without carrier signal mode is used to generate envelop PWM signal without carrier signal. In this mode, IR TX (REM) pin just output high or low, and is controlled by TimerA's enable or disable control bit or Timer B's overflow events in turn. The same as PWM with carrier signal mode, the 8-bit timer is an down counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When the Timer A is started, the value of 8-bit pre-value Register (P_TMA_PWMF, $23) will firstly be loaded into the 8-bit counter and then the counter starts to count down from the loaded value. If an overflow occurs, the value of pre-value register will be reloaded into the counter automatically and the counter starts to count down again. The internal carrier signal is generated but does not be sent to IR TX pin.

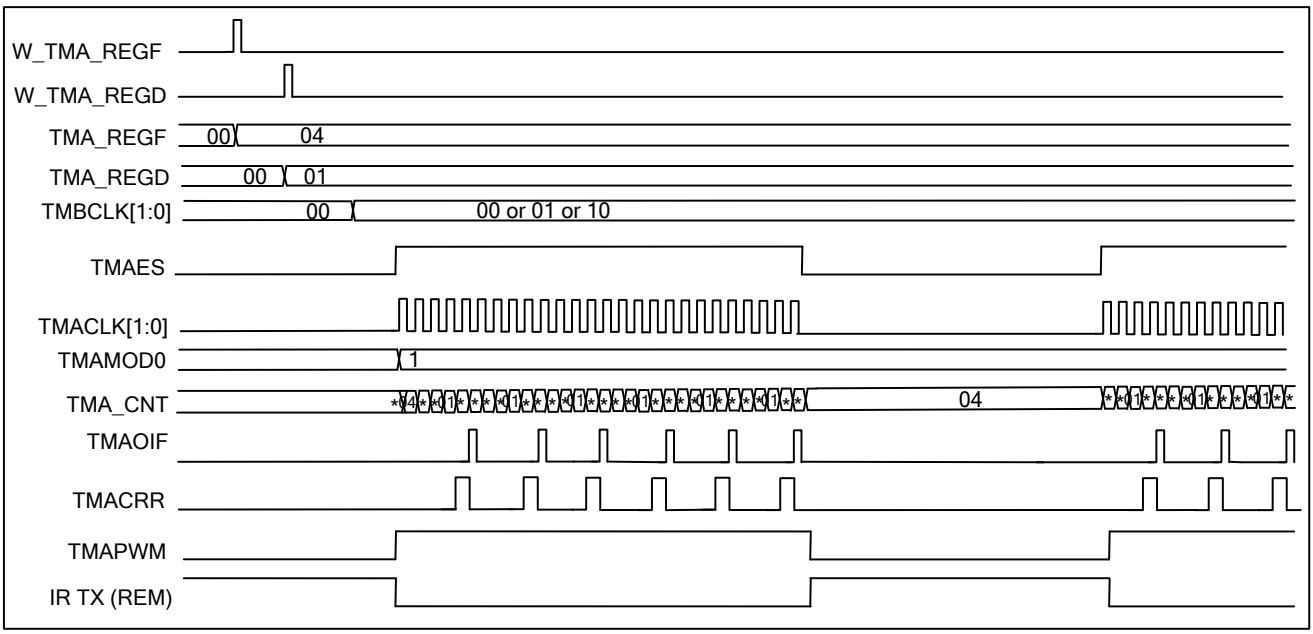


Figure 5-32 The Waveform of mode 1 timer A PWM without carrier signal mode (on/off control by TMAES)

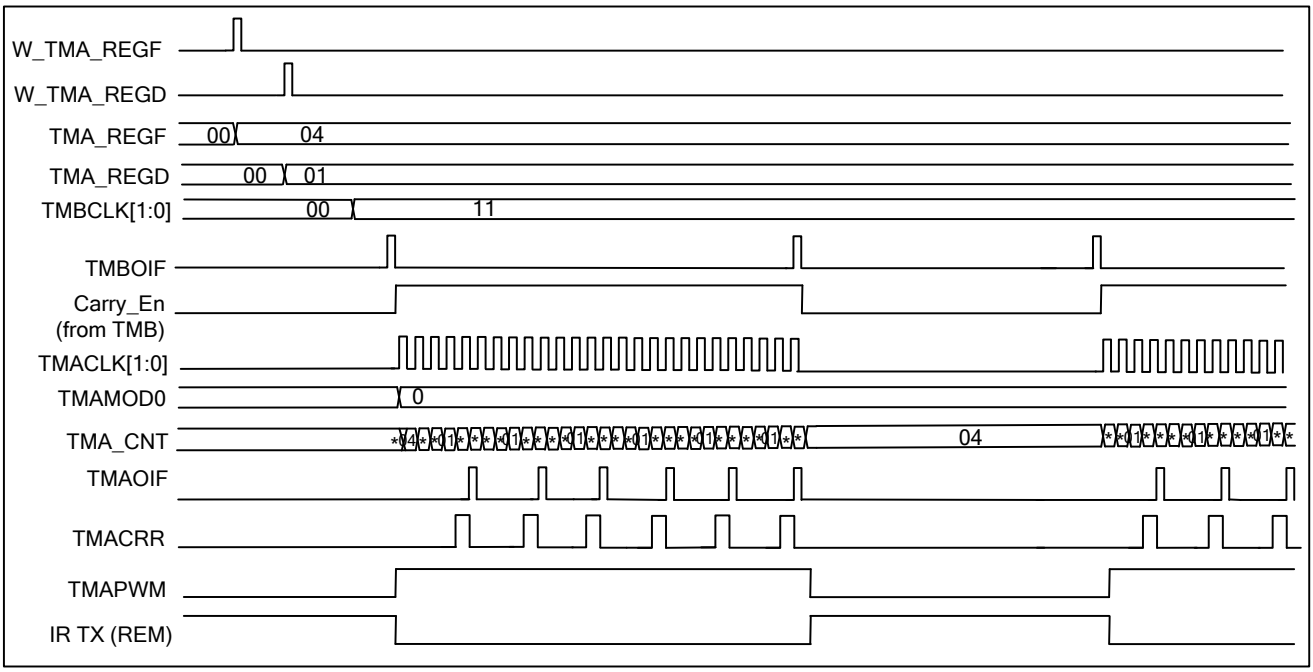


Figure 5-33 The Waveform of mode 1 timer A PWM without carrier signal mode (on/off control by mode 1 Timer B overflow events)

