DATA SHEET

Generalplus

GPM6P1816A

48-pin Remote Controller with 16KB OTP ROM

Preliminary

Mar. 05, 2013

Version 0.1

Table of Contents

PAGE

48-PIN REMOTE CONTROLLER WITH 16KB OTP ROM

1. GENERAL DESCRIPTION

The GPM6P1816A is a special chip for remote control with 256 bytes built-in SRAM and 16K bytes built-in OTP ROM. It includes three timers and up to 25 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 2.0V - 3.6V@4/8MHz. It has a sleep mode for power saving mode which retains the contents of RAM, but stops the oscillator and causes all other chip functions to be inoperative. Sleep mode can be released by using external wakeup sources. This device is capable for many application fields such as low power watch and other LCD related products. Meanwhile, the built-in IR transfer module can make IR control and usage easily.

2. FEATURES

- **CPU**
  - 151 instructions
  - 13 addressing modes
  - Up to 8MHz clock operation
- **Memories**
  - 16K bytes program OTP ROM
  - 256 bytes RAM including stack area
- **Reset Management**
  - Enhanced reset system
  - Power On Reset (POR)
  - Low Voltage Reset (LVR)
  - Watchdog Reset (WDR)
- **Interrupt Management**
  - 13 internal interrupts
- **I/O Ports**
  - Up to 25 bi-direction tri-state I/O ports:
    - ✧ PortA[7:0]: with programmable pull high/ pull low, share pads with SEGMENT[23:16].
    - ✧ PortB[1:0]: with programmable pull high/ pull low, share pads with INT[1:0].
    - ✧ PortB[2]: with programmable pull high/ pull low.
    - ✧ PortC[7:0]: with programmable pull high/ pull low, share pads with SEGMENT[31:24].
    - ✧ PortD[1:0]: with programmable pull high/ pull low, share pads with XTI, XTO.
    - ✧ PortD[3:2]: with programmable pull high/ pull low, share pads with X32O, X32I.
    - ✧ PortD[5]: with programmable pull high/ pull low, shares a pad with VPP.
    - ✧ PortD[7]: with programmable pull high/ pull low, shares a pad with REM.
  - I/O ports with 8mA current drive.
  - I/O ports with 16mA current sink
- **Clock Management**
  - Internal oscillator: 4MHz or 8MHz (selectable by code option) ±1.5%(typ), @ 2.0V~3.6V for system operation.
  - Crystal input: 4~ 8MHz @ 2.0V~3.6V for system operation
  - Internal oscillator: 32768Hz±3.5%(typ)@ 2.0V~3.6V for timebase.
  - Crystal input: 32768Hz @ 2.0V~3.6V for timebase.
- **Power Management**
  - Three power saving modes: Sleep , Green, Halt modes
- **RFC** (Resister to Frequency Converter)
  - 12-bit timer (Timer B) counter.
  - PC7 as RFC input, PC6/5/4 as RFC output.
- **LCD** configuration
  - Frame rate: 85Hz.
  - 4 commons x 32 segments (MAX).
  - LCD 1/2, 1/3 bias; 1/2, 1/3, 1/4 duty; VLCD = VDD.
- **Low Voltage Reset**
  - LVR: Low Voltage Reset (V_{LVR}=1.85V ± 0.15V or V_{LVR}=2.25V ± 0.15V by option).
- **Four timers**
  - Watchdog timer: basic timer provides Fosc/4194304 source.
  - Timer A provides 15Hz~1MHz frequency controllable duty cycle with carrier signal in PWM mode.
  - Timer B is a 12-bit general purpose timer with selectable input clock.
  - 32KHz timer is a time base wakeup source with selectable frequency.
- **LVD** (Low voltage detect)
  - Sense VDD voltage@ 2.1V / 2.4V (register option)
- **Wake up Source**
  - Key change wake-up from Sleep / Halt / Green mode
  - 32768Hz time base wakeup(TBHF/TBLF)
- **IR**
  - Built-in IR TX drives IR LED with up to 200mA driving capability @ VDD=3.0V & V_{RMT}=3.0V.

[Table] 2-1 GPM6P1816A Configuration

Part NO.	ROM Type	Voltage (V)	Speed (MHz)	ROM (Byte)	RAM (Byte)	IR Tx/Rx	CCP			CPU OSC.		IO No.	PKG
							CAP	CNT	PWM	INT	XTAL		
GPM6P1816A	OTP	2.4~3.6	8	16K	256	Tx	-	1	1	•	•	25	LQFP48
		2.0~3.6	4										

3. BLOCK DIAGRAM

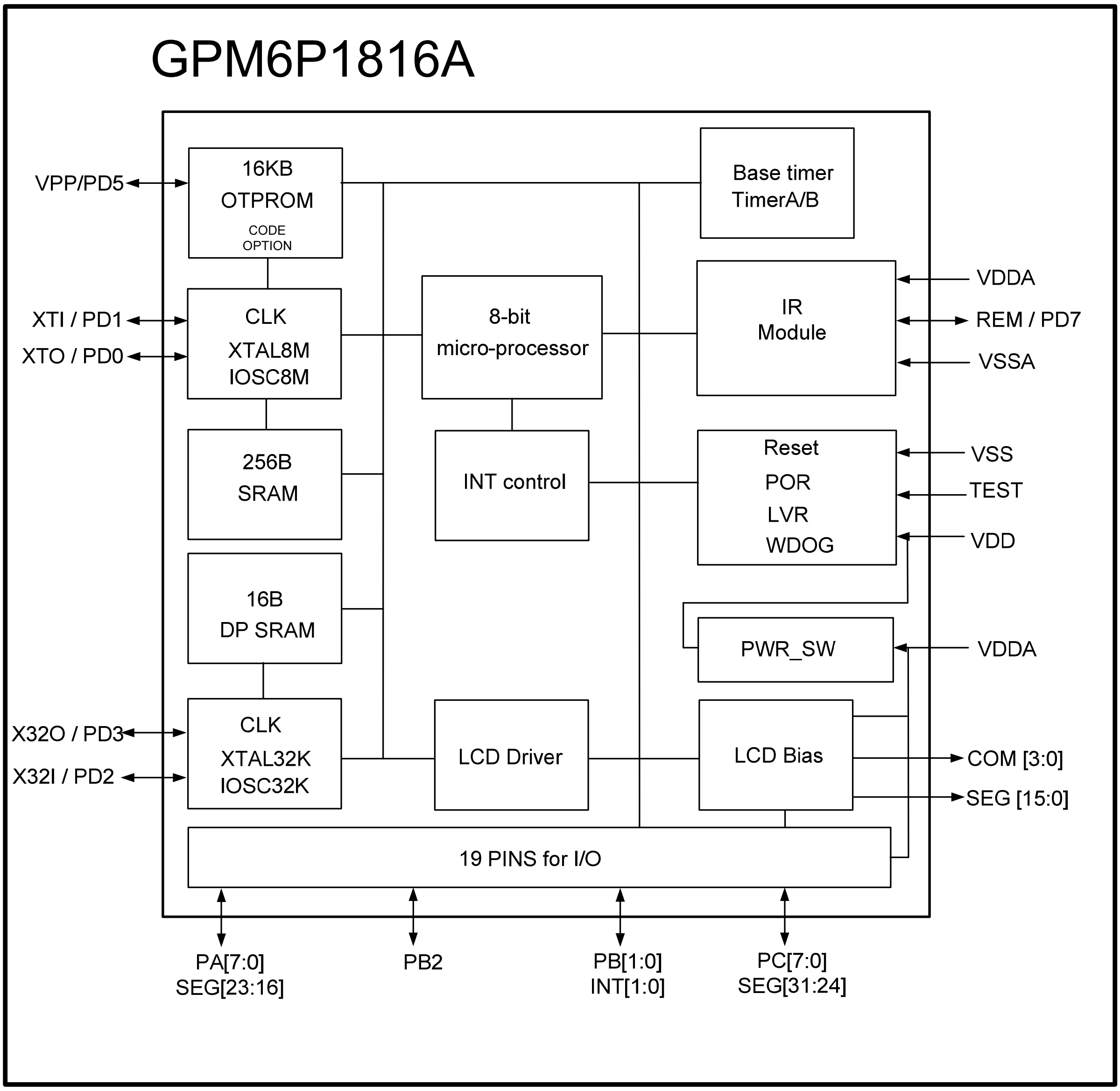


Figure 3-1 The block diagram of GPM6P1816A

4. SIGNAL DESCRIPTIONS

4.1. Pin Description

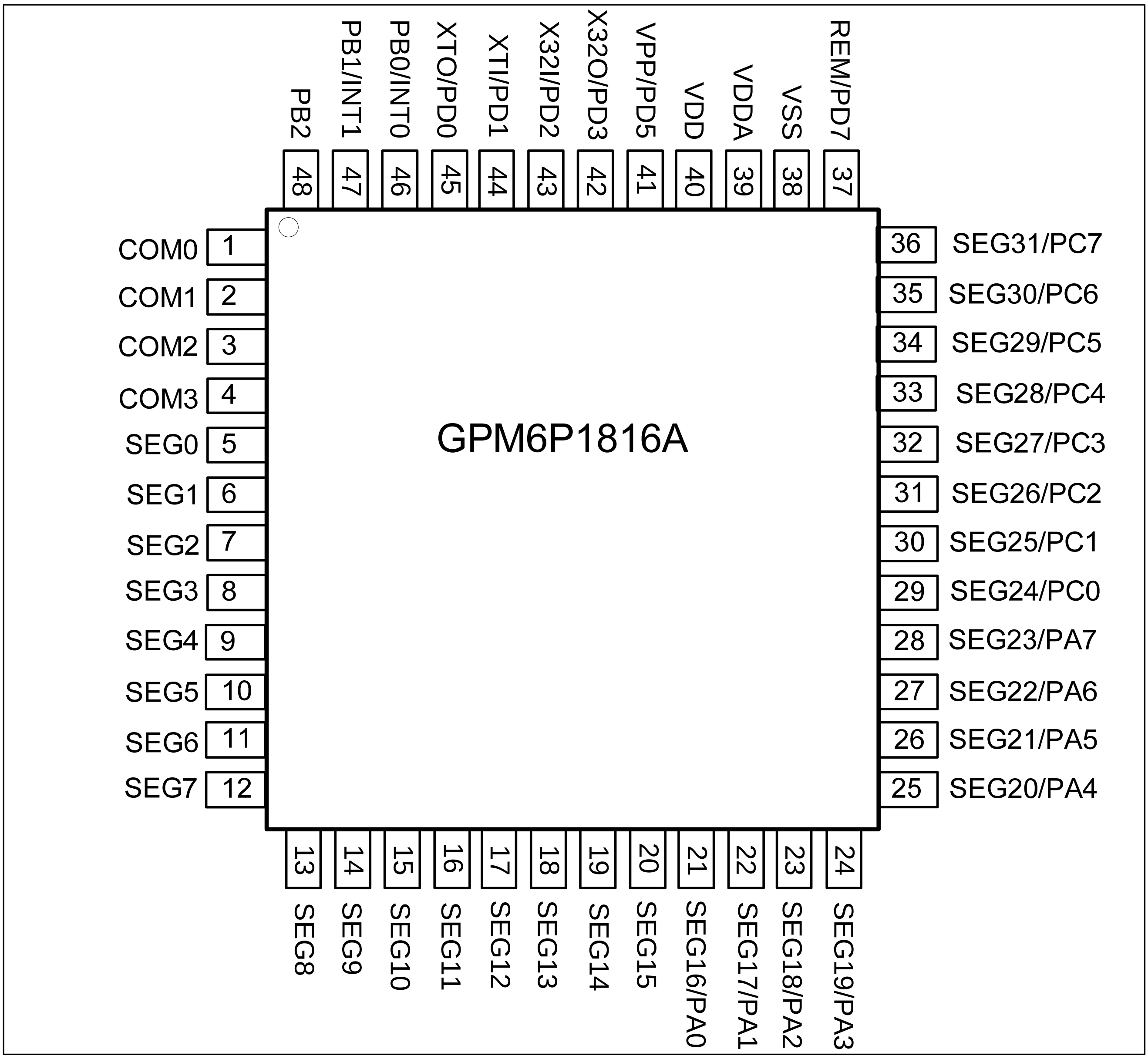
Type: I = Input, O = Output, S = Supply

Pin Name	Dice Pin No.	PKG Pin No.	Type	Main Function	Alternate Function
COM[0:3]	1~4	1~4	O	LCD driver common output	
SEG[0:15]	5~20	5~20	O	LCD driver segment output	
PA0/SEG16	21	21	I/O	**PortA[7:0]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **SEG[23:16]:** LCD driver segment output	
PA1/SEG17	22	22	I/O		
PA2/SEG18	23	23	I/O		
PA3/SEG19	24	24	I/O		
PA4/SEG20	25	25	I/O		
PA5/SEG21	26	26	I/O		
PA6/SEG22	27	27	I/O		
PA7/SEG23	28	28	I/O		
PC0/SEG24	29	29	I/O	**PortC[7:0]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup**; If a key is changed, the chip can be awakened from sleep mode. **SEG[31:24]:** LCD driver segment output	
PC1/SEG25	30	30	I/O		
PC2/SEG26	31	31	I/O		
PC3/SEG27	32	32	I/O		
PC4/SEG28	33	33	I/O		
PC5/SEG29	34	34	I/O		
PC6/SEG30	35	35	I/O		
PC7/SEG31	36	36	I/O		
REM/PD7	37	37	I/O	**PortD[7]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be wakened from sleep mode. **REM:** Remote IR signal transmit pin.	
VSS	38	38	S	Ground	
VDDA	39	39	S	power supply	
VDD	40	40	S	power supply	
VPP/PD5	41	41	I/O	**PortD[5]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output Low. **Normal wakeup;** if a key is changed, the chip can be wakened from sleep mode. **VPP:** OTP Program power supply	
X32O/PD3	42	42	I/O	**PortD[3]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be awakened from sleep mode. **Crystal Output:** It is connected with external crystal for 32K crystal oscillation circuitry in crystal mode.	

Pin Name	Dice Pin No.	PKG Pin No.	Type	Main Function	Alternate Function
X32I/PD2	43	43	I/O	**PortD[2]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be awakened from sleep mode. **Crystal Input:** It is connected with external crystal for 32K crystal oscillation circuitry in crystal mode.	
TEST	44	NC	I	**Test pin.**	
XTI/PD1	45	44	I/O	**PortD[1]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be awakened from sleep mode. **Crystal Input:** It is connected with external crystal for 4M/8M crystal oscillation circuitry in crystal mode.	
XTO/PD0	46	45	I/O	**PortD[0]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be awakened from sleep mode. **Crystal Output:** It is connected with external crystal for 4M/8M crystal oscillation circuitry in crystal mode.	
PB0/INT0	47	46	I/O	**PortB[1:0]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be awakened from sleep mode. **INT[1:0]:** external INT input.	
PB1/INT1	48	47	I/O		
PB2	49	48	I/O	**PortB[2]:** Bi-directional programmable Input/Output port. It can be configured as pull_high resistor, pull_low resistor, floating input or CMOS output. **Normal wakeup;** if a key is changed, the chip can be awakened from sleep mode.	
NC	50	NC		**Reserve.**	

4.2. PIN Assignment (Top View)

4.2.1. LQFP48 Package for GPM6P1816A



5. FUNCTIONAL DESCRIPTIONS

5.1. Central Processing Unit

5.1.1. CPU Introduction

The microprocessor of GPM6P1816A 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 can be varied up to 8.0MHz depending on the application.

5.1.2. CPU Register

The CPU has six registers: 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.

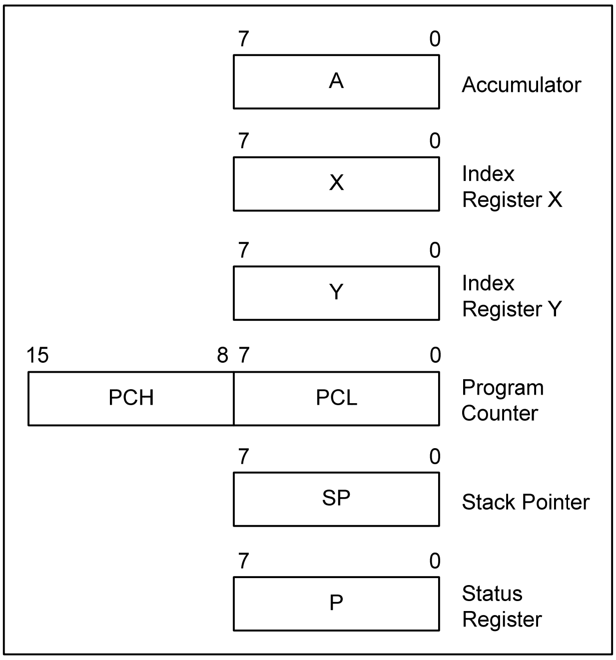


Figure 5-1 System registers

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 transferring function can be used in X and Y registers.

Accumulator

The Accumulation is an 8-bit general-purpose register, which can be operated with functions such as transferring, 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.

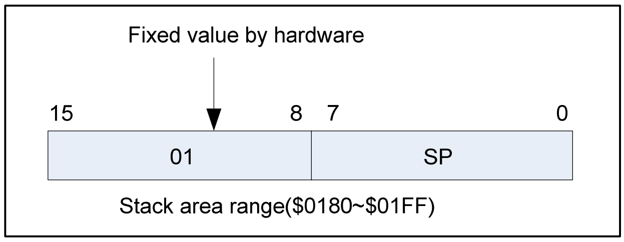


Figure 5-2 Stack point register

[Example] 5-1Initialized 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 Register (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 CPU6502 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 execute some operations, e.g. branch condition or bit operation.

❑ **Overflow flag bit**

This flag indicates whether the overflow has occurred 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 which 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 change modes.

❑ **Interrupt disable flag**

This bit can enable or disable all interrupt 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'. In contrast, 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.

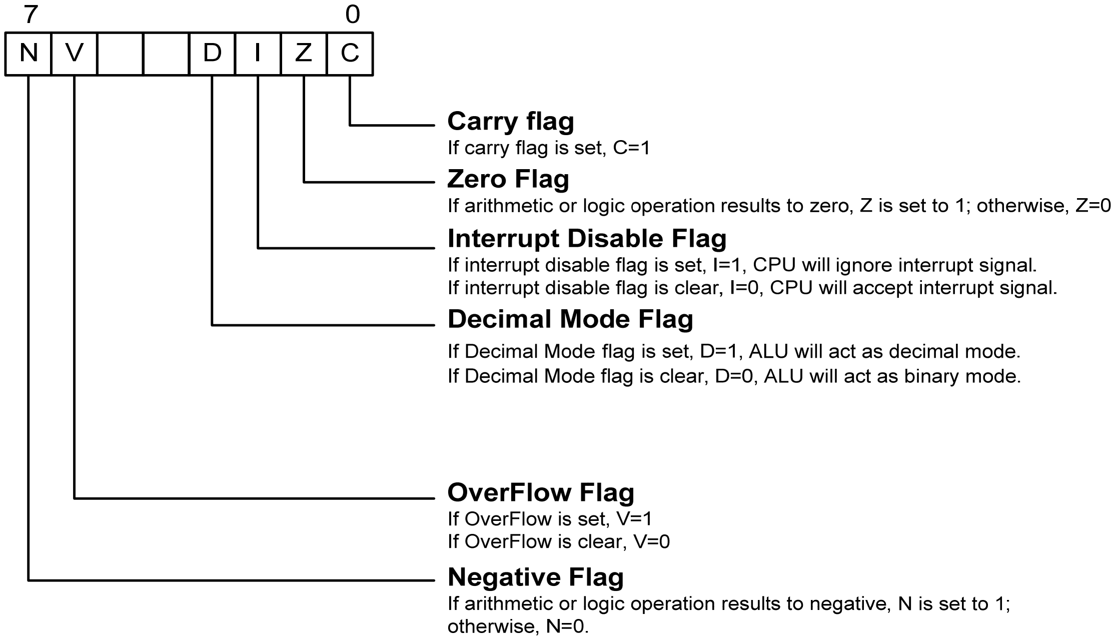


Figure 5-3 Status register

5.2. Memory Organization

5.2.1. Introduction

The GPM6P1816A has separated address spaces for program memory and data memory. Program memory can be read only. It contains up to 16K bytes of program memory. Data memory that contains 256 bytes of RAM including stack area can be read and written.