Mode 1 Timer A Control Register (P_TMA_CTRL, $0021)

BIT	7	6	5	4	3	2	1	0
Name	TMAES		TMACLK1	TMACLK0				TMAMOD0
Access	R/W		R/W	R/W				R/W
Default	0		0	0				0

Bit 7 **TMAES:** Timer A enable/disable control.
0 : disable;(C_TMAES_DIS)
1 : enable.(C_TMAES_EN)

Bit 6 **Reserved**

Bit [5:4] **TMACLK[1:0]:** Timer A clock source select bits
00 : Fosc (C_TMACLK_1)
01 : Fosc/2 (C_TMACLK_2)
10 : Fosc/4 (C_TMACLK_4)
11 : Fosc/16 (C_TMACLK_16)

Bit [3:2] **Reserved**

Bit [1] **Reserved**

Bit [0] 0 : PWM (C_TMAMOD_WTC)
1 : PWM1 (enter the mode, PWM out always high) (C_TMAMOD_WOC)

Mode 1 Timer A Count Register (P_TMA_CNTF, $23)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMACNT7	TMACNT6	TMACNT5	TMACNT4	TMACNT3	TMACNT2	TMACNT1	TMACNT0
Access	R	R	R/	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMACNT**[7：0]: Timer A 8-bit pre-value for the counter.
Read : Timer A Count Value(R)
Write : Timer A Pre-Load Count Value (W)

Mode 1 Timer A PWM Carrier Signal Period (Frequency) Register for PWM Mode(P_TMA_PWMF, $23)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMAPWID7	TMAPWID6	TMAPWID5	TMAPWID4	TMAPWID3	TMAPWID2	TMAPWID1	TMAPWID0
Access	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMAPWID**[7：0]: Timer A carrier signal period(frequency) value for the PWM.

Read : Timer A Count Value(R)

Write : Timer A Pre-Load carrier signal Period(frequency) Value (W)

Mode 1 Timer A PWM Carrier Signal High Pulse (Duty) Width Register for PWM Mode (P_TMA_PWMD, $24) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMAHWID7	TMAHWID6	TMAHWID5	TMAHWID4	TMAHWID3	TMAHWID2	TMAHWID1	TMAHWID0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMAHWID**[7：0]: Timer A 8-bit high pulse (duty) value for the carrier signal of PWM.

Read : Timer A high pulse (duty) Value (R)

Write : Timer A Pre-Load carrier signal high pulse (duty) Value (W)

[Example] 5-14 Set Timer A as PWM with carrier signal mode.

```
LDA     #C_TIMAB_DN + #C_PWM_EN
STA     P_TIM_SEL                                   ;set timer as down count and enable PWM output function
LDA     #$0F                                        ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_PWMF                                   ; set Period pre-value
LDA     #$08
STA     P_TMA_PWMD                                   ;set high pulse pre-value (DUTY=($08+1)/($0F+1)=9/16)
LDA     #C_TMAES_EN + #C_TMACLK _4 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                  ;Set clock source Fosc/4, PWM with carrier signal mode
```

5.8.4.3. PWM Carrier Signal Algorithm

The frequency of PWM carrier signal (F_{PWM}) generated by Timer A depends on three factors.

- The initial value (V_{REGF}=8-bit Preload PREIOD) is filled into register (P_TMA_PWMF [7:0]).
- The initial value (V_{REGD}=8-bit Preload HIGH PULSE Value) is filled into register (P_TMA_PWMD [7:0]).
- The frequency of timer A clock source (F_{timer})

V_{REGF} = P_TMA_PWMF[7:0]

V_{REGD} = P_TMA_PWMD[7:0]

If

F_{timer} = F_{osc}/1 or F_{osc}/2 or F_{osc}/4 or F_{osc}/16, defined by P_TMA_CTRL[5:4]

Then

V_{REGF} = F_{timer} / F_{PWM} -1

V_{REGD} = (F_{timer} / F_{PWM}) * DUT

For example, if user needs to generate 38 KHz 2/5 duty PWM carrier frequency and TIMER clock source is 4MHz/1 (system clock is 4MHz).

Condition: F_{PWM} = 38 KHz, F_{timer} =4MHz, DUT=2/5

V_{REGF} = F_{timer} / F_{PWM} -1= 104 =68H

V_{REGD} = (F_{timer} / F_{PWM}) * DUT = 42 =2AH

Then the result 68H and 2AH can be written into the PWM Period register and High pulse register separately, and the 38 KHz PWM signal is generated.

[Example] 5-15 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38Hz with 2/5 duty (clock source=Fosc/1).

```
LDA     #C_TIMAB_DN + #C_PWM_EN
STA     P_TIM_SEL                                   ;set timer as down count and enable PWM output function
LDA     #$68                                        ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_ PWMF                                 ; set low Period pre-value
LDA     #$2A
STA     P_TMA_ PWMD                                  ;set high pulse pre-value(2/5 duty)
LDA     #C_TMAES_EN + #C_TMACLK _1 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                  ;Set clock source Fosc/1, PWM with carrier signal mode
```

5.8.5. Mode 1 Timer B (12-bit down count timer)

The Timer B is special for envelope signal generation in IR controller application. The 12-bit timer is an down counter with input clock selectable (Fosc/1, Fosc/4, Fosc/64, TMACAR) via configuring the control register P_TMB_CTRL [5:4] (TMBCLK [1:0]). And the value of low-byte register (P_TMB_CNTL) and high-byte (low-nibble) register (P_TMB_CNTH) will be reloaded into the 12-bit up counter and an interrupt (TMBOIF) will be generated whenever an overflow occurs. The interrupt frequency can be freely selected by selecting different clock source and configuring the low-byte register and high-byte (low-nibble) register with different values.

The Timer B module has the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$000 to #$FFF

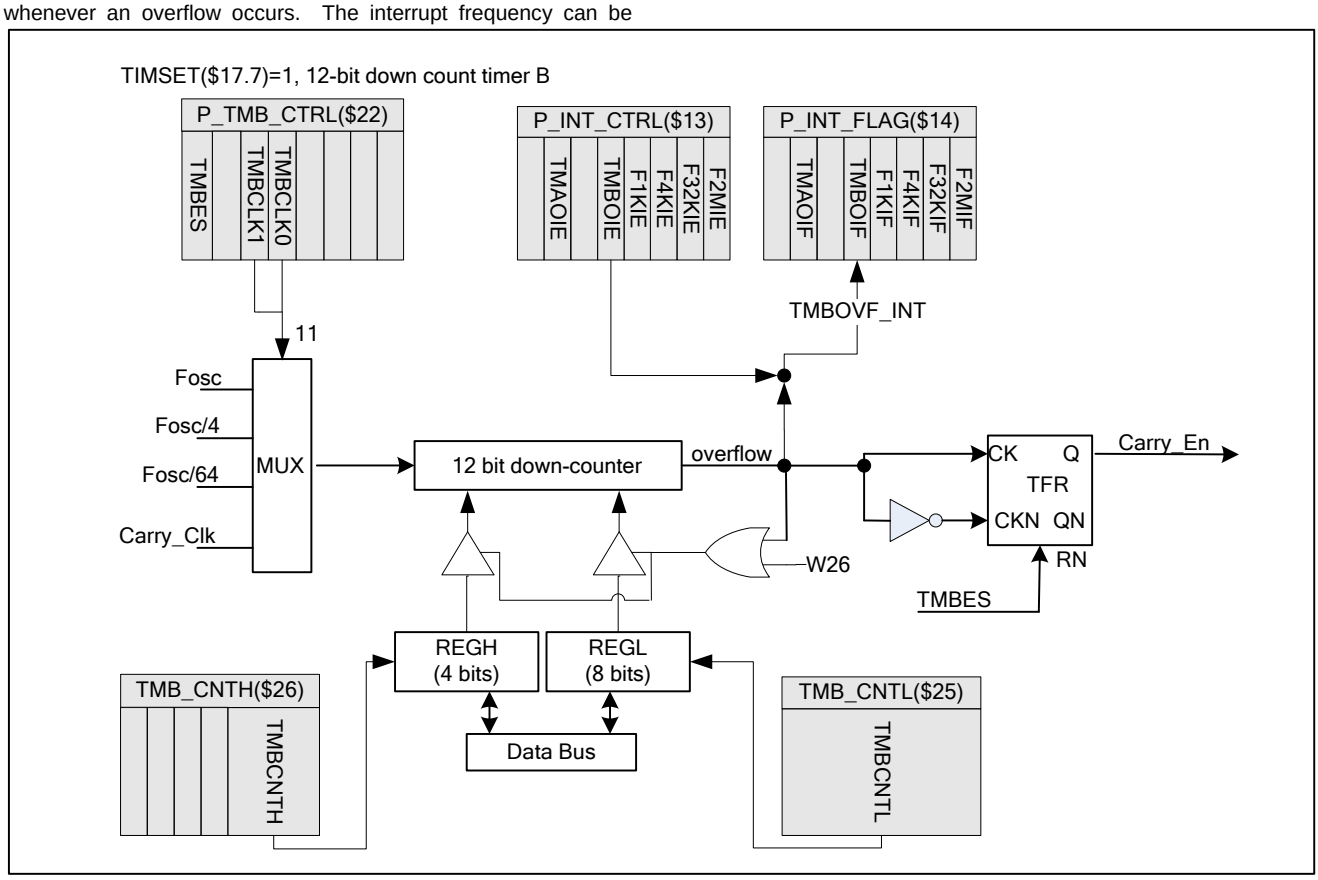


Figure 5-34 Mode 1 Timer B block diagram

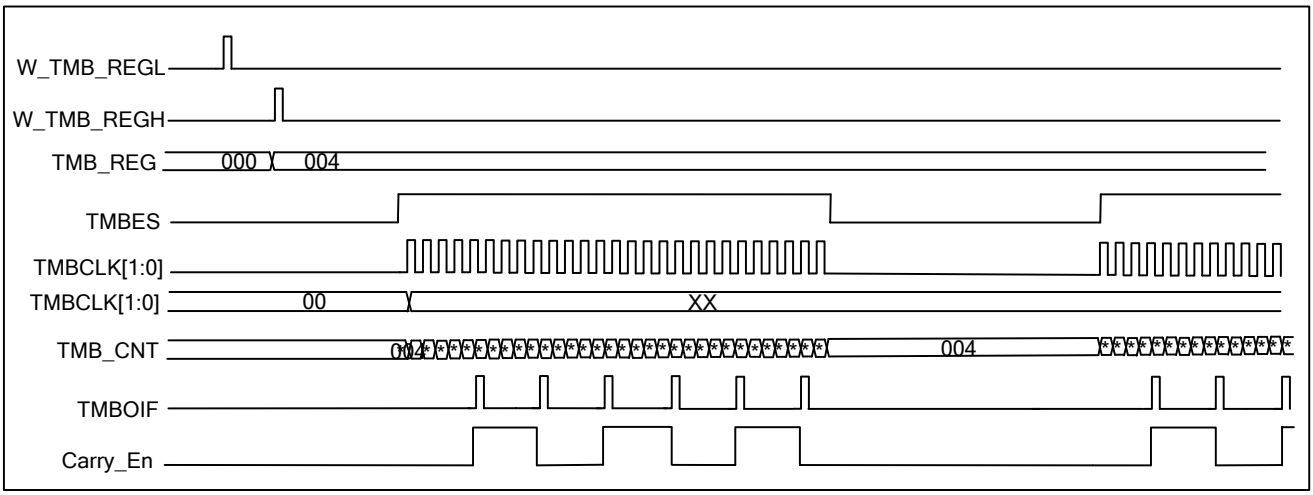


Figure 5-35 The Waveform of mode 1 Timer B

Mode 1 Timer B Control Register (P_TMB_CTRL, $0022)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBES	-	TMBCLK1	TMBCLK0	-	-	-	-
ACCESS	R/W	-	R/W	R/W	-	-	-	-
DEFAULT	0	-	0	0	-	-	-	-

Bit [7] **TMBES**: Timer B enable/disable control selected bit.