5.2.2. Memory Space

Memory address allocations on the GPM6P1816A are divided into several parts. 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 to set up this register and help to reduce program size. The memory mapping, please check the following figures:

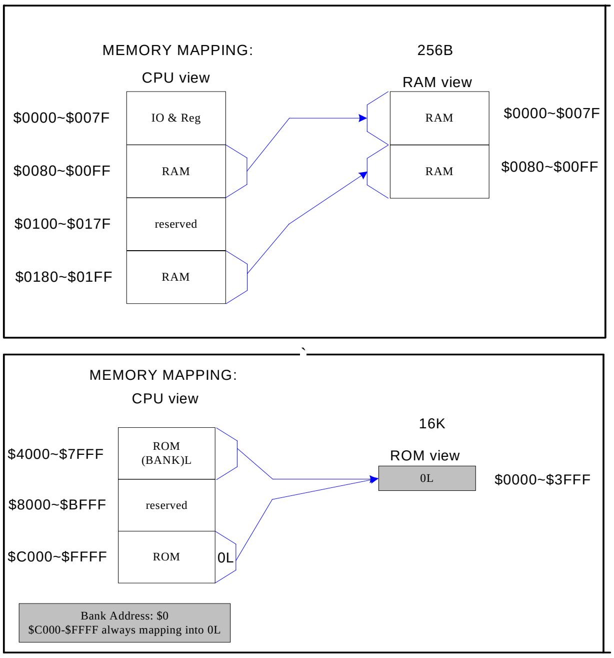


Figure 5-4 shows GPM6P1816A memory map.

The address of NMI, $\overline{\text{RESET}}$ and IRQ exception vectors are located from $FFFA to $FFFF. The exception vectors should be specified in the program to have proper operation.

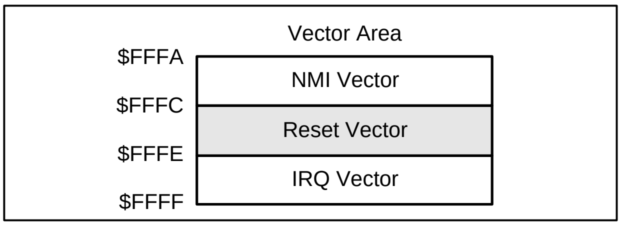


Figure 5-5 Interrupt vector area

[Example] 5-2 Interrupt vector table in software

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

5.2.3. SRAM protect

SRAM protect function can protect SRAM data from writing unexpectedly. Refer to SRAM protection register, as long as any bank is set 1, the bank cannot execute writing function any more only when the register is cleared to 0 again.

Considering the stack region should not be protected, the SRAM protection is in the way of bank controlled. And for more information about the bank mapping address, please refer to the SRAM protect Register.

SRAM protect Register (P_SRAMP_CTRL, $003F)

BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	--	--	BANK5	BANK4	--	--	BANK1	BANK0
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 [7:6] no function

Bit [5] **BANK5**: $1C0H ~ $1FFH

0 = SRAM protect disable
1 = SRAM protect enable

Bit [4] **BANK4**: $180H ~ $1BFH
0 = SRAM protect disable
1 = SRAM protect enable

Bit [3:2] no function

Bit [1] **BANK1**: $C0H ~ $FFH
0 = SRAM protect disable
1 = SRAM protect enable

Bit [0] **BANK0**: $80H ~ $BFH
0 = SRAM protect disable
1 = SRAM protect enable

5.2.4. Configuration Register

The configuration register is used to setup the operation condition. And its CPU view address is $FFF8 & $FFF9. It is mapped to the special reserved ROM address $3FF8 & $3FF9 (for 16K ROM); GPM6P1816A has the following configuration options.

- Crystal resonator or internal oscillator clock source option for system operation clock.
- Crystal resonator or internal oscillator clock source option for time-base clock.
- LVR enable or disable option.
- LVR trigger voltage 1.85V or 2.25V selection option.
- Watchdog enable or disable option.
- IOSC frequency 4MHz or 8MHz selection option.

Device Configuration Register (OPCODE0, $FFF8)

BIT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	Reserve	Reserve	Reserve	SECURITY	32KCLK	WDTENB	LVRENB	SYSCLKS
Access	R	R	R	R	R	R	R	R
Default	1	1	1	1	1	1	1	1

Bit [7:5] Reserved.

Bit [4] **SECURITY:** disable/enable security protection.
1: Security disable
0: Security enable

Bit [3] **32KCLK:** IOSC (internal) / Crystal selection.
0= IOSC
1= Crystal

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

Device Configuration Register (OPCODE1, $FFF9)

BIT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LVRVSEL	IOSCFSEL
Access	R	R	R	R	R	R	R	R
Default	1	1	1	1	1	1	1	1

Bit [7:2] Reserved.

Bit [1] **LVRVSEL:** LVR trigger voltage selection
0= LVR trigger voltage is 1.85V
1= LVR trigger voltage is 2.25V

Bit [0] **IOSCFSEL:** IOSC frequency selection
0= IOSC frequency is 4MHz
1= IOSC frequency is 8MHz

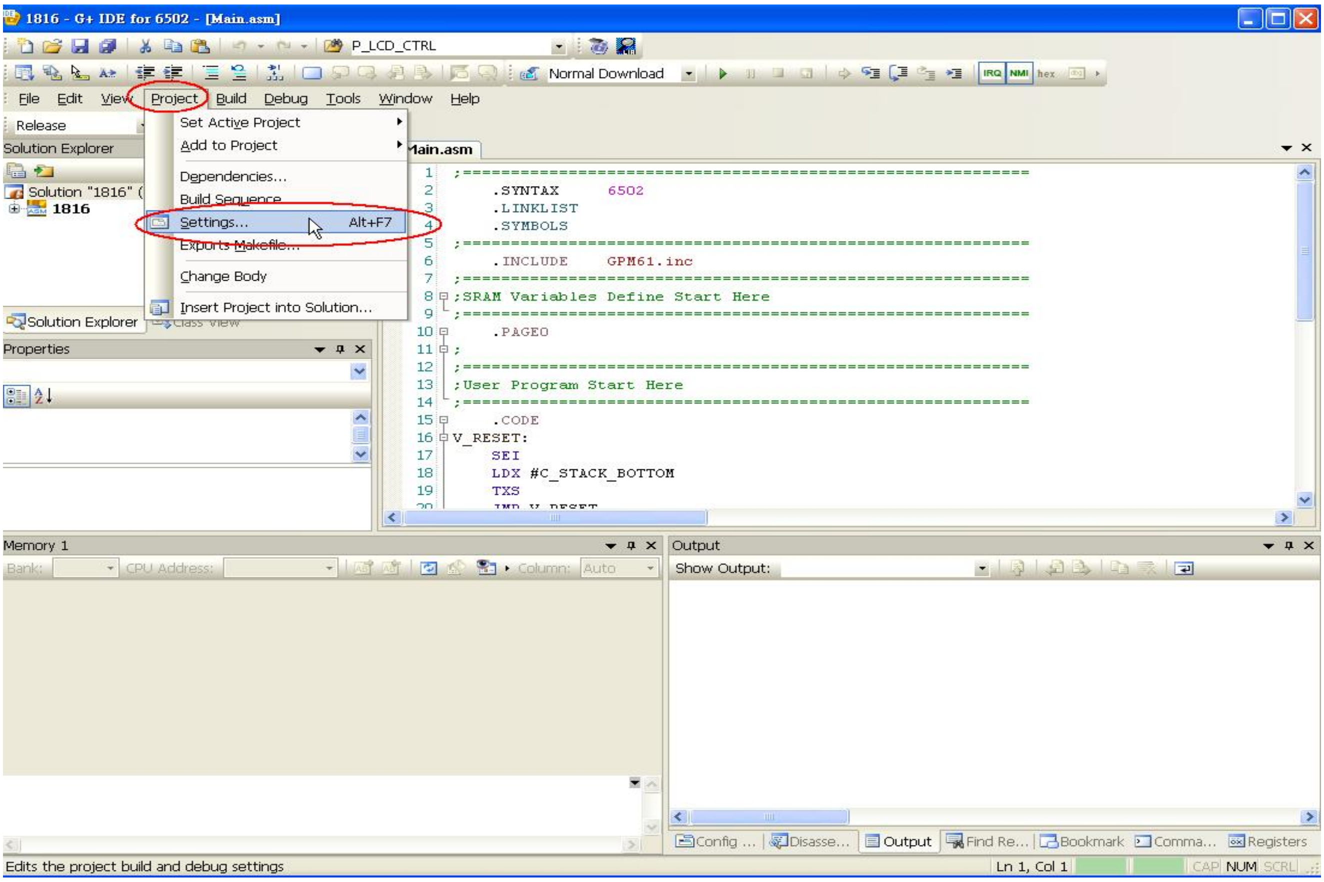


Figure 5-6 Device Configuration Register set in G+ IDE

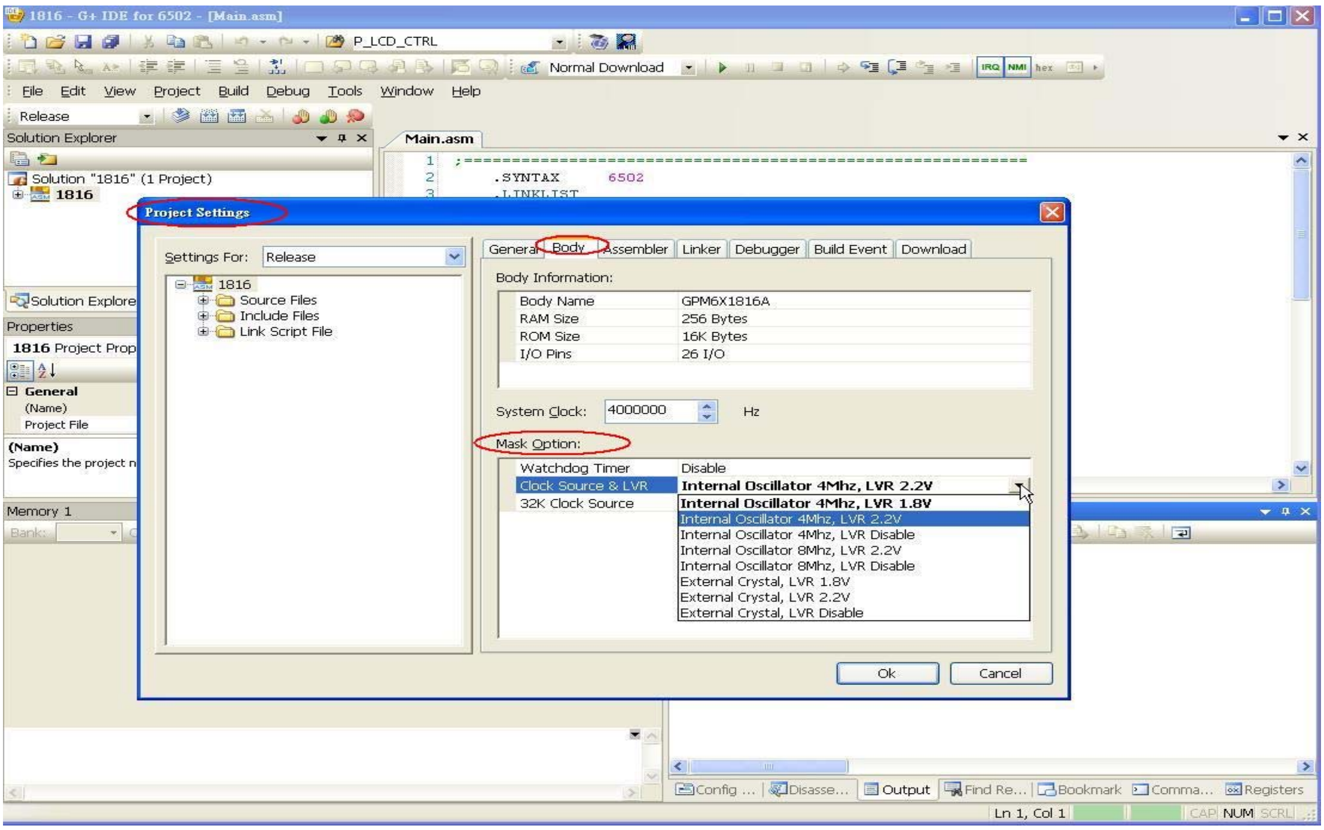


Figure 5-7 Device Configuration Register set in G+ IDE

5.2.5. Special Function Registers (SFR)

GPM6P1816A has many control registers. All of the control registers are used by MCU and peripheral function block for controlling the desired operations. 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 are no any effects 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.

GPM6P1816A Special Function Registers Description

$0000~$000F: 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	R/0	R/0	R/0	Port B direction		
$01	P_IOC_DIR	00h	R/W	Port C direction							
$02	P_IOD_DIR	00h	R/W	PD7 direction	R/0	PD5 direction	R/0	Port D direction			
$03	P_IOA_ATT	00h	R/W	Port A attribute							
$04	P_IOB_ATT	00h	R/W	R/0	R/0	R/0	R/0	R/0	Port B attribute		
$05	P_IOC_ATT	00h	R/W	Port C attribute							
$06	P_IOD_ ATT	00h	R/W	PD7 attribute	R/0	PD5 attribute	R/0	Port D attribute			
$07	P_IOA_DAT	00h	W	Port A Data							
$08	P_IOB_ DAT	00h	W	R/0	R/0	R/0	R/0	R/0	Port B Data		
$09	P_IOC_ DAT	00h	W	Port C Data							
$0A	P_IOD_ DAT	00h	W	PD7 Data	R/0	PD5 Data	R/0	Port D Data			
$0B	P_IOA_ DAT	00h	R/W	Port A direction							
$0C	P_IOA_DATBF	00h	R/W	Port A Data Buffer							
$0D	P_IOB_ DATBF	00h	R/W	R/0	R/0	R/0	R/0	R/0	Port B Data Buffer		
$0E	P_IOC_ DATBF	00h	R/W	Port C Data Buffer							
$0F	P_IOD_ DATBF	00h	R/W	PD7 Buffer	R/0	PD5 Buffer	R/0	Port D Data Buffer			

$0010~$001E: 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	--	PWMDRV1	PWMDRV0	--	--	--
	P_REM_S					REMPD7S	--	--	--	--	--
$12	P_SYS_SLEEP	00h	W	C_SYS_SLEEP= AAH (Write other data system to reset.)							
$13	P_INT_CTRL	00h	R/W	--	TMAOIE	--	TMBOIE	F1KIE	F4KIE	F32KIE	F2MIE
$14	P_INT_FLAG	00h	R/W	--	TMAOIF	--	TMBOIF	F1KIF	F4KIF	F32KIF	F2MIF
$15	P_INT_CTRLC	00h	R/W	R/0	R/0	TBHIE	TBLIE	--	R/0	R/0	R/0
$16	P_INT_FLAGC	00h	R/W	R/0	ENV_STU (R)	TBHIF	TBLIF	--	R/0	R/0	R/0
$17	P_TIMER_S	00h	R/W	R/0	IRENB	--	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
$1E	P_RFC_CTRL	00h	R/W	R/0	R/0	R/0	R/0	RFCEN	RFC2	RFC1	RFC0

$0020~002F: 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 system to reset.)							
$21	P_TMA_CTRL	00h	R/W	TMAES	--	TMACLK1	TMACLK0	R/0	R/0	TMAMOD1	TMAMOD0
$22	P_TMB_CTRL	00h	R/W	TMBES	TMBCLKIO	TMBCLK1	TMBCLK0	R/0	R/0	--	TMBMOD2
$23	P_TMA_CNTL	00h	R	8bits timer PWM Mode: TMA Counter 8 Bits							
	P_TMA_PWMF		W	8bits timer PWM Mode: TMA PWM 8 Bits Register for frequency set							
$24	P_TMA_CNTH	00h	R	8bits timer PWM Mode: TMA Counter 8 Bits							
	P_TMA_PWMD		R	8bits timer PWM Mode: TMA PWM 8 Bits Register for duty set							
$25	P_TMB_CNTL	00h	R	TMB Counter Low 8 Bits							
	P_TMB_REGL		W	TMB Low 8 Bits Register							
$26	P_TMB_CNTH	00h	R	ENVHL	R/0	R/0	R/0	TMB Counter High 4 Bits			
	P_TMB_REGH		W		-	-	-	TMB High 4 Bits Register			
$2A	P_C32K_EN	00h	R/W	C32KEN	R/0	R/0	R/0	R/0	R/0	R/0	R/0
$2B	P_TBS_SEL (32K)	00h	R/W	R/0	R/0	--	STRONG	TBLSEL1	TBLSEL0	TBHSEL1	TBHSEL0
$2C	P_Misc_reg	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	R/0	StrobeEn
$2D	P_InStrobEn_B	00h	R/W	R/0	R/0	R/0	R/0	R/0	InStrobEn_B2	InStrobEn_B1	InStrobEn_B0
$2E	P_InStrobEn_C	00h	R/W	InStrobEn_C7	InStrobEn_C6	InStrobEn_C5	InStrobEn_C4	InStrobEn_C3	InStrobEn_C2	InStrobEn_C1	InStrobEn_C0
$2F	P_InStrobEn_D	00h	R/W	InStrobEn_D7	R/0	InStrobEn_D5	R/0	InStrobEn_D3	InStrobEn_D2	InStrobEn_D1	InStrobEn_D0

$0030~0033: Wake up & LCD

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$30	P_WAKEUP_CTRL	00h	R/W	R/0	R/0	R/0	R/0	TBHWEN	R/0	TBLWEN	KEYWEN
$31	P_WAKEUP_STA	00h	R/W	R/0	R/0	R/0	R/0	TBHWFC	R/0	TBLWFC	KEYWFC
$32	P_LCD_CTRL	00h	R/W	R/0	R/0	R/0	Green Mode	LBSDT1	LBSDT0	LCDMD1	LCDMD0
$33	P_LVD_CTRL	00h	R/W	LVDEN	R/0	LVDS	R/0	R/0	R/0	R/0	LVD(R)
$37	PB_NMWKEN	00h	R/W	R/0	R/0	R/0	R/0	R/0	PBWK2	PBWK1	PBWK0
$38	PC_NMWKEN	00h	R/W	PCWK7	PCWK6	PCWK5	PCWK4	PCWK3	PCWK2	PCWK1	PCWK0
$39	PD_NMWKEN	00h	R/W	PDWK7	R/0	PDWK5	R/0	PDWK3	PDWK2	PDWK1	PDWK0
$3F	P_SRAMP_CTRL	00h	R/W	-	-	BANK5	BANK4	-	-	BANK1	BANK0

$004A~004C: IO & Segment Selection

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$4A	P_IOA_SEL	00h	R/W	PAS7	PAS6	PAS5	PAS4	PAS3	PAS2	PAS1	PAS0
$4C	P_IOC_SEL	00h	R/W	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0

$0050~005F: DPRAM ADDRESS

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$50	COM0	00h	R/W	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
$51				SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
$52				SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
$53				SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
$54	COM1	00h	R/W	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
$55				SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
$56				SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
$57				SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
$58	COM2	00h	R/W	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
$59				SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
$5A				SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
$5B				SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
$5C	COM3	00h	R/W	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
$5D				SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
$5E				SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
$5F				SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24

Note: -- means no function.

5.3. Clock Source

GPM6P1816A supports Crystal / Ceramic or Internal oscillator, as shown in the following diagram, Figure 5-8 .   It can be selected by device configuration register at address ($FFF8.0) and can be set in G+ IDE.