0 : disable (C_TMBES_DIS)

1 : enable (C_TMBES_EN)

Bit [6] **Reserved**

Bit [5:4] **TMBCLK[1：0]**: Timer B clock source selected bits

00 : Fosc (C_TMBCLK_1)

01 : Fosc/4 (C_TMBCLK_4)

10 : Fosc/64 (C_TMBCLK_64)

11 : TMACRR (C_TMBCLK_TMACRR)

Bit [3:0] **Reserved**

Mode 1 Timer B Count low 8-bit Register (P_TMB_CTRL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBCNTL7	TMBCNTL6	TMBCNTL5	TMBCNTL4	TMBCNTL3	TMBCNTL2	TMBCNTL1	TMBCNTL0
ACCESS	R	R	R	R	R	R	R	R
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **TMBCNTL**[7：0]: Timer B Count low byte 8-bit value for the counter.

Mode 1 Timer B low 8-bit data Register (P_TMB_REGL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBREGL7	TMBREGL6	TMBREGL5	TMBREGL4	TMBREGL3	TMBREGL2	TMBREGL1	TMBREGL0
ACCESS	W	W	W	W	W	W	W	W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **TMBREGL**[7：0]: Timer B low byte 8-bit pre-value for the counter.

Mode 1 Timer B Count high 4-bit Register (P_TMB_CNTH, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBCNTH3	TMBCNTH2	TMBCNTH1	TMBCNTH0
ACCESS	-	-	-	-	R	R	R	R
DEFAULT	-	-	-	-	0	0	0	0

Bit [7:4] **Reserved**

Bit [3:0] **TMBCNTH**[3：0]: Timer B Count High byte 4-bit value for the counter.

Mode 1 Timer B high 4-bit data Register (P_TMB_CTRL, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBREGH3	TMBREGH2	TMBREGH1	TMBREGH0
ACCESS	-	-	-	-	R	R	R	R
DEFAULT	-	-	-	-	0	0	0	0

Bit [7:4] **Reserved**

Bit [3:0] **TMBREGH**[3：0]: Timer B High byte 4-bit pre-value for the counter.

[Example] 5-16 Set Timer B select timer A carrier signal as counter clock.

```
LDA     #C_TIMAB_DN + #C_PWM_EN
STA     P_TIM_SEL                       ; set timer as down count and enable PWM output function
LDA     #$FC                            ; Before starting timer, set Timer B counter initial value first
STA     P_TMB_CNTL                      ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMB_CNTH                      ; set high 4-bit pre-value
LDA     #C_TMBES_EN + #C_TMBCLK_TMACRR
STA     P_TMB_CTRL                      ;Set clock source for TMA_Carrier
```

5.9. LVD (Low Voltage Detect)

A real time low voltage detector is built-in GPM6P1004B/1001B for user to sense the VDD voltage.

LVD Control Register (P_LVD_CTRL, $0033)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	LVDEN	R/0	LVDS	R/0	R/0	R/0	R/0	LVD
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
DEFAULT	00h							

Bit [7] **LVDEN[7]:** Low voltage detector enable control.

0: LVD disable.

1: LVD enable.

Bit [5] **LVDS:** Low voltage detect level selection.

0: LVD trigger H when VDD < 2.0V .

1: LVD trigger H when VDD < 2.4V .

Bit [0] **LVD:** Low voltage detect, read only.

0: VDD is upper 2.0V / 2.4V

1: VDD is under 2.0V / 2.4V

5.10. IR Transfer Module

IR TX is an analog block of GPM6P1004B/1001B, which can drive LED by TX. **TX PWM driving current control** register can control this block. Users can adjust PWM driving ability by setting the value of P_PWM_DRV [3] (PWMDRV0). TMAPWM signal as shown in Figure 5-35 controls the LED driving MOS. When Timer A is in PWM mode, it can generate PWM signal, and the PWM duty, frequency, on/off switch can be accurately controlled by timer A. The Envelope PWM signal can be generated by timer A & timer B. And it has been illustrated in timer instruction.

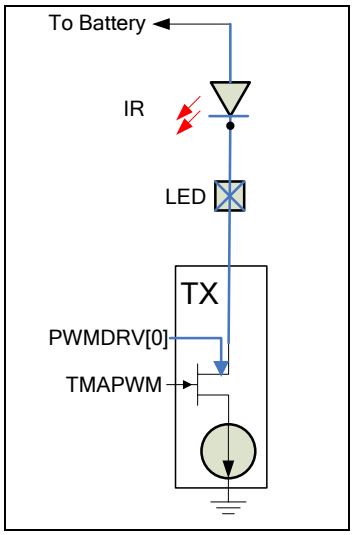


Figure 5-35 IR TX module diagram

TX PWM Driving Current Control Register (P_PWM_DRV, $0011)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	PWMDRV0	-	-	-
ACCESS	-	-	-	-	W	-	-	-
DEFAULT	-	-	-	-	0	-	-	-

Bit [7:4] Reserved

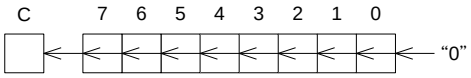
Bit [3] **PWMDRV**0: PWM driving current select.

0 = 1/2 driving current (C_PWMDRV_1)

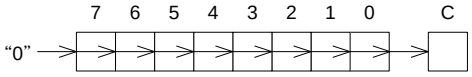
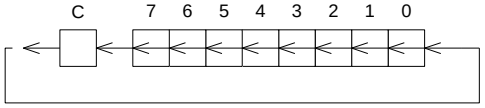
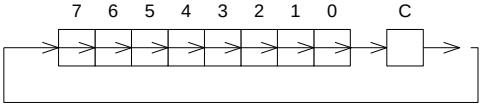
1 = 2/2 driving current (C_PWMDRV_2)

Bit [2:0] Reserved

5.11. Alphabetical List of Instruction Set

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
1.	ADC #dd	69	2	2	Add to accumulator with carry. A ← (A) + (M) + C If D-flag set to 1, the ADC performs decimal operation.	NV--D-ZC
2.	ADC aa	65	2	3		
3.	ADC aa, X	75	2	4		
4.	ADC aaaa	6D	3	4		
5.	ADC aaaa,X	7D	3	4(A)		
6.	ADC aaaa,Y	79	3	4(A)		
7.	ADC (aa,X)	61	2	6		
8.	ADC (aa), Y	71	2	5(A)		
9.	AND #dd	29	2	2	And memory data with accumulator. A ← (A) ^ (M)	N-----Z-
10.	AND aa	25	2	3		
11.	AND aa, X	35	2	4		
12.	AND aaaa	2D	3	4		
13.	AND aaaa,X	3D	3	4(A)		
14.	AND aaaa,Y	39	3	4(A)		
15.	AND (aa,X)	21	2	6		
16.	AND (aa), Y	31	2	5(A)		
17.	ASL A	0A	1	2	Arithmetic Shift Left 	N-----ZC
18.	ASL aa	06	2	5		
19.	ASL aa,X	16	2	6		
20.	ASL aaaa	0E	3	6		
21.	ASL aaaa,X	1E	3	6(A)		
22.	BCC aa	90	2	2(C)	Branch if carry bit clear If (C) = 0, then pc ← (pc) + xx	--------
23.	BCS aa	B0	2	2(C)	Branch if carry bit set If (C) = 1, then pc ← (pc) + xx	--------
24.	BEQ aa	F0	2	2(C)	Branch if equal If (Z) = 1, then pc ← (pc) + xx	--------
25.	BIT aa	24	2	3	Test bit in memory with accumulator Z ← (A) ^ (M), N ← (M_7), V ← (M_6)	NV----Z-
26.	BIT aaaa	2C	3	4		
27.	BMI aa	30	2	2(C)	Branch if minus If (N) = 1, then pc ← (pc) + xx	--------
28.	BNE aa	D0	2	2(C)	Branch if not equal If (Z) = 0, then pc ← (pc) + xx	--------
29.	BPL aa	10	2	2(C)	Branch if plus If (N) = 0, then pc ← (pc) + xx	--------
30.	BRK	00	1	7	Software interrupt If (B) = 1, then pc ← (pc) + 1	---B-I--
31.	BVC aa	50	2	2(C)	Branch if overflow bit clear If (V) = 0, then pc ← (pc) + xx	--------
32.	BVS aa	70	2	2(C)	Branch if overflow bit set If (V) = 1, then pc ← (pc) + xx	--------
33.	CLC	18	1	2	Clear C-flag：C ← "0"	-------0
34.	CLD	D8	1	2	Clear D-flag：D ← "0"	----0---