The detailed configuration register setting of device has been given in *Section 5.2.3 Configuration Registe.*

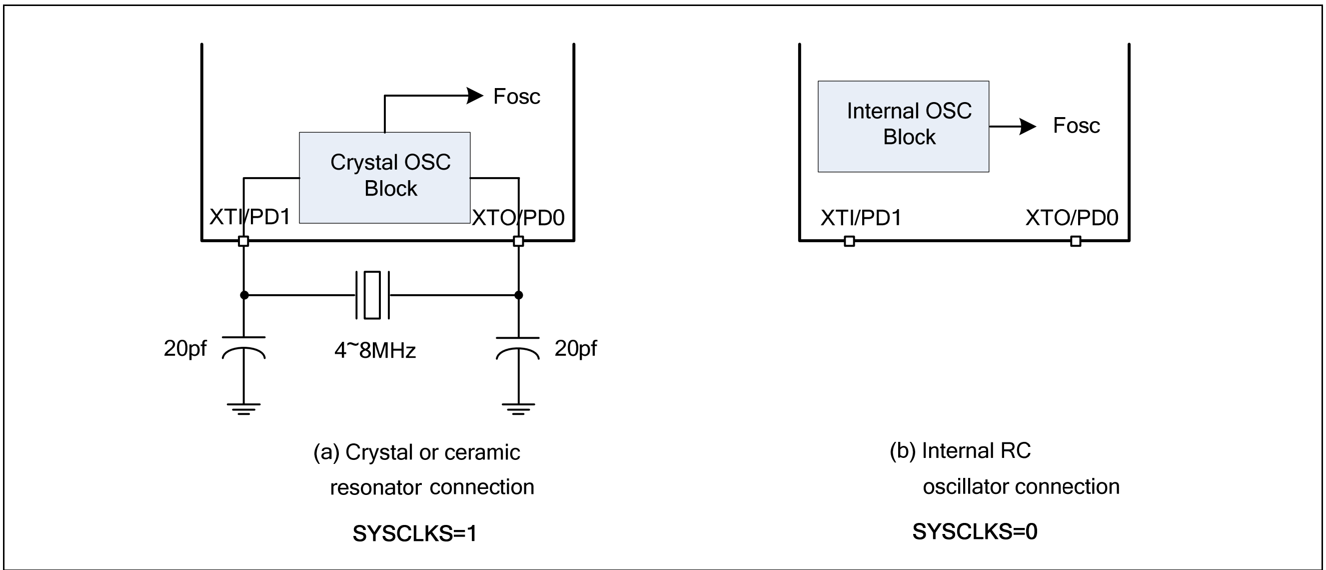


Figure 5-8 Two types of clock sources

5.4. Power Saving Mode

5.4.1. Introduction

To reduce the current consumption when the system does not need to be active, SLEEP mode, HALT mode and GREEN mode can be utilized. Those modes are able to reduce power consumption and save power. User must write corresponding value to SLEEP Control Register to enter SLEEP mode. And the HALT mode and GREEN mode are just like SLEEP mode except turning on LCD driver. The difference between HALT mode and GREEN mode is LCD working current. The HALT mode needs more working current than GREEN mode does but with better LCD display quality. For more information about SLEEP modes, please see Figure 5-9 and they will be depicted in the next two sections.

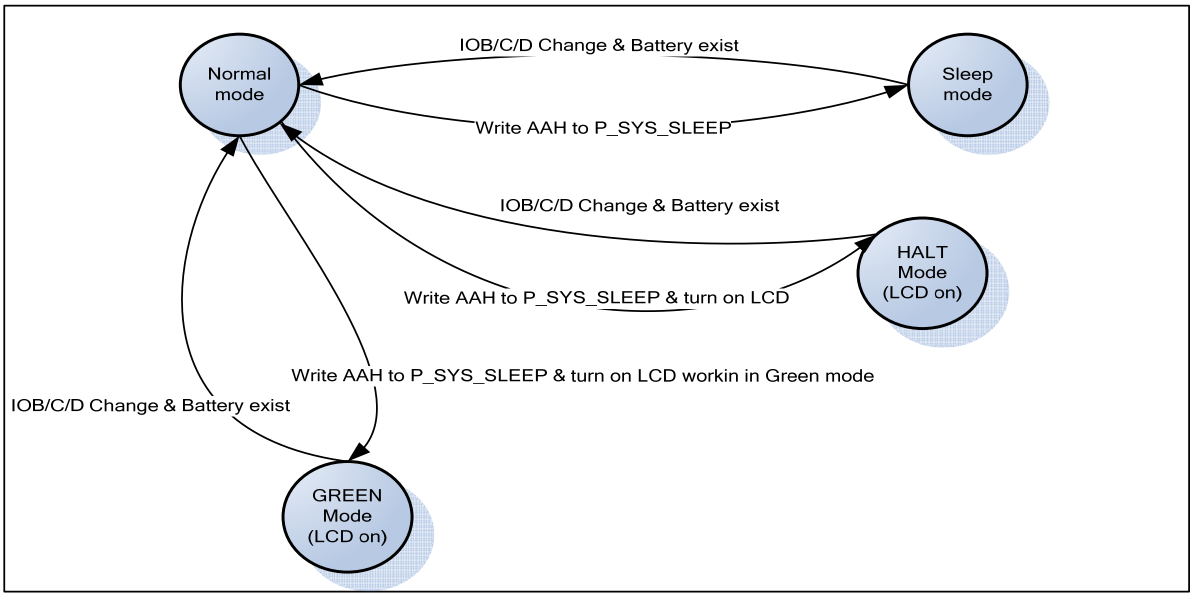


Figure 5-9 Power saving mode operation

5.4.2. SLEEP / HALT / GREEN Mode

SLEEP mode function will disable all system clocks, including the clock generation circuit. Once the system enters SLEEP mode, LVR function is disabled, RAM and I/Os will remain at their previous states until awake. The system will be awakened by any change occurred on PortB/C/D (M-Type Key). After GPM6P1816A is awakened, the internal CPU will remain on previous state until Tw ≧ 16384 x T1 (Tw = waiting time & T1 = system clock cycle); and then continue processing the program. (See Figure 5-10).

$$T1 = 1 / (F_{CPU}),\ Tw \geqq 16384 \times T1$$

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

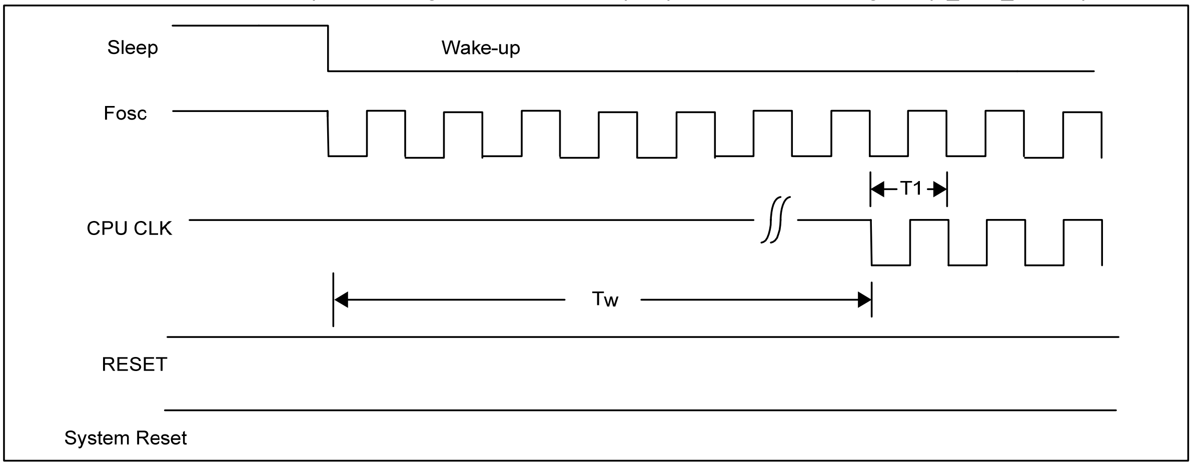


Figure 5-10 SLEEP mode

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

Bit [7:0] **SLEEPCTRL** [7:0]: Operation mode control.
$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 PortB
        LDA     #FFH
        STA     P_PB_NMWKEN             ; set port B as wake up source
        LDA     # C_KEYWEN
        STA     P_WAKEUP_CTRL           ; set Key wakeup
        LDA     #C_SYS_SLEEP            ; SLEEP command $AA
        STA     P_SYS_ SLEEP            ; go to sleep mode
```

Sleep mode will disable all clock sources for power savings. The way entering Halt mode is just like the way entering sleep mode. However, halt mode has to enable 32KHz clock source (default is enable) and turn on LCD driver. So, the only difference between sleep mode and halt mode is LCD on or off. The way entering green is the same as entering halt mode. If $32 bit[5]=0, then write AAH to $12 with LCD on means entering halt mode. If $32 bit[5]=1, write AAH to $12 with LCD on means entering green mode. In green mode, LCD operates under less working current but the display quality is not as good as halt mode is. The selection between halt mode and green mode is the trade off between current consumption and LCD display quality.

32K clock Control Register (P_C32K_EN, $002A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	C32KEN	R/0	R/0	R/0	R/0	R/0	R/0	R/0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit [7] **C32KEN:** 32k clock source enable, default is enable. if Write 0 , disable this 32K counter.
if Read, reset 32K counter.

HALT / GREEN mode Control Register (P_LCD_CTRL, $0032)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	R/0	R/0	R/0	Green Mode	LBSDT1	LBSDT0	LCDMD1	LCDMD0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit [1:0] **LCDMD [1:0]**: LCD mode control. Default is '00'.
11 = display off, 10 = all off,
01 = all on , 00 = normal;

Bit [4] **Green mode:** LCD green mode enable.
0: disable ,1:enable;

5.4.3. Wake up Event

There are two way to wake up system from sleep / halt mode.
Either one set can be entering sleep / halt mode.

1. Key wakeup by setting specific IO as wakeup source.
   IO wakeup source: PORT B/C/D. Refer to relative IO setting.
2. Time base (32768) wake up event.
   There are two frequencies to select: High frequency and low frequency. For other relevant frequency, please refer to Time Base Frequency selection Register ($002B).

Wakeup Control Register (P_WAKEUP_CTRL, $0030)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	R/0	TBHWEN	R/0	TBLWEN	KEYWEN
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 [7:4],[2] Reserved.

Bit [3] **TBHWEN**: Time base high frequency wakeup enable
0 = wake up disable
1 = wake up enable

Bit [1] **TBLWEN:** Time base low frequency wakeup enable
0 = wakeup disable
1 = wakeup enable

Bit [0] **KEYWEN**: Key wakeup enable
0 = wakeup disable
1 = wakeup enable

Wakeup Control Flag (P_WAKEUP_STA, $0031)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	R/0	TBHWFC	R/0	TBLWFC	KEYWFC
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 [7:4],[2] Reserved.

Bit [3] **TBHWEN**: Time base high frequency wakeup flag
0 = no event
1 = event has occurred
Disable TBHWEN to clear flag.

Bit [0] **KEYWEN**: Key wakeup flag
0 = no event
1 = event has occurred
Disable KEYWEN to clear flag.

Bit [1] **TBLWEN:** Time base low frequency wakeup flag
0 = no event
1 = event has occurred
Disable TBLWEN to clear flag.

Time Base Frequency selection Register (P_TBS_SEL, $002B)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	--	STRONG	TBLSEL1	TBLSEL0	TBHSEL1	TBHSEL0
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 [7:6] Reserved.

Bit [5] No function.

Bit[4] **STRONG:** Crystal-OSC strong mode duration selection
0 = 2 sec(Default)
1 = 4 sec

Bit [3:2] **TBLSEL[1:0]:** Time Base Low Frequency select
00 = 2Hz 10:4Hz
01 = 8Hz 11:16Hz

Bit [1:0] **TBHSEL[1:0]:** Time Base High Frequency select
00 = 128Hz 10:256Hz
01 = 512Hz 11:1KHz

5.4.4. Strobe function

In order to combine LCD function and Key wake at the same IO PAD, Considering segment output level will seriously infect the key wake up function. The strobe function can solved the problem. As the figure below, when strobe function is enabled (refer to register Misc register $002C), all the com & segment pins & wake up source need to enable the strobe function (refer to register $002D / $002E / $002F).System will detect key press or not at strobe point.

In this way can avoid key pressing action to inflect the LCD display.

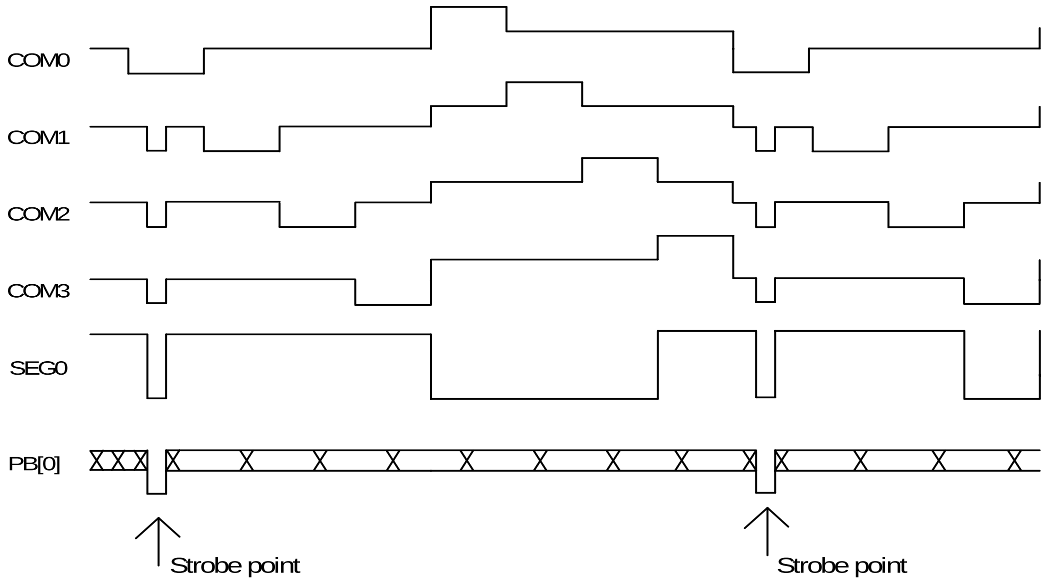


Figure 5-11 Leave interrupt routine

Misc Register (P_Misc_reg, $002C)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	R/0	--	--	--	StrobEn
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 [7:4] Reserved.

Bit [3:1] No function.

Bit [0] **StrobEn:** Strobe Enable / Disable
0 = Disable.
1 = Enable.

Port B Strobe Control Register (P_InStrobEn_B, $002D)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	R/0	--	InStrobEn_B2	InStrobEn_B1	InStrobEn_B0
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 [7:4] Reserved.

Bit [3] No function.

Bit [2] **InStrobEn_B2:** Port B[2] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [1] **InStrobEn_B1:** Port B[1] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [0] **InStrobEn_B0:** Port B[0] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Port C Strobe Control Register (P_InStrobEn_C, $002E)

BIT	7	6	5	4	3	2	1	0
Name	InStrobEn_C7	InStrobEn_C6	InStrobEn_C5	InStrobEn_C4	InStrobEn_C3	InStrobEn_C2	InStrobEn_C1	InStrobEn_C0
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] **InStrobEn_C7:** Port C[7] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [6] **InStrobEn_C6:** Port C[6] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [5] **InStrobEn_C5:** Port C[5] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [4] **InStrobEn_C4:** Port C[40] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [3] **InStrobEn_C3:** Port C[3] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [2] **InStrobEn_C2:** Port C[2] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [1] **InStrobEn_C1:** Port C[1] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [0] **InStrobEn_C0:** Port C[0] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Port D Strobe Control Register (P_InStrobEn_D, $002F)

BIT	7	6	5	4	3	2	1	0
Name	InStrobEn_D7	R/0	InStrobEn_D5	R/0	InStrobEn_D3	InStrobEn_D2	InStrobEn_D1	InStrobEn_D0
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 [6],[4] Reserved.

Bit [7] **InStrobEn_D7:** Port D[7] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [5] **InStrobEn_D5:** Port D[5] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [3] **InStrobEn_D3:** Port D[3] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [2] **InStrobEn_D2:** Port D[2] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [1] **InStrobEn_D1:** Port D[1] Strobe Enable / Disable
0 = Disable.
1 = Enable.

Bit [0] **InStrobEn_D0:** Port D[0] Strobe Enable / Disable
0 = Disable.
1 = Enable.

5.5. Interrupt

5.5.1. Introduction

GPM6P1816A provides 10 types of interrupt sources with the same normal interrupt level. The 10 types of interrupt sources are Timer A overflow interrupt, Timer B overflow interrupt, time Fosc/1024 interrupt, time Fosc/4096 interrupt, time Fosc/32768 interrupt, time Fosc/2097152 interrupt, port B INT1 interrupt, port B INT0 interrupt, time-base high frequency interrupt, time-base low frequency interrupt.

These interrupts have individual status (occurred or not occurred) and control (enabled or not enabled) 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 then CPU will execute the interrupt 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. 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 Figure5-11. And abstract the interrupt service routine first address from $FFFE and $FFFF. In a corresponding way, the system abstract the return PC address from the bottom of the stack when finished the interrupt service (See Figure 5-12).

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/2097152	F2MIF($0014.0)	F2MIE($0013.0)
Timer B Overflow	TMBOIF($0014.4)	TMBOIE($0013.4)	Timer base H freq.	TBHIE($0016.5)	TBHIF($0015.5)
Time Fosc/1024	F1KIF($0014.3)	F1KIE($0013.3)	Timer base L freq.	TBLIE($0016.4)	TBLIF($0015.4)
Time Fosc/4096	F4KIF($0014.2)	F4KIE($0013.2)	PBINT1	PBINT1($0019.1)	PBINTF1($0018.1)
Time Fosc/32768	F32KIF($0014.1)	F32KIE($0013.1)	PBINT0	PBINT0($0019.0)	PBINTF0($0018.0)

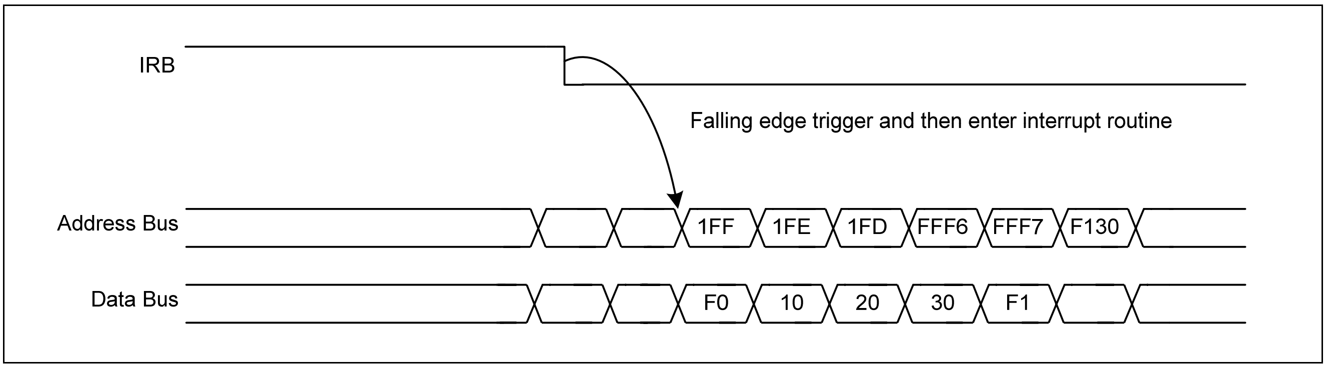


Figure 5-12 Interrupt triggered by IRB

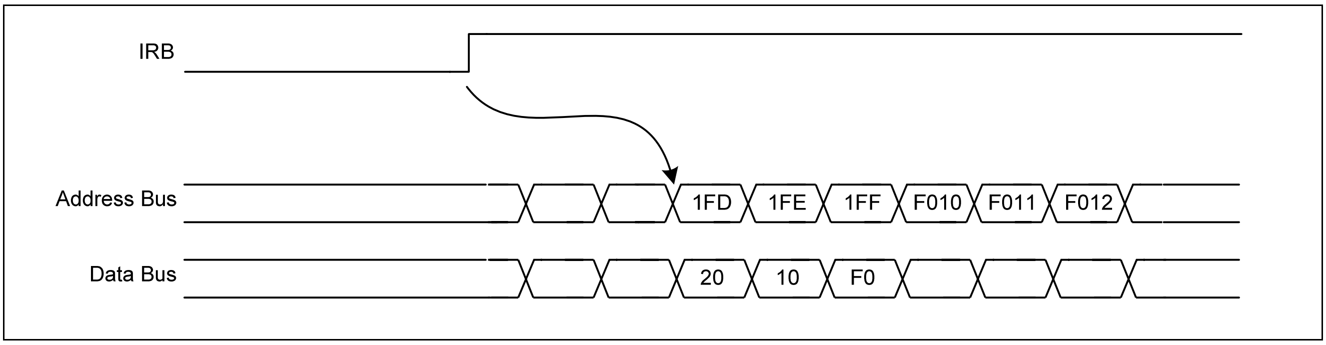


Figure 5-13 Leave interrupt routine

5.5.2. Interrupt Register

Interrupt Flag Register (P_INT_FLAG, $0014)

BIT	7	6	5	4	3	2	1	0
Name	--	TMAOIF	--	TMBOIF	FD1KIF	FD4KIF	FD32KIF	FD2MIF
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 [7] no function.

Bit [6] **TMAOIF**: Timer A overflow interrupt flag
0 = no event
1 = event has occurred

Bit [5] no function.

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

Bit [3] **FD1KIF**: Time Fosc/1024 interrupt flag
0 = no event
1 = event has occurred

Bit [2] **FD4KIF**: Time Fosc/4096 interrupt flag
0 = no event
1 = event has occurred

Bit [1] **FD32KIF**: Time Fosc/32768 interrupt flag
0 = no event
1 = event has occurred

Bit [0] **FD2MIF**: Time Fosc/2097152 interrupt flag
0 = no event
1 = event has occurred

Interrupt Control Register (P_INT_CTRL, $0013)

BIT	7	6	5	4	3	2	1	0
Name	--	TMAOIE	--	TMBOIE	FD1KIE	FD4KIE	FD32KIE	FD2MIE
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] no function.

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

Bit [5] no function.

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

Bit [3] **FD1KIE**: Time Fosc/1024 interrupt enable bit
0 = interrupt disable
1 = interrupt enable

Bit [2] **FD4KIE**: Time Fosc/4096 interrupt enable bit
0 = interrupt disable
1 = interrupt enable

Bit [1] **FD32KIE**: Time Fosc/32768 interrupt enable bit
0 = interrupt disable
1 = interrupt enable

Bit [0] **FD2MIE**: Time Fosc/2097152 interrupt enable bit
0 = interrupt disable
1 = interrupt enable

Time Base Interrupt Flag Register (P_INT_FLAGC, $0016)

BIT	7	6	5	4	3	2	1	0
Name	--	ENV_STUS(R)	TBHIF	TBLIF	--	R/0	R/0	R/0
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 [5] **TBHIF:** Time base High frequency interrupt flag.
0 = no event
1 = event has occurred

Bit [4] **TBLIF:** Time base Low frequency interrupt flag.
0 = no event
1 = event has occurred

Time Base Interrupt Control Register (P_INT_CTRLC, $0015)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	TBHIE	TBLIE	--	R/0	R/0	R/0
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 [5] **TBHIE:** Time base High frequency interrupt.
0 = interrupt disable
1 = interrupt enable

Bit [4] **TBLIE:** Time base Low frequency interrupt.
0 = interrupt disable
1 = interrupt enable

Time Base Frequency selection Register (P_TBS_SEL, $002B)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	--	STRONG	TBLSEL1	TBLSEL0	TBHSEL1	TBHSEL0
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 [7:6] Reserved.

Bit [5] No function.

Bit [4] **STRONG :** Crystal-OSC strong mode duration selection.
0 = 2 sec(Default)
1 = 4 sec

Bit [3:2] **TBLSEL[1:0]:** Time Base Low Frequency select
00 = 2Hz 10:4Hz
01 = 8Hz 11:16Hz

Bit [1:0] **TBHSEL[1:0]:** Time Base High Frequency select
00 = 128Hz 10:256Hz
01 = 512Hz 11:1KHz

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

Example] 5-4 Enable Timer A overflow interrupt