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
35.	CLI	58	1	2	Clear I-flag : I ← "0"	-----0--
36.	CLV	B8	1	2	Clear V-flag : V ← "0"	-0------
37.	CMP #dd	C9	2	2	Compare memory data with accumulator,	N-----ZC
38.	CMP aa	C5	2	3	(A) – (M)	
39.	CMP aa, X	D5	2	4		
40.	CMP aaaa	CD	3	4		
41.	CMP aaaa,X	DD	3	4(A)		
42.	CMP aaaa,Y	D9	3	4(A)		
43.	CMP (aa,X)	C1	2	6		
44.	CMP (aa), Y	D1	2	5(A)		
45.	CPX #dd	E0	2	2	Compare memory data with X-register,	N-----ZC
46.	CPX aa	E4	2	3	(X) – (M)	
47.	CPX aaaa	EC	3	4		
48.	CPY #dd	C0	2	2	Compare memory data with Y-register,	N-----ZC
49.	CPY aa	C4	2	3	(Y) – (M)	
50.	CPY aaaa	CC	3	4		
51.	DEC aa	C6	2	5	Decrement	N-----Z-
52.	DEC aa, X	D6	2	6	M ← (M) - 1	
53.	DEC aaaa	CE	3	6		
54.	DEC aaaa,X	DE	3	7		
55.	DEX	CA	1	2		
56.	DEY	88	1	2		
57.	EOR #dd	49	2	2	Exclusive OR	N-----Z-
58.	EOR aa	45	2	3	A ← (A) ⊕ (M)	
59.	EOR aa, X	55	2	4		
60.	EOR aaaa	4D	3	4		
61.	EOR aaaa,X	5D	3	4(A)		
62.	EOR aaaa,Y	59	3	4(A)		
63.	EOR (aa,X)	41	2	6		
64.	EOR (aa), Y	51	2	5(A)		
65.	INC aa	E6	2	5	Increment	N-----Z-
66.	INC aa, X	F6	2	6	M ← (M) + 1	
67.	INC aaaa	EE	3	6		
68.	INC aaaa,X	FE	3	7		
69.	INX	E8	1	2	X ← X + 1	N-----Z-
70.	INY	C8	1	2	Y ← Y + 1	N-----Z-
71.	JMP aaaa	4C	3	3	Unconditional jump	--------
72.	JMP (aaaa)	6C	3	6	Pc ← jump address	
73.	JSR aaaa	20	3	6	Jump to subroutine (sp) ← (pc_H), sp ← sp – 1, (sp) ← (pc_L), sp ← sp – 1, pc ← aaaa	--------
74.	LDA #dd	A9	2	2	Load accumulator	N-----Z-
75.	LDA aa	A5	2	3	A ← (M)	
76.	LDA aa, X	B5	2	4		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
77.	LDA aaaa	AD	3	4		
78.	LDA aaaa,X	BD	3	4(A)		
79.	LDA aaaa,Y	B9	3	4(A)		
80.	LDA (aa,X)	A1	2	6		
81.	LDA (aa), Y	B1	2	5(A)		
82.	LDX #dd	A2	2	2	Load X-register	
83.	LDX aa	A6	2	3	X ← (M)	
84.	LDX aa, Y	B6	2	4		N-----Z-
85.	LDX aaaa	AE	3	4		
86.	LDX aaaa,Y	BE	3	4(A)		
87.	LDY #dd	A0	2	2	Load Y-register	
88.	LDY aa	A4	2	3	Y ← (M)	
89.	LDY aa, X	B4	2	4		N-----Z-
90.	LDY aaaa	AC	3	4		
91.	LDY aaaa,X	BC	3	4(A)		
92.	LSR A	4A	1	2	Logical shift right	
93.	LSR aa	46	2	5		
94.	LSR aa, X	56	2	6		N-----ZC
95.	LSR aaaa	4E	3	6		
96.	LSR aaaa,X	5E	3	6(A)		
97.	NOP	EA	1	2	No operation	--------
98.	ORA #dd	09	2	2	Logical OR	
99.	ORA aa	05	2	3	A ← (A) v (M)	
100.	ORA aa, X	15	2	4		
101.	ORA aaaa	0D	3	4		
102.	ORA aaaa,X	1D	3	4(A)		N-----Z-
103.	ORA aaaa,Y	19	3	4(A)		
104.	ORA (aa,X)	01	2	6		
105.	ORA (aa), Y	11	2	5(A)		
106.	PHA	48	1	3	(sp) ← A, sp ← sp - 1	--------
107.	PHP	08	1	3	(sp) ← P status, sp ← sp -1	
108.	PLA	68	1	4	sp ← sp +1, A ← (sp)	--------
109.	PLP	28	1	4	Sp ← sp +1, P status ← (sp)	restored
110.	ROL A	2A	1	2	Rotate left through carry	
111.	ROL aa	26	2	5		
112.	ROL aa, X	36	2	6		N-----ZC
113.	ROL aaaa	2E	3	6		
114.	ROL aaaa,X	3E	3	6(A)		
115.	ROR A	6A	1	2	Rotate right through carry	
116.	ROR aa	66	2	5		
117.	ROR aa, X	76	2	6		N-----ZC
118.	ROR aaaa	6E	3	6		
119.	ROR aaaa,X	7E	3	6(A)		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
120.	RTI	40	1	6	Return from interrupt Sp ← sp + 1, P status ← (sp), sp ← sp + 1, pc_L ← (sp), sp ← sp +1, pc_H ← (sp)	restored
121.	RTS	60	1	6	Return from subroutine Sp ← sp + 1, pc_L ← (sp), sp ← sp +1, pc_H ← (sp)	--------
122.	SBC #dd	E9	2	2	Subtract with carry A ← (A) – (M) - ~(C)	NV----ZC
123.	SBC aa	E5	2	3		
124.	SBC aa, X	F5	2	4		
125.	SBC aaaa	ED	3	4		
126.	SBC aaaa,X	FD	3	4(A)		
127.	SBC aaaa,Y	F9	3	4(A)		
128.	SBC (aa,X)	E1	2	6		
129.	SBC (aa), Y	F1	2	5(A)		
130.	SEC	38	1	2	Set C-flag：C ← “1”	-------1
131.	SED	F8	1	2	Set D-flag：D ← “1”	----1---
132.	SEI	78	1	2	Set I-flag：I ← “1”	-----1--
133.	STA aa	85	2	3	Store accumulator in memory (M) ← A	--------
134.	STA aa, X	95	2	4		
135.	STA aaaa	8D	3	4		
136.	STA aaaa,X	9D	3	5		
137.	STA aaaa,Y	99	3	5		
138.	STA (aa,X)	81	2	6		
139.	STA (aa), Y	91	2	6		
140.	STX aa	86	2	3	Store X-register in memory (M) ← X	--------
141.	STX aa, Y	96	2	4		
142.	STX aaaa	8E	3	4		
143.	STY aa	84	2	3	Store Y-register in memory (M) ← Y	--------
144.	STY aa, X	94	2	4		
145.	STY aaaa	8C	3	4		
146.	TAX	AA	1	2	Transfer accumulator to X-register：X ← A	N-----Z-
147.	TAY	A8	1	2	Transfer accumulator to Y-register：Y ← A	N-----Z-
148.	TSX	BA	1	2	Transfer sp to X-register：X ← sp	N-----Z-
149.	TXA	8A	1	2	Transfer X-register to accumulator：A ← X	N-----Z-
150.	TXS	9A	1	2	Transfer X-register to sp：sp ← X	N-----Z-
151.	TYA	98	1	2	Transfer Y-register to accumulator：A ← Y	N-----Z-

Notes:

1. Cycle (A): Cycle+1 when cross a boundary.
2. Cycle(C): Cycle+1 if the branch condition is true; Cycle+2 if the branch condition is true and cross a boundary.
3. where “xx” represents hexadecimal digit, 0 ~F.

6. ELECTRICAL CHARACTERISTICS

6.1. Absolute Maximum Ratings

Characteristics	Symbol	Ratings
DC Supply Voltage	V_+	< 5.0V
Input Voltage Range	V_{IN}	-0.5V to V_+ + 0.5V
Operating Temperature	T_A	0℃ to +70℃
Storage Temperature	T_{STO}	-50℃ to +150℃
Average PWM MAX Driving Current	I_{LEDM}	150mA
VDD Total MAX Current	I_{VDDM}	100mA
VSS Total MAX Current	I_{VSSM}	120mA

Note: Stresses beyond those given in the Absolute Maximum Rating table may cause permanent damage to the device. For normal operational conditions see AC/DC Electrical Characteristics.

6.2. AC Characteristics

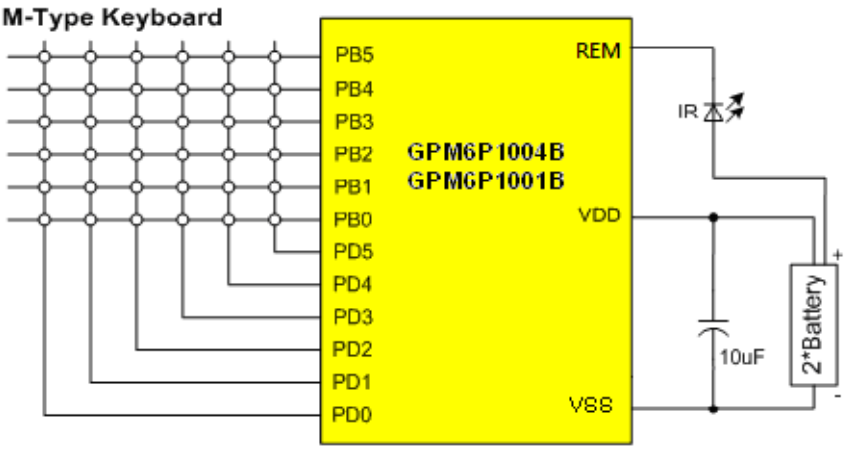
Characteristics	Limit			Unit	Test Condition
	Min.	Typ.	Max.		
Internal OSC Accuracy @ Freq=4MHz					
OSC Variation	-1.5	-	+1.5	%	VDD = 2.0V - 3.6V, T =0°C~60°C

6.3. DC Characteristics (VDD = 3.0V, T_A = 25℃)

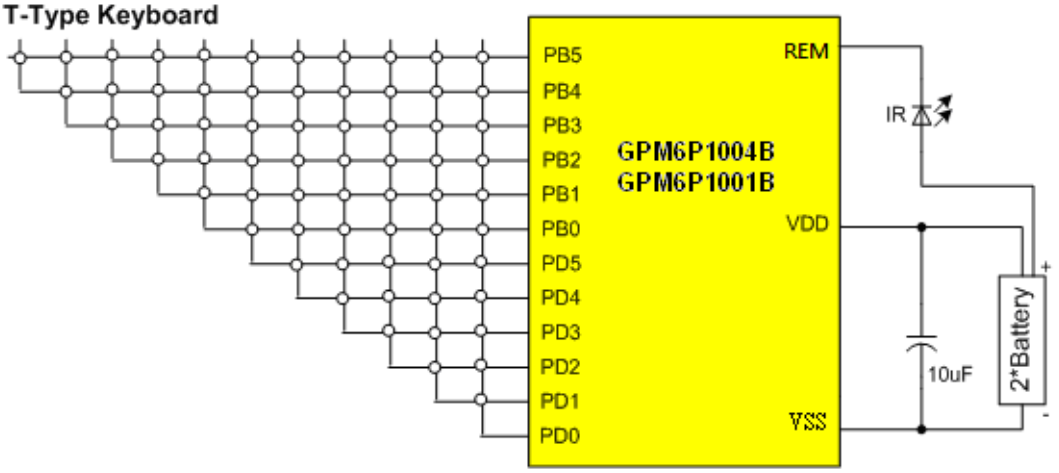
Characteristics	Symbol	Limit			Unit	Test Condition
		Min.	Typ.	Max.		
Operating Voltage	VDD	V_{LVR}	-	3.6	V	For 2-battery
Operating Current	I_{OP}	-	2.0	4.0	mA	F_{CPU} = 4.0MHz @ 3.6V, no load
Normal Standby Current	I_{STBY}	-	-	1.0	uA	VDD = 3.6V
Key scan Standby Current	I_{STBY_scan}	-	-	2.0	uA	VDD = 3.0V, no load
Input High Level	V_{IH}	0.7VDD	-	-	V	VDD = 3.0V
Input Low Level	V_{IL}	-	-	0.3VDD	V	VDD = 3.0V
Output High Level PB, PD	V_{OH}	0.8VDD	-	-	V	VDD = 3.0V I_{OH} = -6mA
Output Low Level PB, PD	V_{OL}	-	-	0.2VDD	V	VDD = 3.0V I_{OL} = 16mA
Input Pull High Resistor PB, PD	R_H	30	50	70	Kohm	Pull High VDD = 3.0V
Input Pull Low Resistor PB, PD	R_L	30	50	70	Kohm	Pull Low VDD = 3.0V
Max PWM Driving Current	I_{PWM}	150	-	-	mA	VDD = 3.0V, V_{REM} = 0.5V PWMDRV0=1
LVR Active Voltage	V_{LVR}	1.6	1.7	1.8	V	

7. APPLICATION CIRCUITS

M-Type Keyboard



T-Type Keyboard



8.PACKAGE/PAD LOCATIONS

8.1. Ordering Information

Product Number	Package Type
GPM6P1004B-NnnV-C	Chip form
GPM6P1001B-NnnV-C	Chip form
GPM6P1004B -NnnV-QS03x	Halogen Free Package
GPM6P1001B -NnnV-QS03x	Halogen Free Package
GPM6P1004B -NnnV-QS01x	Halogen Free Package
GPM6P1001B -NnnV-QS01x	Halogen Free Package