```
;=========================;
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
```

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 internal events come from the program run away. Figure 5-14 shows the affected region for each reset source.

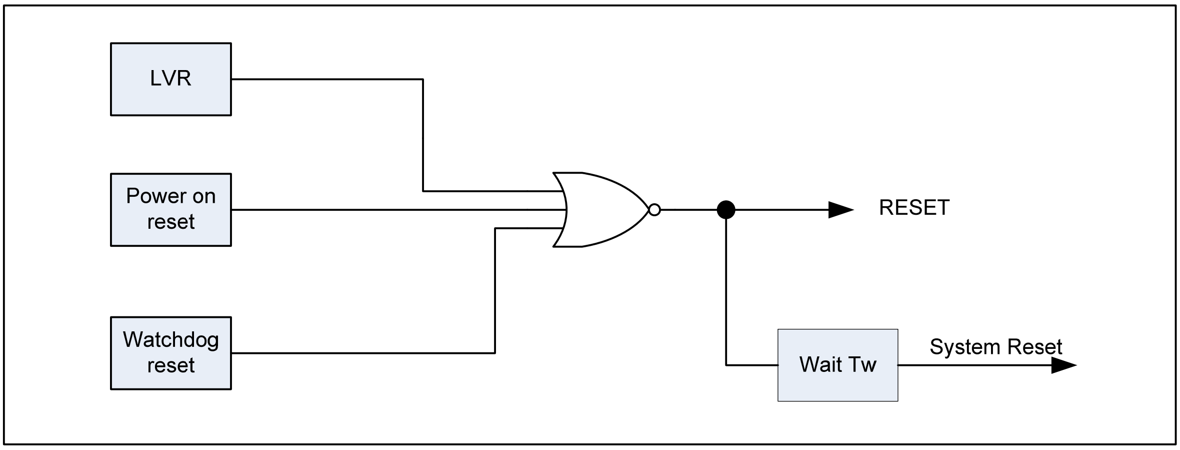


Figure 5-14 Reset sources

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 state 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 register bit $FFF8.1(can be set in G+ 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 the power is lower than the specific level for a specific period, the system will enter reset state and all I/Os will be locked.

5.6.4. Watchdog Timer Reset (WDR)

On-chip watchdog circuitry makes the device entering reset when MCU enters an unknown state without watchdog clearing information. This function prevents the MCU from being stuck in an abnormal condition. Watchdog Timer (WDT) can be disabled or enabled through configuring register bit $FFF8.2 (can be set in G+ IDE as Figure 5-7). Watchdog Timer Reset will be generated by a time-out event of the WDT automatically when watchdog is enabled.

Watchdog Timer Reset will reset the CPU and restart the program. To avoid a WDT time-out reset, user should 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.

Different Reset Sequences as the following figures:

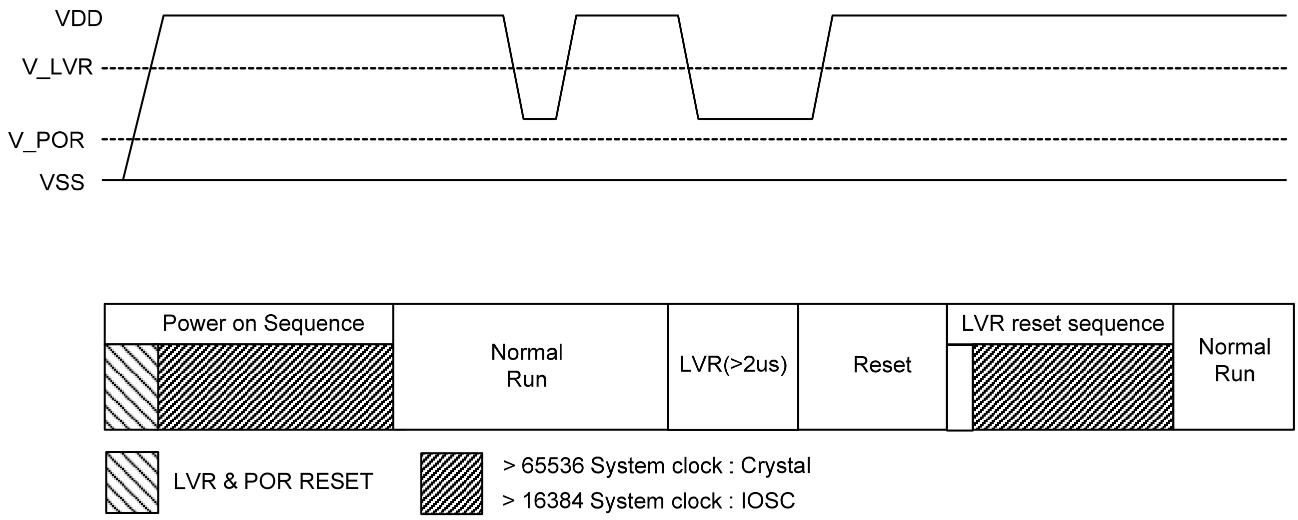


Figure 5-15 Reset Sequence

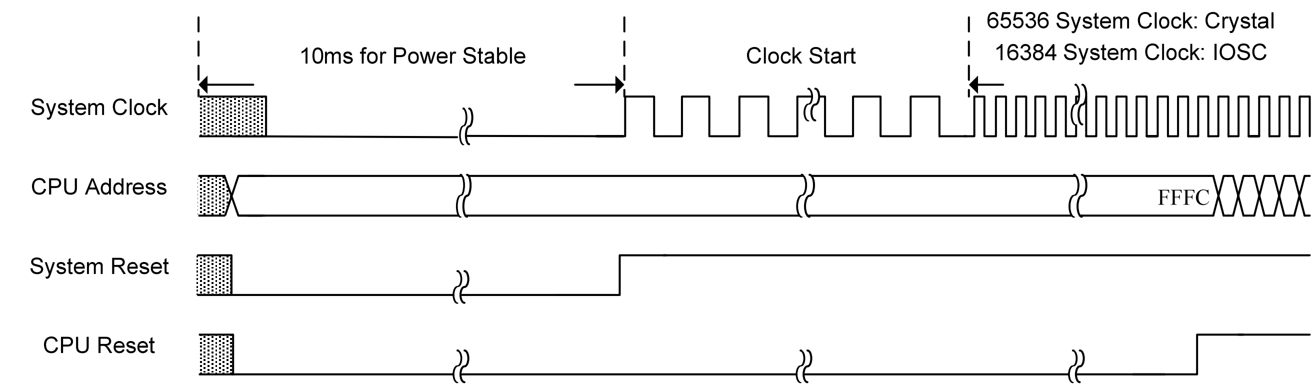


Figure 5-16 Power-On Reset Sequence

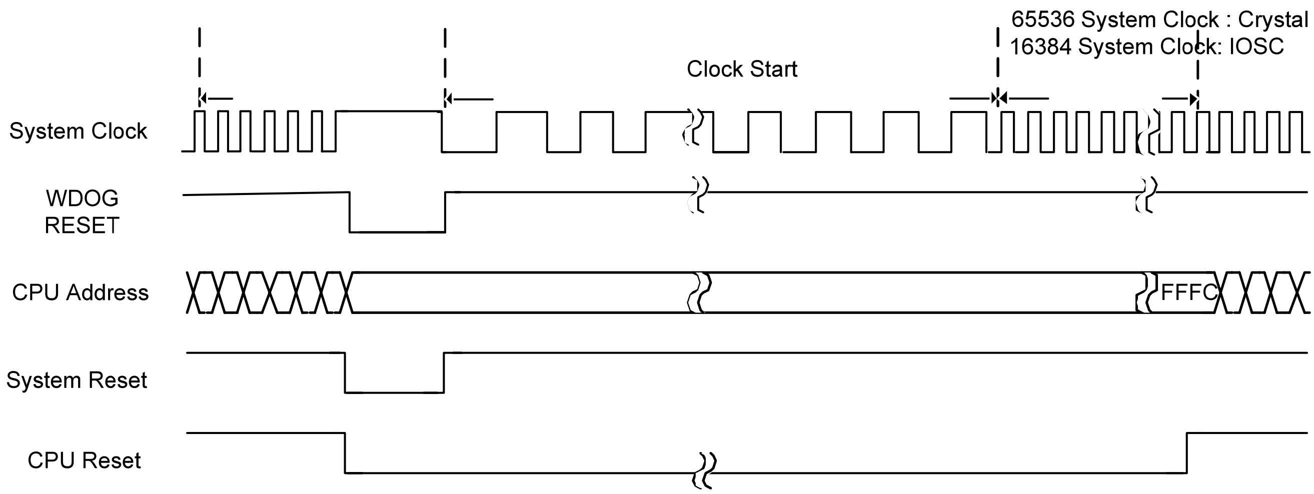


Figure 5-17 Watchdog Reset Sequence

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

Bit [7:0] **WDTCTRL [7:0]:** Operation mode control register Other data = reset system
$AA = write to clear watchdog CNT (C_WDT_CLR)

[Example] 5-5 Clear watchdog counter

```
LDA     # C_WDT_CLR                 ; Clear watchdog command $AA
STA     P_WDT_ CTRL
```

5.7. I/O PORTS

5.7.1. Introduction

GPM6P1816A has 4 ports, Port A, Port B, Port C and Port D. These port pins may be multiplexed with alternate functions for the peripheral features on the device. In general, when an initial reset state occurs, all ports are used as a general purpose input port, except PD7. PD7 default is IR TX pin. There are three parts, data, direction and attribution registers, in these IO structures. Each corresponding bit in these ports should be given a value.

In M-Type keyboard application, Port B/C/D can be configured as input ports, and in sleep mode any change occurred on these ports will trigger system wakeup.

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 can be different when programmer sets the port to different configuration (input pull-high/pull- low).

Please refer to the
[Table] 5-2 for PD[5] and **[Table] 5-3** for PA[7:0], PB[2:0], PC[7:0], PD[7], PD[3:0]'s setting.

[Table] 5-2 I/O Configurations (for PD[5])

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	Floating	Float
1	0	0	Floating	Input with float
1	0	1	Pull high	Input with pull-high
1	1	0	Floating	Float
1	1	1	Driving low	Output Data

[Table] 5-3 I/O Configurations (for PA[7:0], PB[2:0], PC[7:0], PD[7], PD[3:0])

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

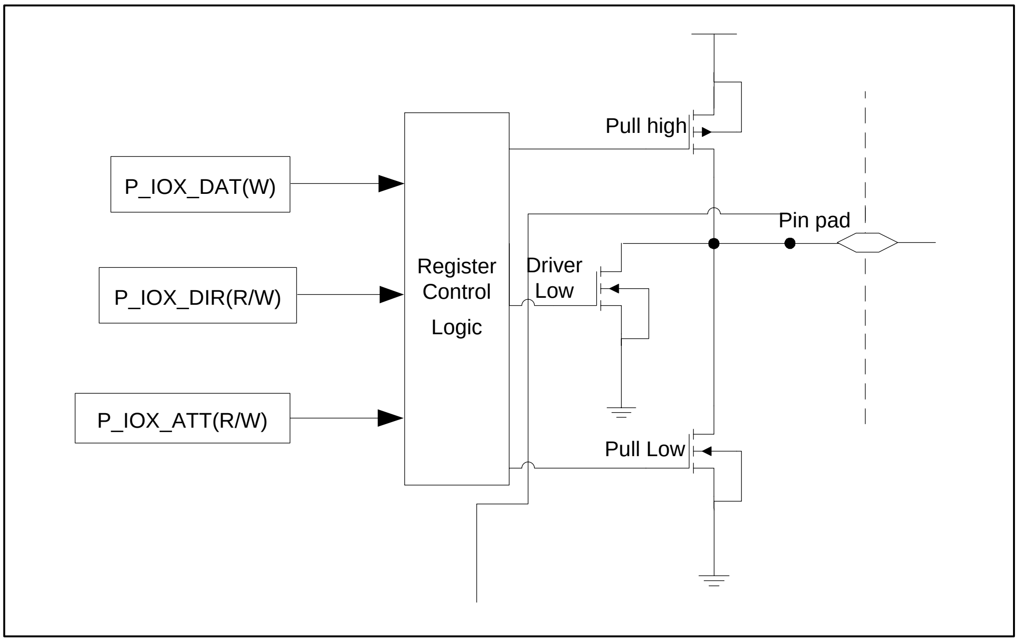


Figure 5-18 Block diagram of I/O port (PD[5])

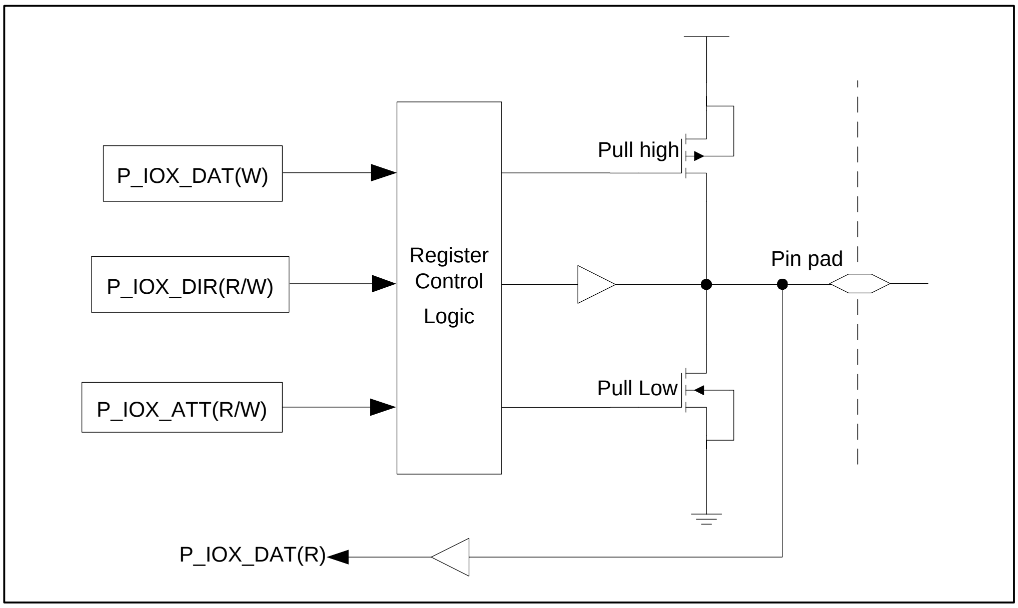


Figure 5-19 Block diagram of I/O port (PA[7:0], PB[2:0], PC[7:0], PD[7], PD[3:0])

5.7.2. Port A

Port A is an 8-bit programmable bi-directional port. The port is controlled by direction control register P_IOA_DIR, and attribution register P_IOA_ATT, and data register P_IOA_DAT. Reading P_IOA_DAT will get the real IO value. Read P_IOA_DATBF to get data register value.

[Table] 5-4 Port A Function List

Port A Pin	BIT	Shared function
PA[7:0]	7:0	LCD SEGMENT output [23:16]. Default is GPIO.

Port A Direction Register (P_IOA_DIR, $000B)

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

Bit [7:0] **P_IOA_DIR:** Port A direction register.
0 = input
1 = output

Port A Attribution Register (P_IOA_ATT, $0003)

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

Bit [7:0] **P_IOA_ATT:** Port A attribution register

Port A Data Register (P_IOA_ DAT, $0007)

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

Bit [7:0] **P_IOA_ DAT:** Port A Data value.
Read to get Port A pin PAD value
Write to configure output high/low or configure input with pull high/low resistor.

Port A Data Buffer Register (P_IOA_ DATBF, $000C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	P_IOA_ DATBF							
ACCESS	R							
DEFAULT	00h							

Bit [7:0] **P_IOA_ DATBF:** Port A Data Buffer value.
Read to get Port A data register value

[Example] 5-6 Set Port A [7:4] as output with low data and Port A [3:0] as input with pulling high.

```
LDA    #$F0          ; store accumulator with $F0
STA    P_IOA_DIR     ; set direction
LDA    #$0F          ; store accumulator with $0F
STA    P_IOA_ATT     ; set attribute
LDA    #$0F          ; store accumulator with $0F
STA    P_IOA_DAT     ; set Port Data
```

[Example] 5-7 Set Port A [7:0] as input with float.

```
LDA    #$00          ; store accumulator with $00
STA    P_IOA_ATT     ; set direction
STA    P_IOA_DIR     ; set attribute
STA    P_IOA_DAT     ; set Port Data
```

Port A Segment Function Control Register (P_IOA_SEL, $004A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	PAS7	PAS6	PAS5	PAS4	PAS3	PAS2	PAS1	PAS0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:0] **PAS0[7:0]:** Port A Segment function control.
0: set port A as normal I/O.
1: set port A as segment output.

[Example] 5-8 Set Port A [7:0] as segment output.