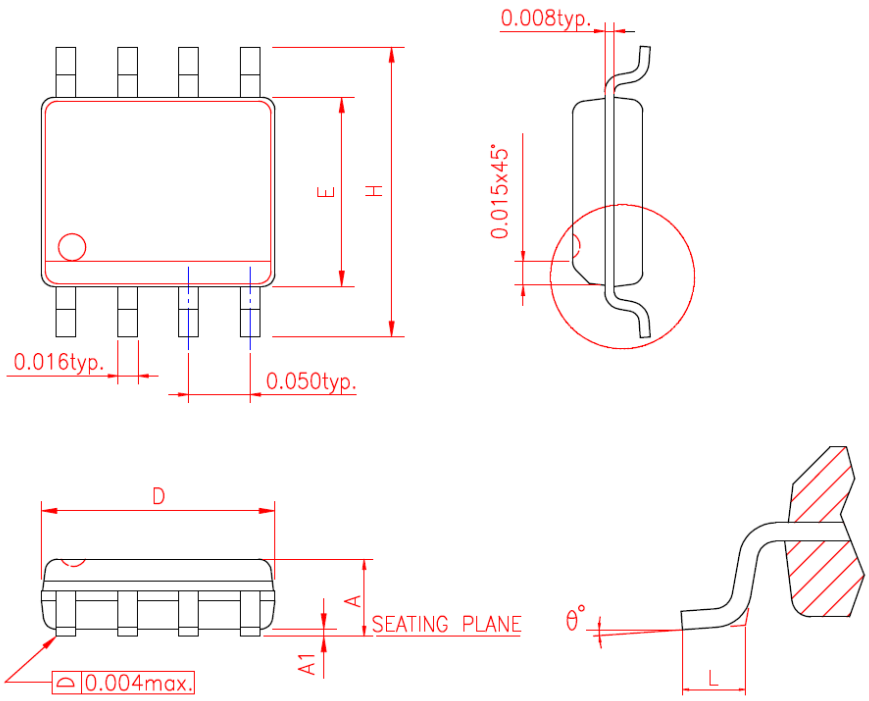
Note1: Code number is assigned for customer.

Note2: Code number (N = A - Z or 0 - 9, nn = 00 - 99); version (V = A - Z).

Note3: Package form number (x = 1 - 9, serial number).

8.2. Package Information

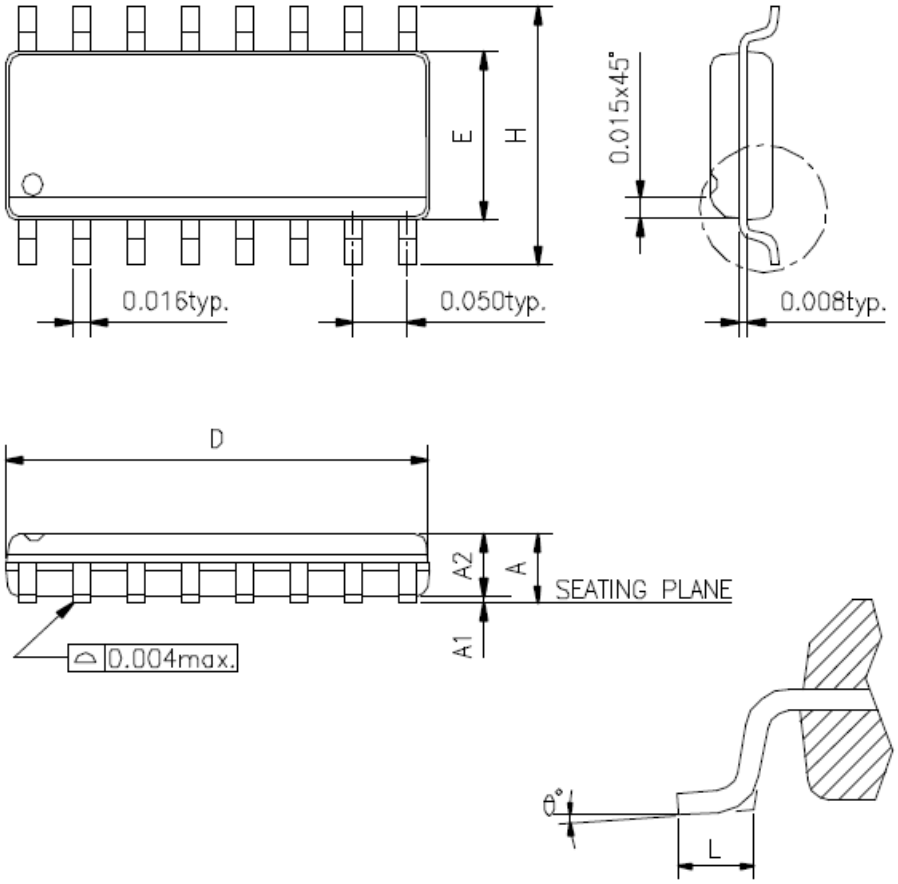
8.2.1. SOP8



Symbol	Min.	Max.
A	0.053	0.069
A1	0.004	0.010
D	0.189	0.196
E	0.150	0.157
H	0.228	0.244
L	0.016	0.050
θ°	0°	8°

UNIT: INCH

8.2.2. SOP16



Symbol	Min.	Max.
A	0.053	0.069
A1	0.004	0.010
D	0.386	0.394
E	0.150	0.157
H	0.228	0.244
L	0.016	0.050
θ°	0°	8°

UNIT: INCH

9. DISCLAIMER

The information appearing in this publication is believed to be accurate.

Integrated circuits sold by Generalplus Technology are covered by the warranty and patent indemnification provisions stipulated in the terms of sale only. GENERALPLUS makes no warranty, express, statutory implied or by description regarding the information in this publication or regarding the freedom of the described chip(s) from patent infringement. FURTHERMORE, GENERALPLUS MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE. GENERALPLUS reserves the right to halt production or alter the specifications and prices at any time without notice. Accordingly, the reader is cautioned to verify that the data sheets and other information in this publication are current before placing orders. Products described herein are intended for use in normal commercial applications. Applications involving unusual environmental or reliability requirements, e.g. military equipment or medical life support equipment, are specifically not recommended without additional processing by GENERALPLUS for such applications. Please note that application circuits illustrated in this document are for reference purposes only.

10. REVISION HISTORY

Date	Revision #	Description	Page
Jun 25, 2015	0.2	Update spec	54
Jul 21, 2014	0.1	Preliminary	54