```
LDA    #$FF          ; store accumulator with $FF
STA    P_IOA_SEL     ; set segment function
```

5.7.3. Port B

Port B is a 3-bit programmable bi-direction port. The port is controlled by direction control register P_IOB_DIR, and attribution register P_IOB_ATT, and data register P_IOB_DAT.

Reading P_IOB_DAT will get the real IO value.
Read P_IOB_DATBF to get data register value.

[Table] 5-5 Port B Function List

Port A Pin	BIT	Shared function
PB[1:0]	1:0	External interrupt. Default is GPIO.

Port B Direction Register (P_IOB_DIR, $0000)

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

Bit [7:3] Reserved
Bit [2:0] **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	R/0	R/0	R/0	R/0	R/0	P_IOB_ATT		
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W		
DEFAULT	00h							

Bit [7:3] Reserved

Bit [2:0] **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	R/0	R/0	R/0	R/0	R/0	P_IOB_ DAT		
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W		
DEFAULT	00h							

Bit [7:3] Reserved

Bit [2:0] **P_IOB_ DAT:** Port B Data value.

Read to get Port B pin PAD value

Write to configure output high/low or configure input with pull high/low resistor.

Port B Data Buffer Register (P_IOB_ DATBF, $000D)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	R/0	R/0	R/0	R/0	R/0	P_IOB_ DATBF		
ACCESS	R/W	R/W	R/W	R/W	R/W	R		
DEFAULT	00h							

Bit [7:3] Reserved

Bit [2:0] **P_IOB_ DATBF:** Port B Data Buffer value.

Read to get Port B data register value

[Example] 5-9 Set Port B [2] as output with low data and Port B [1:0] as input with pulling high.

```
LDA   #$04          ; store accumulator with $04
STA   P_IOB_DIR     ; set direction
LDA   #$03          ; store accumulator with $03
STA   P_IOB_ATT     ; set attribute
LDA   #$03          ; store accumulator with $03
STA   P_IOB_DAT     ; set Port Data
```

[Example] 5-10 Set Port B [2:0] as input with float.

```
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
```

Port B Strobe Function Control Register (P_InStrobEn_B, $002D)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	R/0	R/0	R/0	R/0	R/0	InStrobEn_B2	InStrobEn_B1	InStrobEn_B0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:3] Reserved

Bit [2:0] **InStrobEn_B[2:0]:** Port B strobe function enable.
0: strobe disable.
1: strobe enable.

Port B can be configured as key wake up source or not by wake up source control register.

Port B Wake Up Source Control Register (PB_NMWKEN, $0037)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	R/0	R/0	R/0	R/0	R/0	PBWK2	PBWK1	PBWK0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:3] Reserved
Bit [2:0] **PBWK [2:0]:** Port B wake up source enable.
0: wakeup source disable.
1: wakeup source enable.

[Example] 5-11 Set PB[2:0] as key wakeup source.

```
LDA     #$07                ; store accumulator with $07
STA     PB_NMWKEN           ; set PB[2:0] as key wake up source
```

5.7.4. Port C

Port C is a 8-bit programmable bi-directional port. The port is controlled by direction control register P_IOC_DIR, and attribution register P_IOC_ATT, and data register P_IOC_DAT.

Reading P_IOC_DAT will get the real IO value. Read P_IOC_DATBF to get data register value.

[Table] 5-6 Port C Function List

Port A Pin	BIT	Shared function
PC[7:0]	BIT[7:0]	LCD SEGMENT output [31:24]. Default is GPIO.
PC[6:4]	BIT[6:4]	RFC function output. Default is GPIO.
PC[7]	BIT[7]	RFC function input. Default is GPIO.

Port C Direction Register (P_IOC_DIR, $0001)

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

Bit [7:0] **P_IOC_DIR:** Port C direction register.
0 = input
1 = output

Port C Attribution Register (P_IOC_ATT, $0005)

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

Bit [7:0] **P_IOC_ATT:** Port C attribution register

Port C Data Register (P_IOC_ DAT, $0009)

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

Bit [7:0] **P_IOC_ DAT:** Port C Data value.
Read to get Port C pin PAD value
Write to configure output high/low or configure input with pull high/low resistor.

Port C Data Buffer Register (P_IOC_ DATBF, $000E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	P_IOC_ DATBF							
ACCESS	R							
DEFAULT	00h							

Bit [7:0] **P_IOC_ DATBF:** Port C Data Buffer value.
Read to get Port C data register value

[Example] 5-12 Set Port C [7:4] as output with low data and Port C [3:0] as input with pulling high.

```
LDA    #$F0          ; store accumulator with $F0
STA    P_IOC_DIR     ; set direction
LDA    #$0F          ; store accumulator with $0F
STA    P_IOC_ATT     ; set attribute
LDA    #$0F          ; store accumulator with $0F
STA    P_IOC_DAT     ; set Port Data
```

[Example] 5-13 Set Port C [7:0] as input with float.

```
LDA    #$00          ; store accumulator with $00
STA    P_IOC_ATT     ; set direction
STA    P_IOC_DIR     ; set attribute
STA    P_IOC_DAT     ; set Port Data
```

Port C Strobe Function Control Register (P_InStrobEn_C, $002E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	InStrobEn_C7	InStrobEn_C6	InStrobEn_C5	InStrobEn_C4	InStrobEn_C3	InStrobEn_C2	InStrobEn_C1	InStrobEn_C0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:0] **InStrobEn_C[7:0]:** Port C strobe function enable.
0: strobe disable.
1: strobe enable.

Port C can be configured as key wakeup source or not by wakeup source control register.

Port C Wake Up Source Control Register (PC_NMWKEN, $0038)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	PCWK7	PCWK6	PCWK5	PCWK4	PCWK3	PCWK2	PCWK1	PCWK0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:0] **PCWK [7:0]:** Port C wake up source enable.
0: wakeup source disable.
1: wakeup source enable.

[Example] 5-14 Set PC[3:0] as key wakeup source.

```
LDA    #$0F            ; store accumulator with $0F
STA    PC_NMWKEN       ; set PC[3:0] as key wake up source
```

Port C Segment Function Control Register (P_IOC_SEL, $004C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:0] PCS0[7:0]: Port C Segment function control.

0: set port C as normal I/O.

1: set port C as segment output.

[Example] 5-15 Set Port C [7:0] as segment output.

```
LDA    #$FF            ; store accumulator with $FF
STA    P_IOC_SEL       ; set segment function
```

Port C RFC Function Control Register (P_RFC_CTRL, $001E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	R/0	R/0	R/0	R/0	RFCEN	RFC2	RFC1	RFC0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [3] RFCEN: RFC function control.

0:Disable RFC function.

1:Enable RFC function.

Note: when enable RFC function, PC7 is configured as input and PC[6:4] are configured as output.

Bit [2:0] RFC[2:0] : RFC port control register.

0: RFC port disable.

1: RFC port enable.

[Example] 5-16 Set Port C [7:4] as RFC function

```
LDA    #$08            ; store accumulator with $08
STA    P_RFC_CTRL,     ; set RFC function
LDA    #$09            ; store accumulator with $09
STA    P_RFC_CTRL,     ; set RFC0 function
LDA    #$0A            ; store accumulator with $0A
STA    P_RFC_CTRL,     ; set RFC1 function
LDA    #$0C            ; store accumulator with $0C
STA    P_RFC_CTRL,     ; set RFC2 function
```

5.7.5. Port D

Port D is a 6-bit programmable bi-directional port. The port is controlled by direction control register P_IOD_DIR, and attribution register P_IOD_ATT, and data register P_IOD_DAT.

Reading P_IOD_DAT will get the real IO value.

Read P_IOD_DATBF to get data register value.

And PD5 can set as input pull low/high or driver low but without driver high function.

[Table] 5-7 Port D Function List

Port A Pin	BIT	Shared function
PD0	Bit0	4M/8M Crystal output (XTO). Default is GPIO.
PD1	Bit1	4M/8M Crystal input (XTI). Default is GPIO.
PD2	Bit2	32K Crystal output (X32TI). Default is GPIO.
PD3	Bit3	32K Crystal input (X32TO). Default is GPIO.
PD5	Bit5	VPP function.
PD7	Bit7	REM function. Default is REM.

Port D Direction Register (P_IOD_DIR, $0002)

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

Bit [6][4] Reserved

Bit [7],[5],[3:0] **P_IOD_DIR**: Port D direction register.

0 = input

1 = output

Port D Attribution Register (P_IOD_ATT, $0005)

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

Bit [6][4] Reserved

Bit [7],[5],[3:0] **P_IOD_ATT**: Port D attribute register.

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

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	P_IOD_ DAT	R/0	P_IOD_ DAT	R/0	P_IOD_ DAT			
ACCESS	R/W	R/W	R/W	R/W	R/W			
DEFAULT	00h							

Bit [6][4] Reserved

Bit [7],[5],[3:0] **P_IOD_DAT**: Port D data register.

Read to get Port D value

Write to configure output high/low or configure input with pull high/low resistor.

PD5 has no output High function.

Port D Data Buffer Register (P_IOD_ DATBF, $000F)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	P_IOD_ DATBF	R/0	P_IOD_ DATBF	R/0	P_IOD_ DATBF			
ACCESS	R/W	R/W	R/W	R/W	R/W			
DEFAULT	00h							

Bit [6][4] Reserved

Bit [7],[5],[3:0] **P_IOD_DATBF**: Port D data register.

Read to get Port D data register value

[Example] 5-17 Set Port D[3:0] as output with low data.

```
LDA    #$0F            ; store accumulator with $0F
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
```

Port D Strobe Function Control Register (P_InStrobEn_D, $002F)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	InStrobEn_D7	R/0	InStrobEn_D5	R/0	InStrobEn_D3	InStrobEn_D2	InStrobEn_D1	InStrobEn_D0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [6][4] Reserved

Bit [7],[5],[3:0] **InStrobEn_D[7],[5],[3:0]:** Port D strobe function enable.

0: strobe disable.

1: strobe enable.

Port D can be configured as key wake up source or not by wakeup source control register.

Port D Wake Up Source Control Register (PD_NMWKEN, $0039)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	PDWK7	R/0	PDWK5	R/0	PDWK3	PDWK2	PDWK1	PDWK0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [6][4] Reserved

Bit [7],[5],[3:0] **PDWK [7],[5],[3:0]:** Port D wake up source enable.

0: wakeup source disable.

1: wakeup source enable.

[Example] 5-18 Set PD[3:0] as key wakeup source.

```
LDA    #$0F            ; store accumulator with $0F
STA    PD_NMWKEN       ; set PD[3:0] as key wake up source
```

5.8. Timer Module

5.8.1. Introduction

GPM6P1816A has two timers, Timer A and Timer B can be used as general counter respectively. Timer A and Timer B contain one powerful PWM function and controlled by corresponding control registers.

This function can be easily configured.

Each timer's function summary is shown as [Table] 5-8

[Table] 5-8 Summary of Timer Function for GPM6P1816A

	Timer Counter	PWM	CAPTURE	ENVELOPE DETECT
Timer A	YES	YES	None	None
Timer B	YES	YES	None	None

5.8.2. Timer A (8-bit down Count Timer)

Timer A is a 8-bit down count timer. Timer A is special for generating carrier signal in IR control application. 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 sent 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 [4:3]).

8-bit down-count Timer A module has the following features:

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

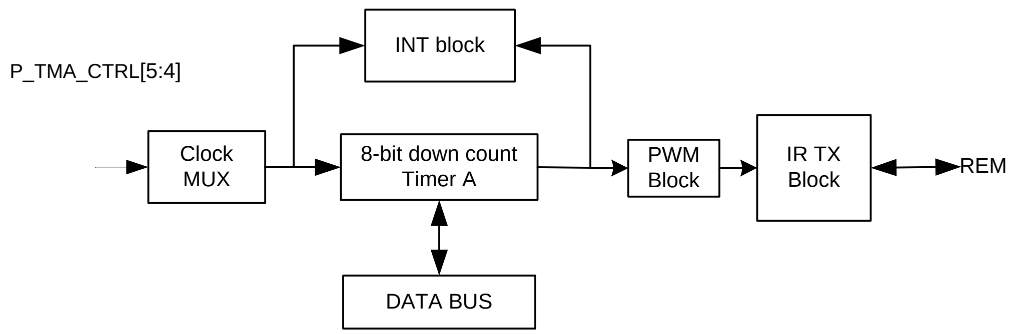


Figure 5-20 Timer A block diagram

5.8.3. Timer A PWM with Carrier Signal Mode

Timer A can be configured as PWM mode for generating carrier signal. In PWM with carrier signal mode, the 8-bit timer is a 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 first 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 the 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 sources (Fosc, Fosc/4 or Fosc/64) by P_TMB_CTRL [5:4] (TMBCLK [1:0]), Timer A's carrier signal on/ off is controlled by Timer A's enabled/ disabled control bit (TMAES) directly. In addition, PWM output function also can be disabled by writing 1 to register IRENB($17.6).

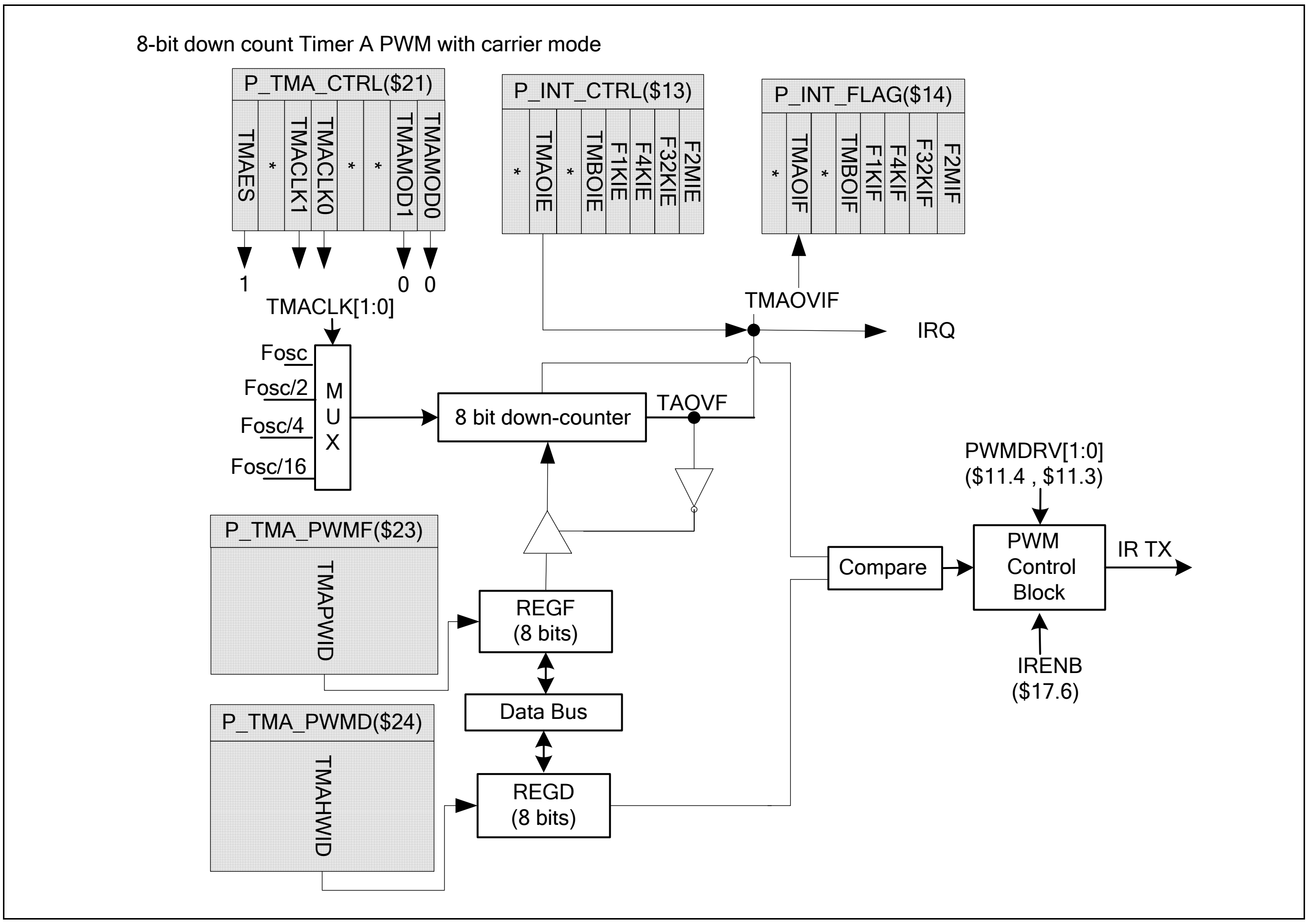


Figure 5-21 Timer A PWM mode diagram

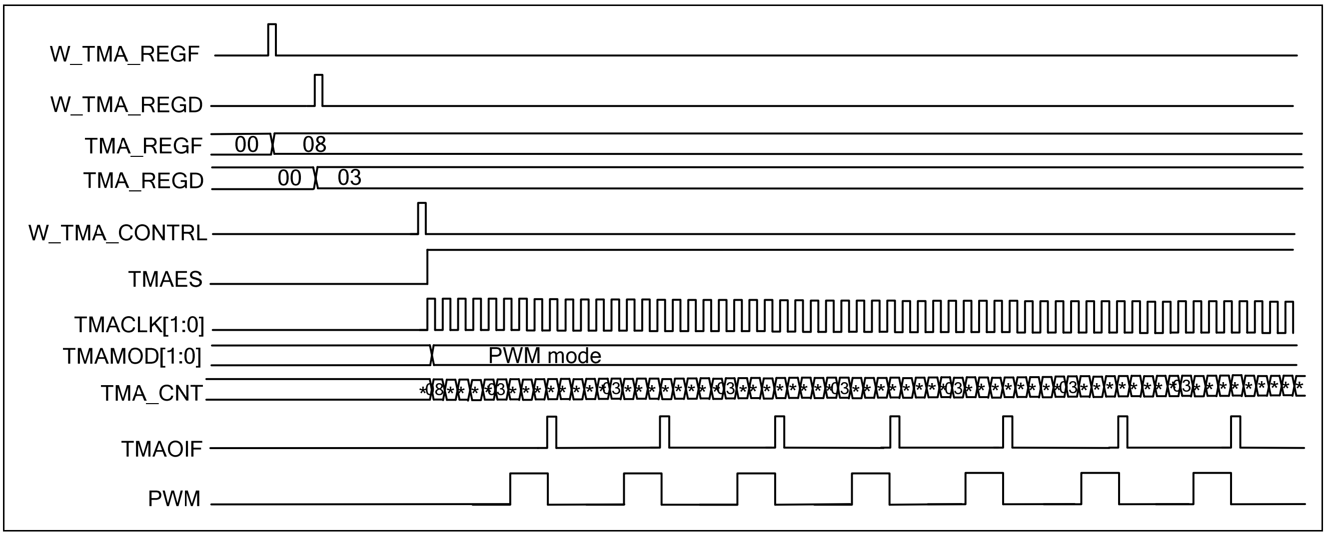


Figure 5-22 Timer A Normal PWM generation without envelope

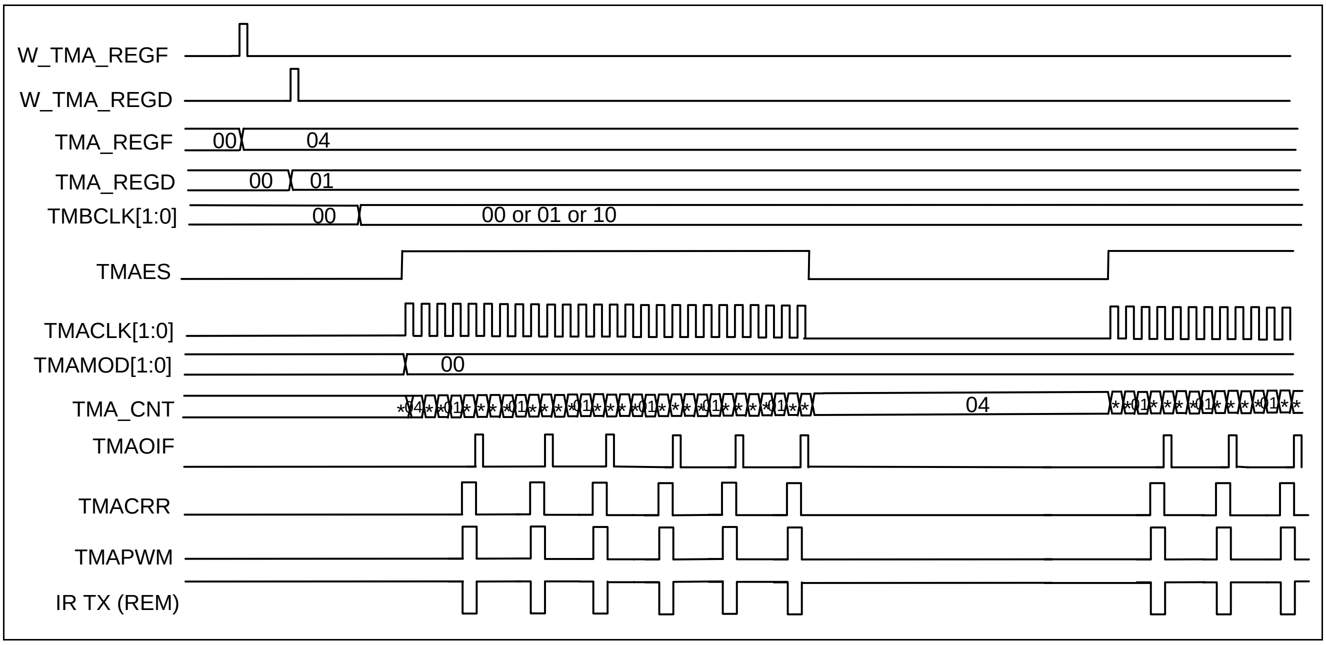
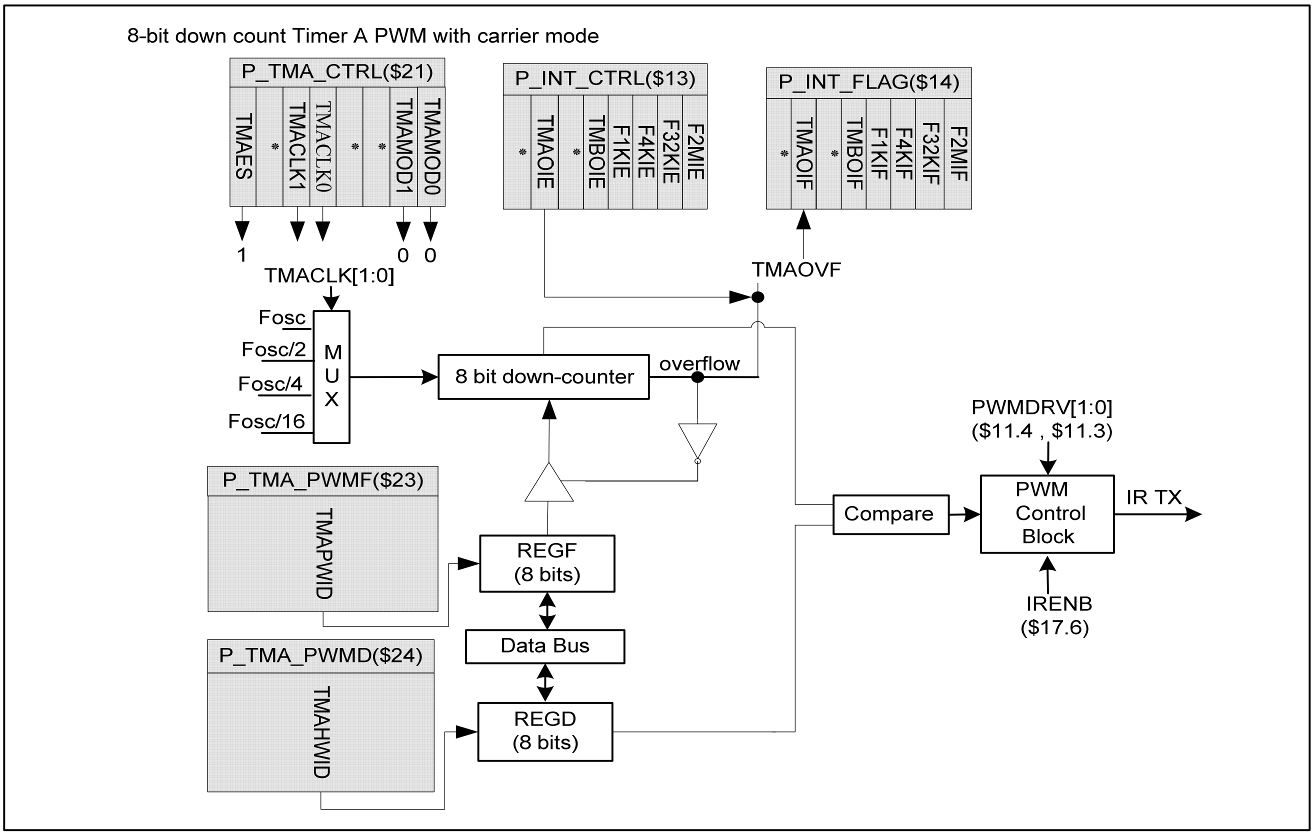


Figure 5-23 The Waveform of timer A PWM with carrier signal mode (1/5 duty, on/off control by TMAES)

Another method to generate a PWM envelope signal is 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 carrier signal as its input clock. And Timer B register must be written in the right data, which represents the carry number. When TMBOVF happen, another value must be written into Timer B register, which represents the no carry clock number. Envelope with carrier is on or off only when Timer B overflow events occur one by one. Then, the PWM envelope signal will be generated at REM port at last.



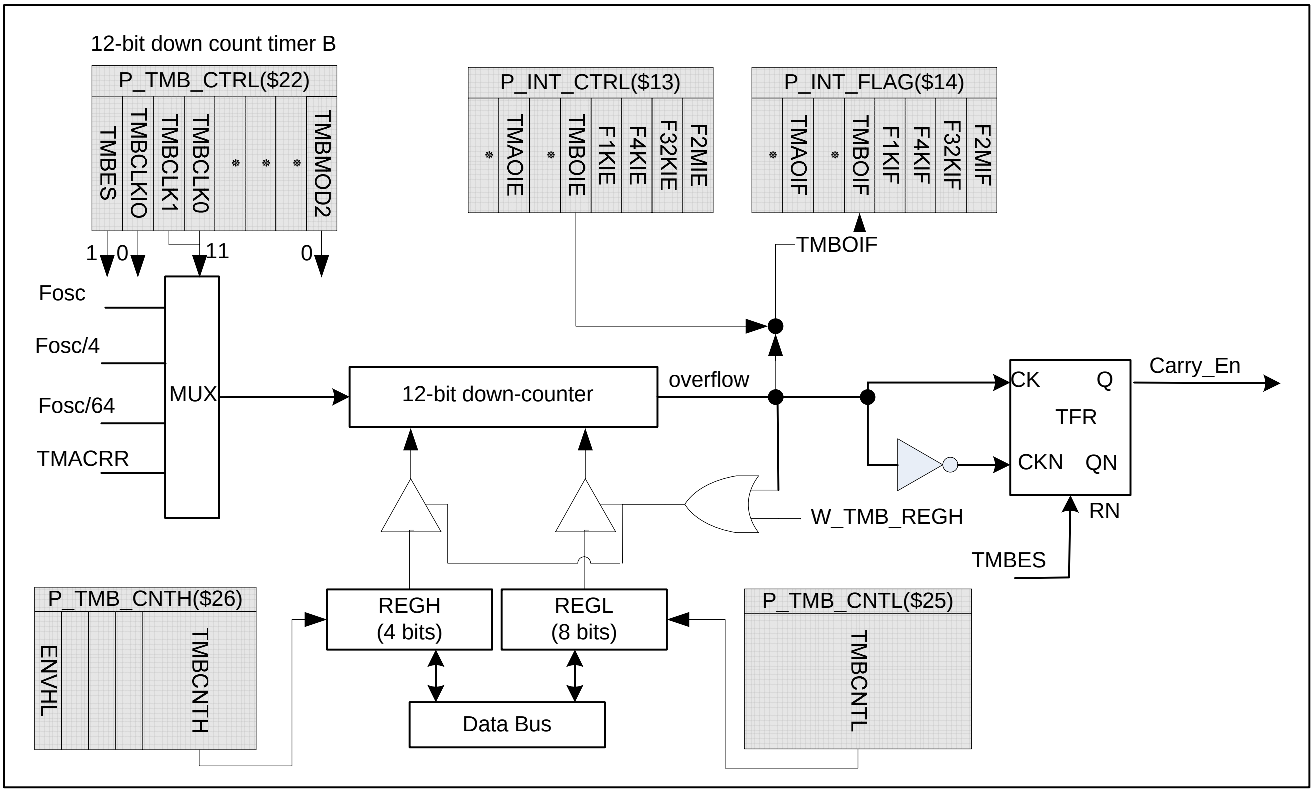


Figure 5-24 PWM Envelope Generated by Timer A & Timer B diagram

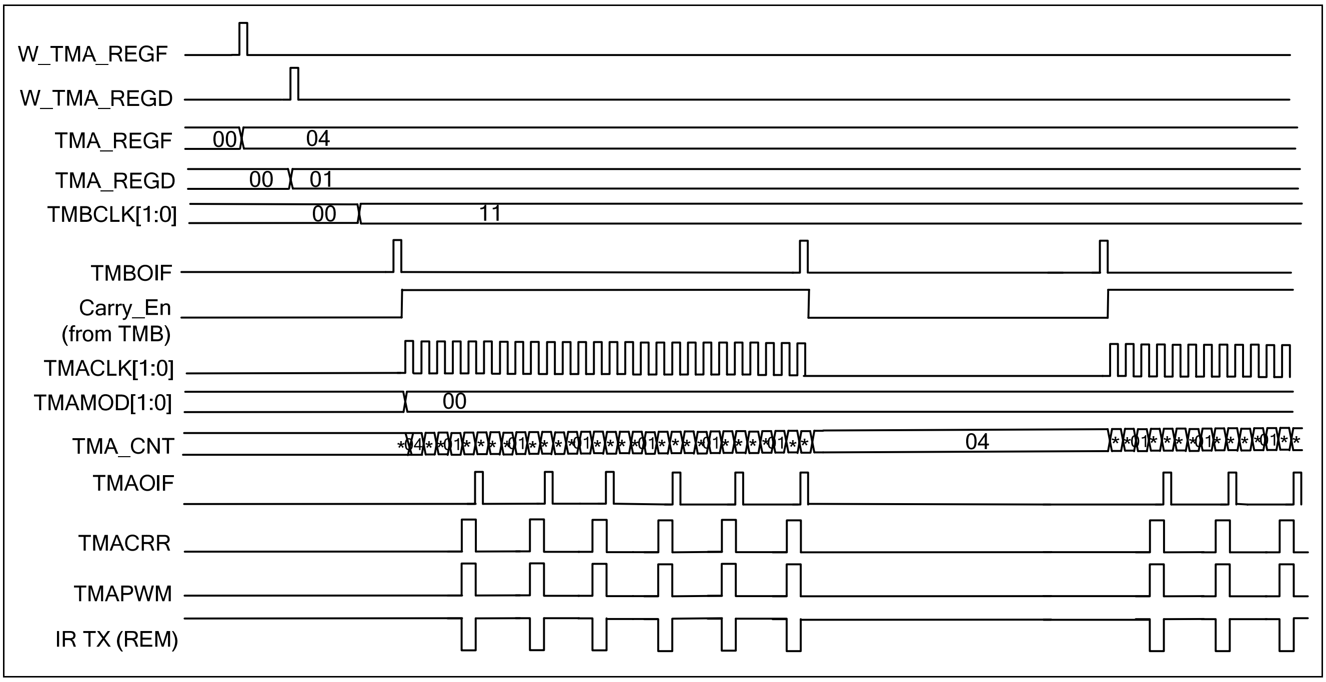


Figure 5-25 The Waveform of Timer A PWM with carrier signal mode (1/5 duty, on/off control by Timer B overflow events)

5.8.4. PWM without Carrier Signal Mode

PWM without carrier signal mode is used to generate envelope PWM signal without carrier signal. In this mode, IR TX (REM) pin just output high or low, and is controlled by TimerA's enabled or disabled control bit or Timer B's overflow events in turn. The same as PWM with carrier signal mode, the 8-bit timer is a down counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When Timer A is started, the value of 8-bit pre-value Register (P_TMA_PWMF, $23) will first 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 it is sent to IR TX pin.

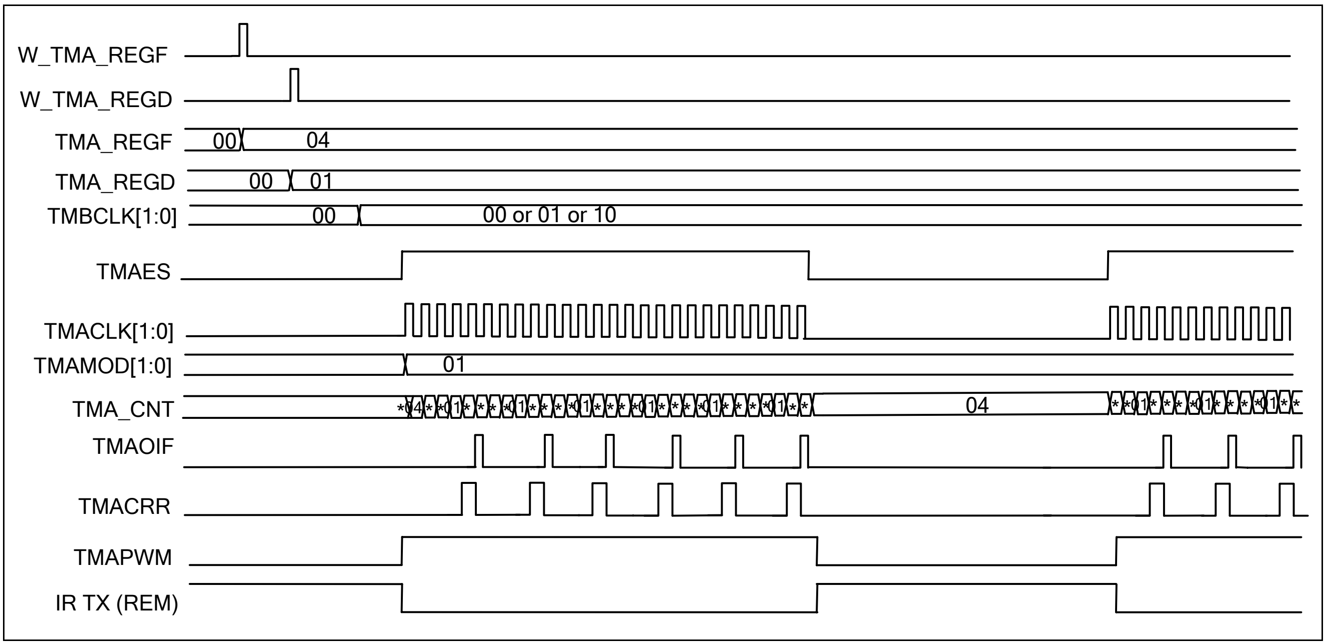


Figure 5-26 The Waveform of Timer A PWM without carrier signal mode (on/off control by TMAES)

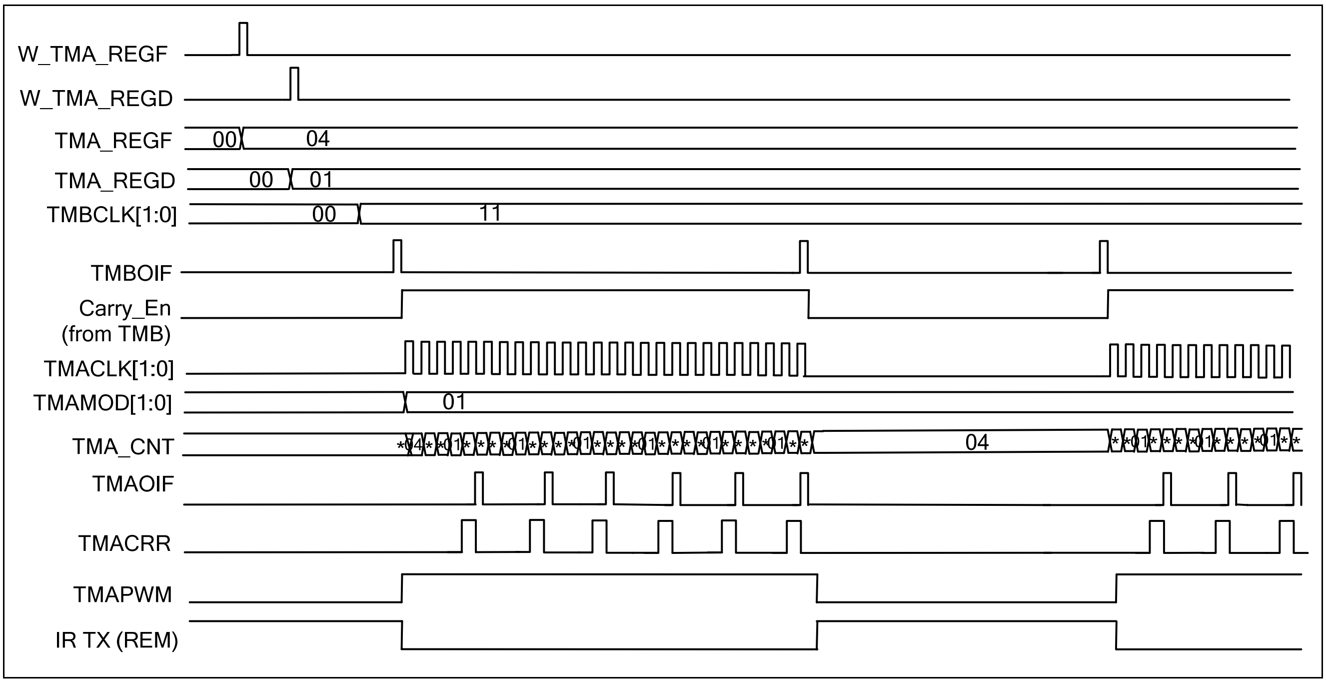


Figure 5-27 The Waveform of Timer A PWM without carrier signal mode (on/off control by Timer B overflow events)

5.8.5. Timer A register

Timer A Control Register (P_TMA_CTRL, $0021)

BIT	7	6	5	4	3	2	1	0
Name	TMAES	--	TMACLK1	TMACLK0	R/0	R/0	TMAMOD1	TMAMOD0
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] **TMAES:** Timer A enable/disable control.
0, disable; (C_TMAES_DIS)
1, enable. (C_TMAES_EN)

Bit [6] No function.

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 [1:0] **TMAMOD[1:0]:** Timer A mode setting
00: PWM (C_TMAMOD_WTC)
01: PWM1 (enter the mode, PWM out always high) (C_TMAMOD_WOC)

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/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] **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)

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	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] **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

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] **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] 5-19 Set Timer A as PWM with carrier signal mode.

```
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
```

5.8.6. 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] 5-20 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38 KHz with 2/5 duty (clock source=Fosc/1).

```
    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
```

5.8.7. Timer B

Timer B is a special for envelope signal generation in IR controller application. The 12-bit timer is a 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.

Timer B module has the following features:

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

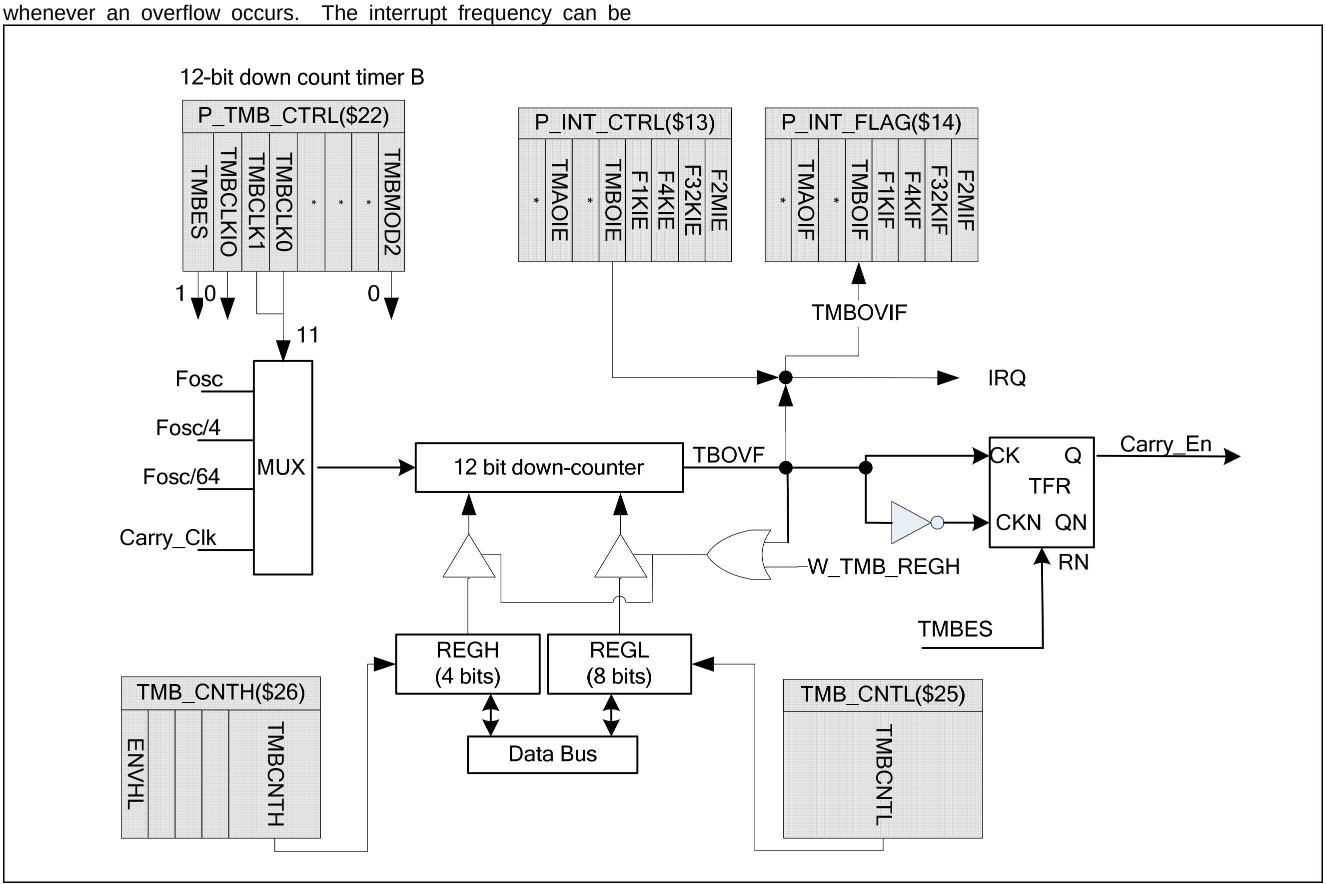


Figure 5-28 Timer B block diagram

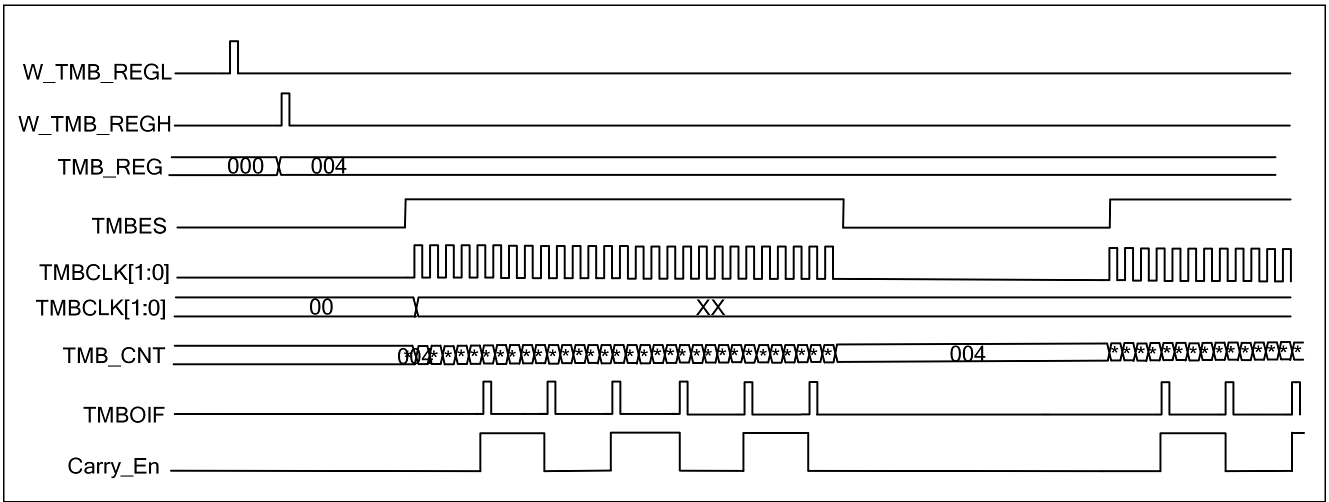


Figure 5-29 The Waveform of Envelope controlled by Timer B

Timer B Control Register (P_TMB_CTRL, $0022)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBES	TMBCLKIO	TMBCLK1	TMBCLK0	R/0	R/0	--	TMBMOD2
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] **TMBES**: Timer B enable/disable selected bit.
0 = disable (C_TMBES_DIS)
1 = enable (C_TMBES_EN)

Bit [6] **TMBCLKIO:** Timer B clock source from IO selection. (using RFC)
0 = Timer B clock source setting from bit[5:4].
1 = Timer B clock source from IO portC[7].

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 [3:1] Reserved

Bit [0] **TMBMOD2:** Timer B mode 2 control
0: Timer B mode 2 function disable.
1: Timer B mode 2 function enable.

Timer B 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/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] **TMBCNTL[7：0]:** Timer B low byte 8-bit pre-value for the counter.
Read: Timer B Count Low Byte Value (R)
Write: Timer B Pre-Load Count Low Byte Value (W)

Timer B High 4-bit Data Register (P_TMB_CNTH, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	ENVHL	R/0	R/0	R/0	TMBCNTH3	TMBCNTH2	TMBCNTH1	TMBCNTH0
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] **ENVHL:** Envelope Mark and Space setting (Timer B mode 2 use)
0 : space
1 : mark.

Bit [6:4] Reserved

Bit [3:0] **TMBCNTH[3：0]:** Timer B High byte 4-bit pre-value for the counter.
Read: Timer B Count High Byte Value (R)
Write: Timer B Pre-Load Count High Byte Value (W)

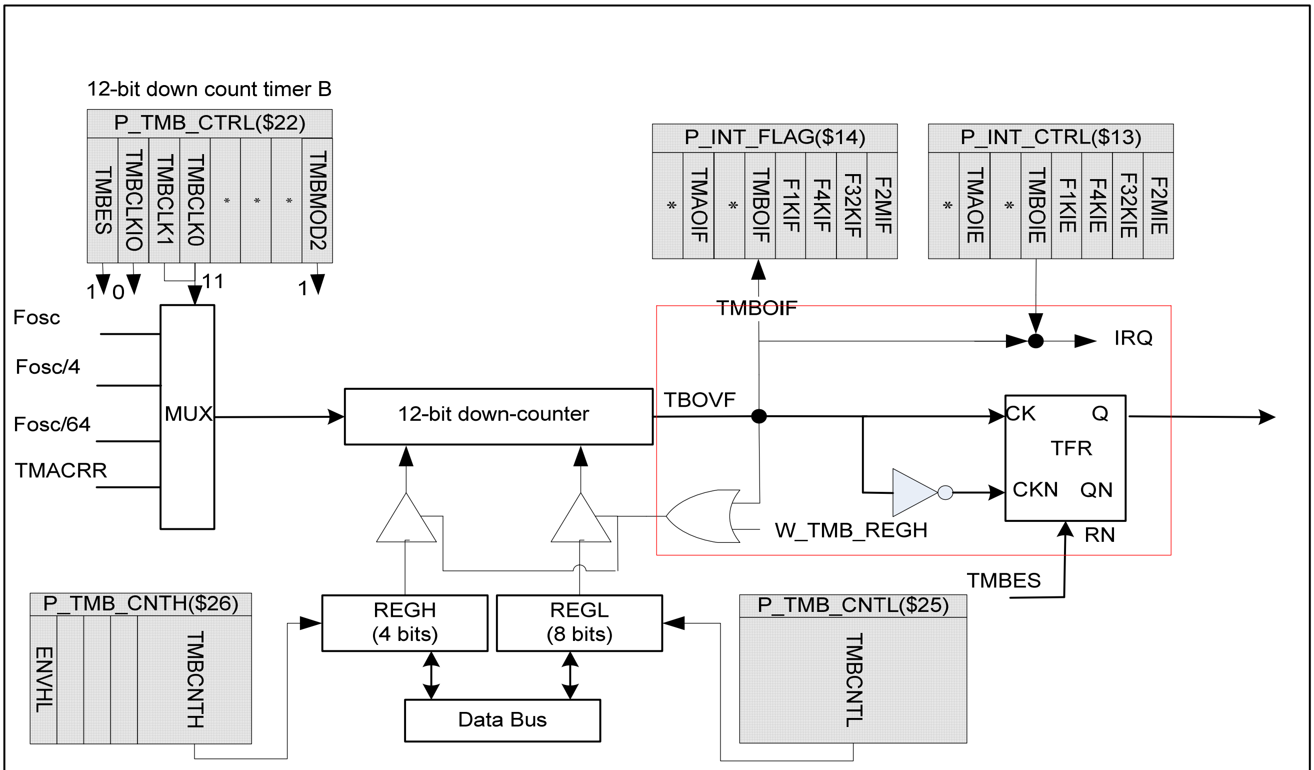
[Example] 5-21 Set Timer B selects timer A carrier signal as counter clock.

```
    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
```

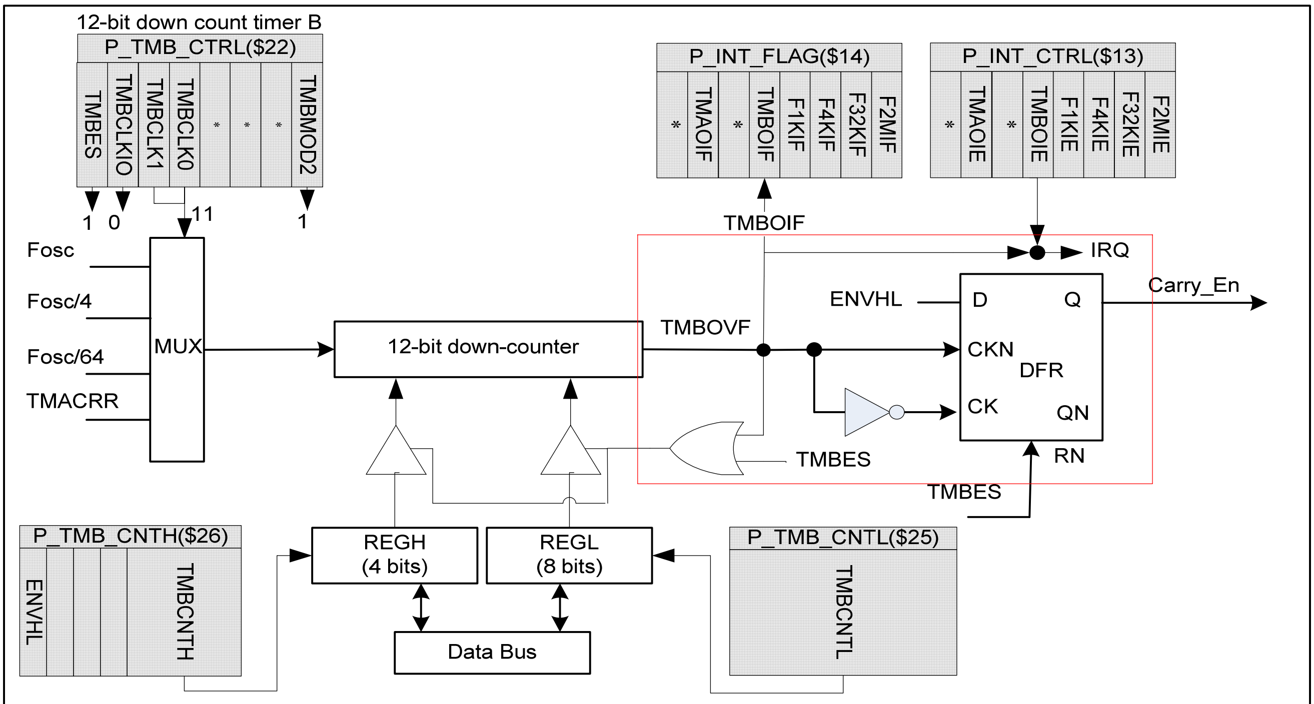
5.8.8. Timer B Mode 2 operation

In order to improve the function Timer B, we add a Timer B mode 2 operation. Mainly adds the features:

We can use a register ENVHL ($26.7) to set Mark & Space in PWM mode in which envelope is controlled by Timer B.



Envelop PWM generated by timer A & timer B



Envelop PWM generated by timer A & timer B mode 2

Figure 5-30 The difference between Timer B mode 2 with the original one in PWM mode

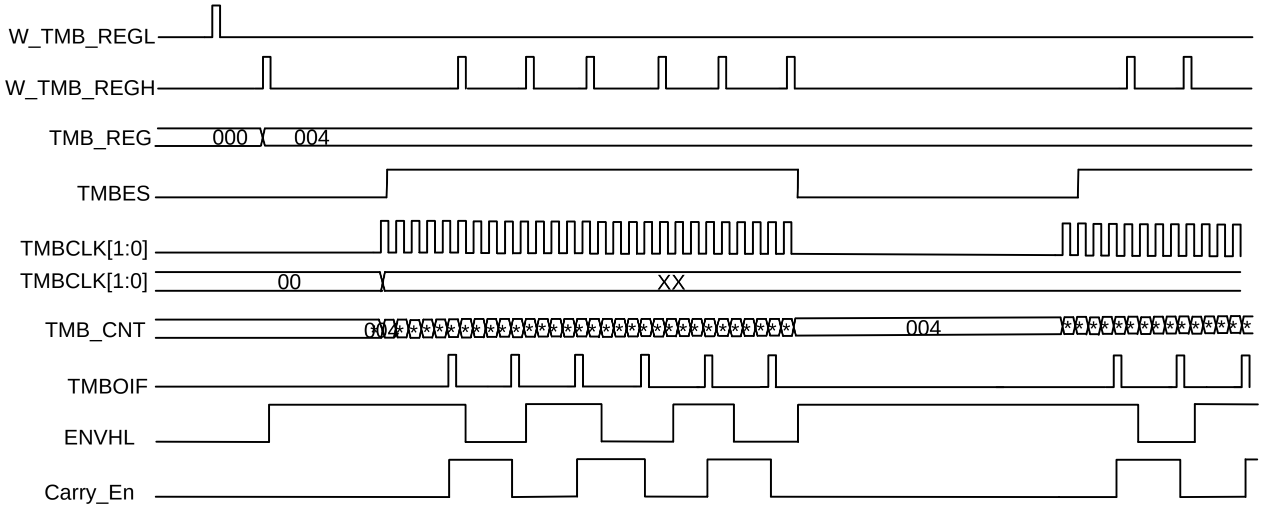


Figure 5-31 The Waveform of Envelope control by Timer B in mode 2

[Example] 5-22 Set Timer B selects timer A carrier signal as counter clock in mode 2.

```
LDA    #$FC                      ; Before starting timer, set Timer B counter initial value first
STA    P_TMB_CNTL                ; set low 8-bit pre-value
LDA    #$1F                      ; bit [7] set ENVHL=1 , Envelope is Mark.
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
------------------------------------   wait   timer B overflow -----------------------------------
LDA    #$0F                      ; bit[3:0] keep original setting, bit[7] set ENVHL=0, Envelope is Space
STA    P_TMB_CNTH
------------------------------------   wait   timer B overflow -----------------------------------
LDA    #$1F                      ; bit[3:0] keep original setting, bit[7] set ENVHL=1, Envelope is Mark
STA    P_TMB_CNTH
```

5.9. LCD function

5.9.1. LCD Introduction

GPM6P1816A supports LCD display COMMON x SEGMENT = 4 X 32 (Maximum).

Frame rate is around 85Hz.

SEGMENT [15:0] are dedicated SEGMENT pin.

SEGMENT [23:16] are shared PADs with Port A [7:0].

SEGMENT [31:24] are shared PADs with Port C[7:0].

LCD driver supports 1/2 Bias, 1/3 Bias, 1/2 Duty, 1/3 Duty, 1/4 Duty. For more information about the functions, please check 0 LCD Register. LCD drive can still work in sleep mode by keeping 32KHz oscillator alive.

5.9.2. LCD Register

LCD Control Register (P_LCD_CTRL, $0032)

BIT	7	6	5	4	3	2	1	0
Name	R/0	R/0	R/0	Green Mode	LBSDT1	LBSDT0	LCDMD1	LCDMD0
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 [7:5] Reserved.

Bit [4] **Green Mode**: Set LCD operating in Green mode.

0 = Green disable

1 = Green enable

Bit [3:2] **LBSDT[1:0]:** LCD bias and duty setting

00 = 1/2 BIAS 1/2 DUTY

01 = 1/2 BIAS 1/3 DUTY

10 = 1/3 BIAS 1/3 DUTY

11 = 1/3 BIAS 1/4 DUTY

Bit [1:0] **LCDMD[1:0]:** LCD display mode control

11 = display off

10 = all off

01 = all on

00 = normal;

LCD DPRAM Register (COM[3:0], $005F~$0050)

ADD	NAME	BIT7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
50	COM0	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
51		SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
52		SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
53		SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
54	COM1	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
55		SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
56		SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
57		SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
58	COM2	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
59		SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
5A		SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
5B		SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
5C	COM3	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
5D		SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
5E		SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
5F		SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24

Via setting IO port A and port C as segment pin to define the LCD panel size.

Port A Segment Function Control Register (P_IOA_SEL, $004A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	PAS7	PAS6	PAS5	PAS4	PAS3	PAS2	PAS1	PAS0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

Bit [7:0] **PAS0[7:0]:** Port A Segment function control.

0: set port A as normal I/O.

1: set port A as segment output.

Port C Segment Function Control Register (P_IOC_SEL, $004C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	00h							

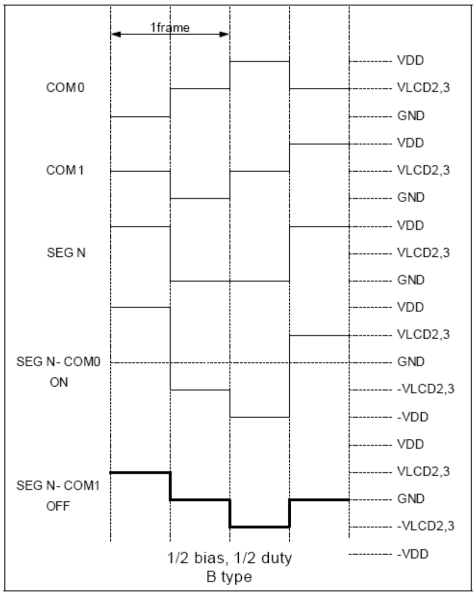
Bit [7:0] **PCS0[7:0]:** Port C Segment function control.

0: set port C as normal I/O.

1: set port C as segment output.

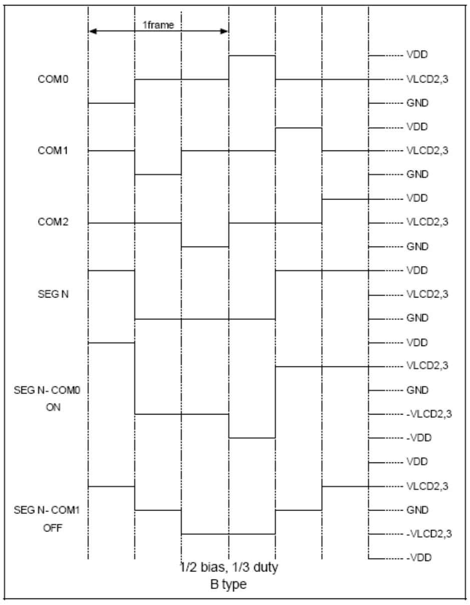
5.9.3. LCD Waveform

◆ LCD waveform for 1/2 Bias , 1/2 Duty.



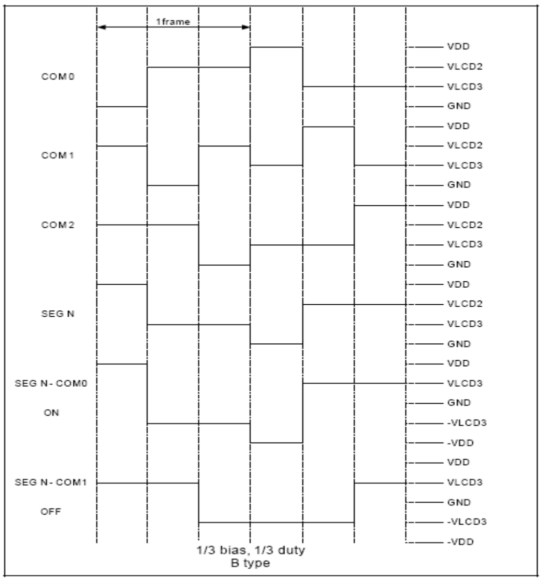
1/2 bias, 1/2 duty
B type

◆ LCD waveform for 1/2 Bias , 1/3 Duty.

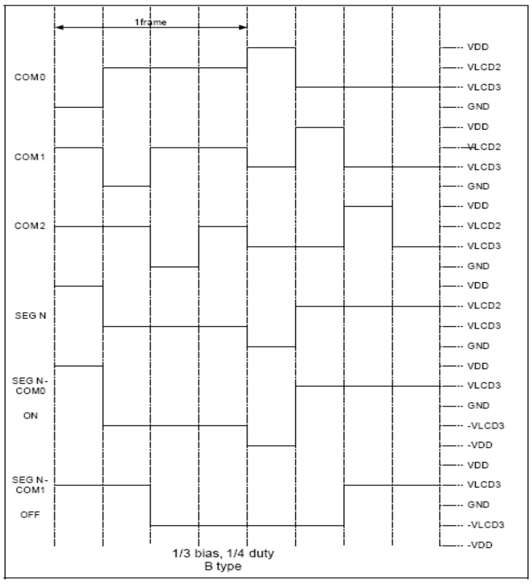


1/2 bias, 1/3 duty
B type

◆ LCD waveform for 1/3 Bias , 1/3 Duty.



◆ LCD waveform for 1/3 Bias , 1/4 Duty.



5.10. LVD (Low Voltage Detect)

A real time low voltage detector is built-in GPM6P1816A 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 < 2.1V .
1: LVD trigger H when VDD < 2.4V .

Bit [0] **LVD:** Low voltage detect, read only.
0: VDD is upper 2.1V / 2.4V
1: VDD is under 2.1V / 2.4V

5.11. RFC (Resistance Frequency Converter)

The RFC circuit contains a RC oscillator circuit and a 12-bits counter (Timer B) to calculate the resistance of temperature or humidity sensor related to reference resistance. The circuit is as follows**:**

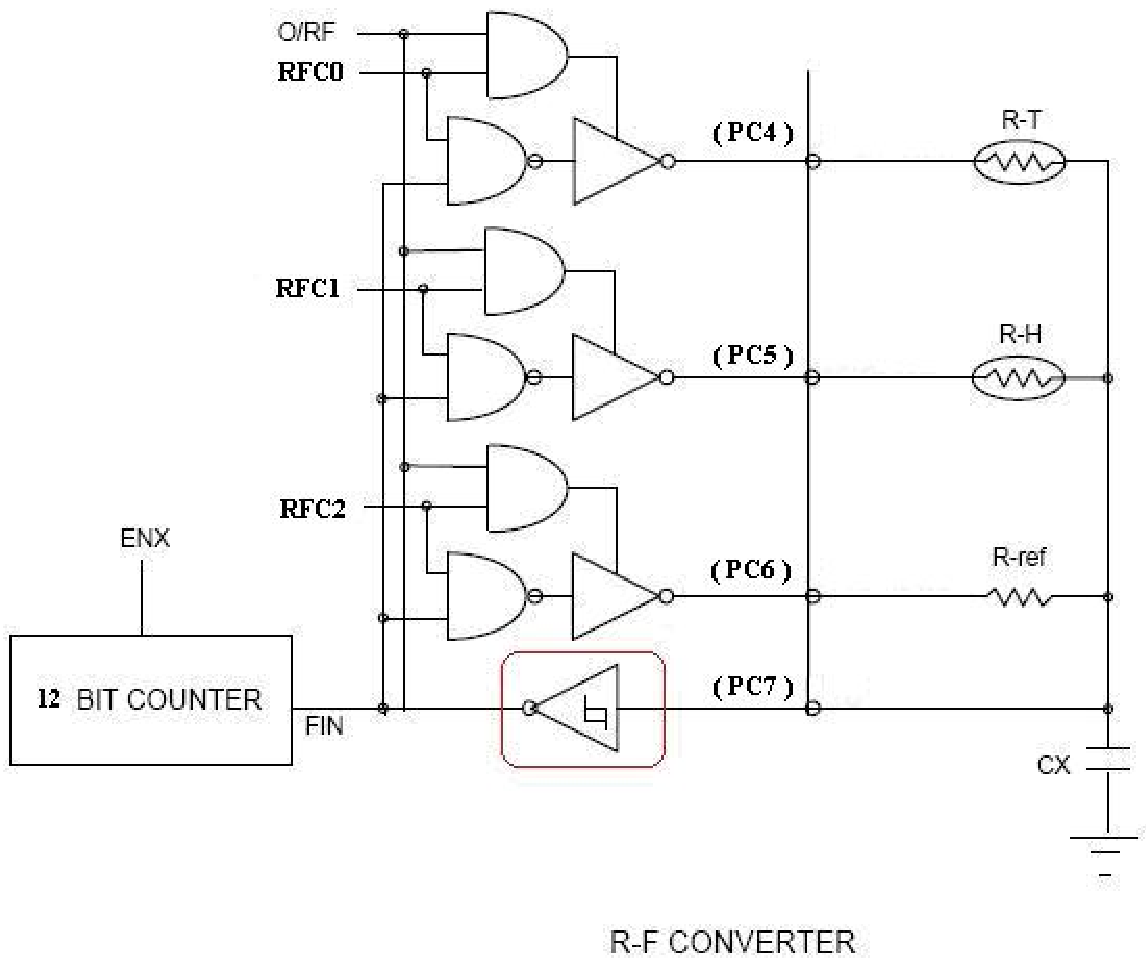


Figure 5-32 The Circuit of RFC

R-T is a resistance of temperature sensor relative,
R-H is a resistance of humidity sensor relative,
R-ref is a reference resistance.

We can use timer B to calculate the frequency of those resistances individually. According to the formula: N=T*F=T/RC, and compare with the reference resistance to get the resistance of R-T / R-H.

Timer B Control Register (P_TMB_CTRL, $0022)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBES	TMBCLKIO	TMBCLK1	TMBCLK0	R/0	R/0	--	TMBMOD2
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 [6] **TMACLKIO:** Timer B clock source from IO selection. (RFC using)
0 = Timer B clock source setting from bit[5:4].
1 = Timer B clock source from IO portC[7]..

RFC Control Register (P_RFC_CTRL, $001E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	R/0	R/0	R/0	R/0	RFCEN	RFC2	RFC1	RFC0
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 [7:4] Reserved

Bit [3] **RFCEN:** RFC control register.
0 = RFC function disable.
1 = RFC function enable.

Bit [2:0] **RFC[2:0] :** RFC port control register
0: RFC port output disable.
1: RFC port output enable.

[Example] 5-23 RFC function

```
LDA   #$08              ; set timer B as RFC use
STA   P_TMB_CTRL
LDA   #$FF              ; initial timer B counter value.
STA   P_TMB_CNTL
LDA   #$0F
STA   P_TMB_CNTH
LDA   #$09              ; enable RFC function and set RFC0 as output port
STA   P_RFC_CTRL
------------------------------------   wait   fixed time ----------------------------------
LDA   #$08              ; keep enable RFC but disable RFC0 port.
LDA   P_TMB_CNTL        ; read the counter value of timer B to calculate the frequency
LDA   P_TMB_CNTH
LDA   #$0A              ; keep enable RFC and enable RFC1 port
------------------------------------   wait   fixed time ----------------------------------
LDA   #$08              ; keep enable RFC but disable RFC1port.
LDA   P_TMB_CNTL        ; read the counter value of timer B to calculate the frequency
LDA   P_TMB_CNTH
LDA   #$0C              ; keep enable RFC and enable RFC2 port
```

5.12. IR Transfer Module

TX is an analog block of GPM6P1816A, which can drive IR LED by TX. User can adjust PWM output driving capability by setting value in PWMDRV [1:0],

TMAPWM signal (as showed in Figure 5-33) controls LED driver MOS. In PWM mode, Timer A generates PWM signal, and the PWM duty, frequency, on/off switch can be precisely controlled by Timer A. The Envelope PWM signal can be generated by Timer A and Timer B. And it has been illustrated in timer instruction.

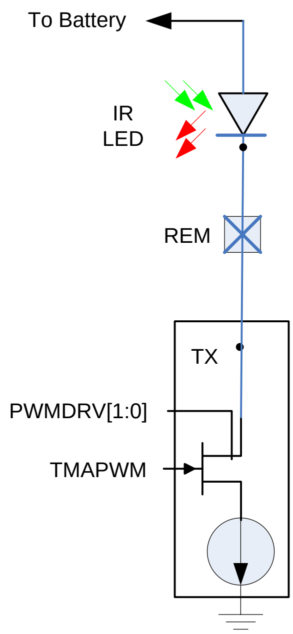


Figure 5-33 TX module diagram

Timer A PWM Drive Register (P_PWM_DRV, $11)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	R/0	R/0	R/0	PWMDRV1	PWMDRV0	--	--	--
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 [7:5] Reserved

Bit [4:3] **PWMDRV[1:0] :** PWM driving current selected bits.

00 = PWM 1/4 driving current

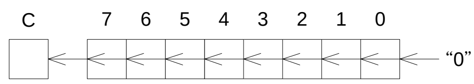
01 = PWM 2/4 driving current

10 = PWM 3/4 driving current

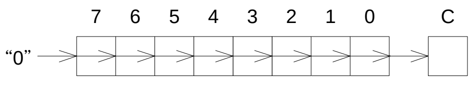
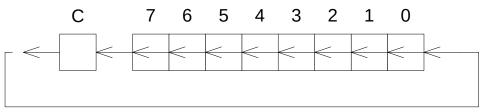
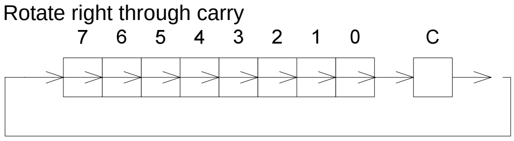
11 = PWM 4/4 driving current

Bit [2:0] No function.

5.13. 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) + ??	--------
23.	BCS aa	B0	2	2(C)	Branch if carry bit set If (C) = 1, then pc ← (pc) + ??	--------
24.	BEQ aa	F0	2	2(C)	Branch if equal If (Z) = 1, then pc ← (pc) + ??	--------
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) + ??	--------
28.	BNE aa	D0	2	2(C)	Branch if not equal If (Z) = 0, then pc ← (pc) + ??	--------
29.	BPL aa	10	2	2(C)	Branch if plus If (N) = 0, then pc ← (pc) + ??	--------
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) + ??	--------
32.	BVS aa	70	2	2(C)	Branch if overflow bit set If (V) = 1, then pc ← (pc) + ??	--------
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, (A) – (M)	N-----ZC
38.	CMP aa	C5	2	3		
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, (X) – (M)	N-----ZC
46.	CPX aa	E4	2	3		
47.	CPX aaaa	EC	3	4		
48.	CPY #dd	C0	2	2	Compare memory data with Y-register, (Y) – (M)	N-----ZC
49.	CPY aa	C4	2	3		
50.	CPY aaaa	CC	3	4		
51.	DEC aa	C6	2	5	Decrement M ← (M) - 1	N-----Z-
52.	DEC aa, X	D6	2	6		
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 A ← (A) ⊕ (M)	N-----Z-
58.	EOR aa	45	2	3		
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 M ← (M) + 1	N-----Z-
66.	INC aa, X	F6	2	6		
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 Pc ← jump address	--------
72.	JMP (aaaa)	6C	3	6		
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 A ← (M)	N-----Z-
75.	LDA aa	A5	2	3		
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 X ← (M)	
83.	LDX aa	A6	2	3		
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 Y ← (M)	
88.	LDY aa	A4	2	3		
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 A ← (A) v (M)	
99.	ORA aa	05	2	3		
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)		
120.	RTI	40	1	6	Return from interrupt	restored

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
					Sp ← sp + 1, P status ← (sp), sp ← sp + 1, pc_L← (sp), sp ← sp +1, pc_H ← (sp)	
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 crossing 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.

6. ELECTRICAL CHARACTERISTICS

6.1. Absolute Maximum Ratings

Characteristics	Symbol	Ratings
DC Supply Voltage	V_+	< 4.0V
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_{RMT}	500mA
VDD Total MAX Current	I_{VDDM}	100mA
VSS Total MAX Current	I_{VSSM}	200mA

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 (T_A = 25℃)

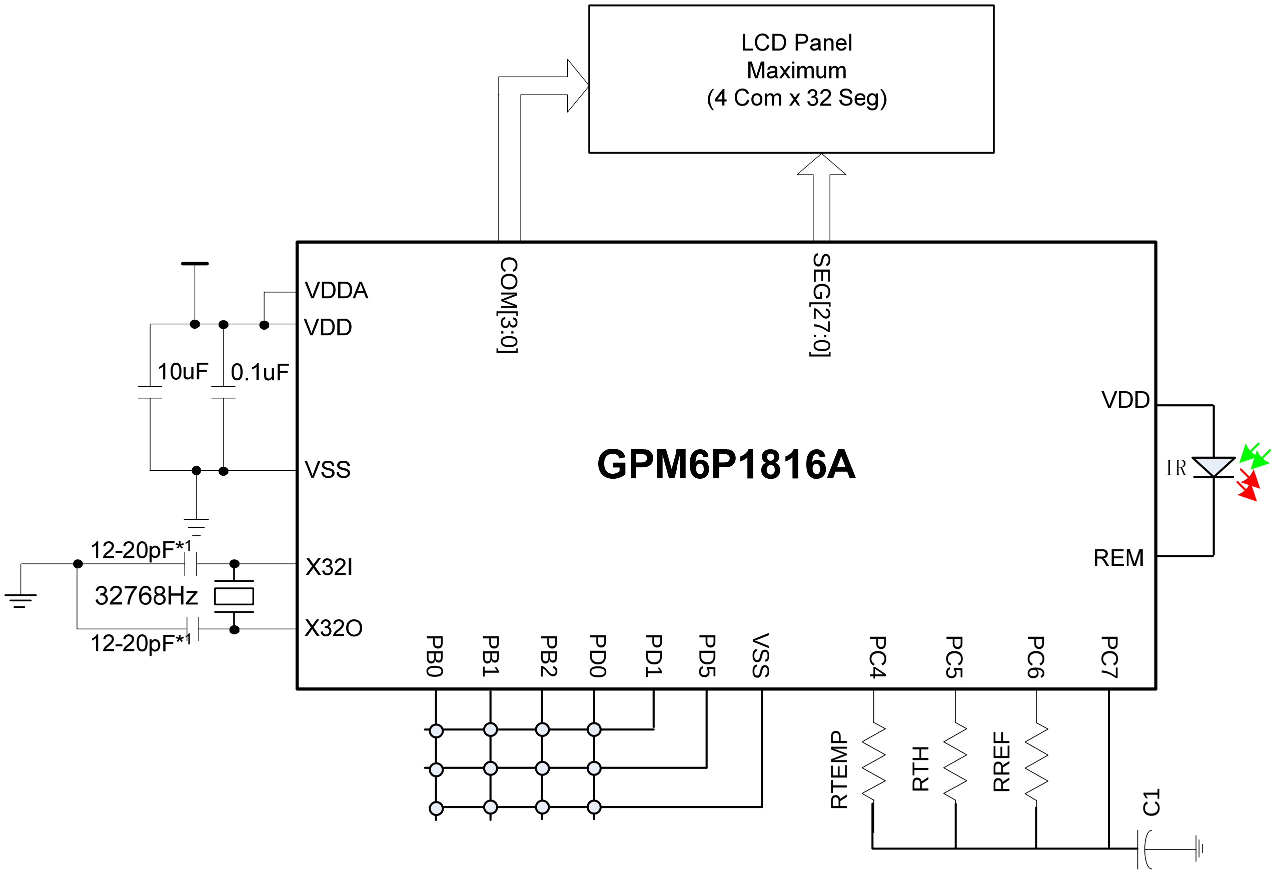
Characteristics	Limit			Unit	Test Condition
	Min.	Typ.	Max.		
OSC Accuracy @ Freq=4MHz					
OSC Variation	-3.0	±1.5	3.0	%	VDD = 2.0V - 3.6V, T_A =25°C
OSC Accuracy @ Freq=32768Hz					
OSC Variation	-7.0	±3.5	7.0	%	VDD = 2.0V - 3.6V, T_A =25°C

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

Characteristics	Symbol	Limit			Unit	Test Condition
		Min.	Typ.	Max.		
Operating Voltage1	VDD	2.0	-	3.6	V	F_{CPU} = 4.0MHz, For 2-battery
Operating Voltage2	VDD	2.4	-	3.6	V	F_{CPU} = 8.0MHz, For 2-battery
Operating Current	I_{OP}	-	2.0	3.0	mA	F_{CPU} = 8.0MHz @ 3.6V, no load
M-Type key Standby Current	I_{MSTBY}	-	-	1.0	uA	VDD = 3.8V, all clock off.
Halt mode current	I_{halt}	-	5.0	8.0	uA	VDD = 3.8V, LCD on, no load.
Green Mode current	I_{green}	-	4.0	7.0	uA	VDD = 3.8V, LCD on, 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 PA, PB, PC, PD	V_{OH}	0.8VDD	-	-	V	VDD = 3.0V I_{OH} = -6mA
Output Low Level PA, PB, PC, PD	V_{OL}	-	-	0.2VDD	V	VDD = 3.0V I_{OL} = 16mA
Input Pull High Resistor PA, PB, PC, PD	R_H	30	50	70	Kohm	Pull High VDD = 3.0V
Input Pull Low Resistor PA, PB, PC, PD	R_L	30	50	70	Kohm	Pull Low VDD = 3.0V
PWM Driving Current	I_{PWM}	200	-	-	mA	VDD = 3.0V,V_{RMT} =3.0V PWMDRV[1:0]=11
LVR Active Voltage (by option)	V_{LVR}	1.7	1.85	2.0	V	LVRVSEL=0
		2.1	2.25	2.4	V	LVRVSEL=1

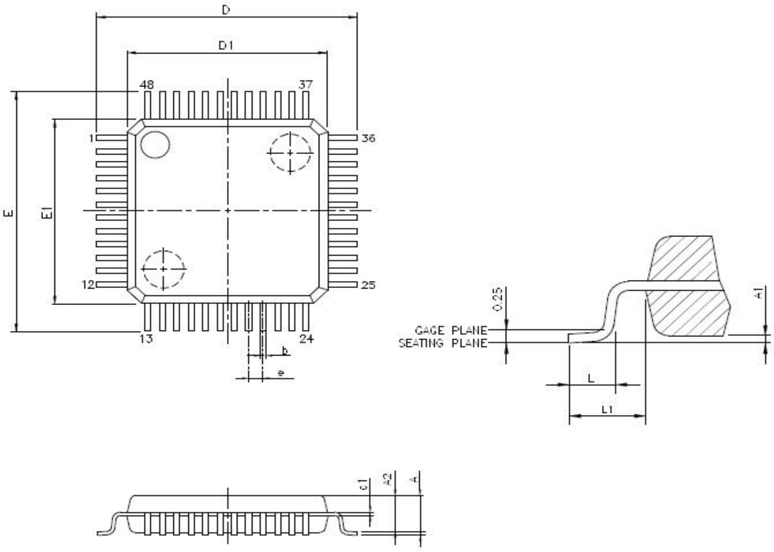
7. APPLICATION CIRCUIT

7.1. Application Circuit



Note1: These capacitor values are for design guidance only. Different capacitor values may be required for different crystal/resonator used.

8. PACKAGE INFORMATION



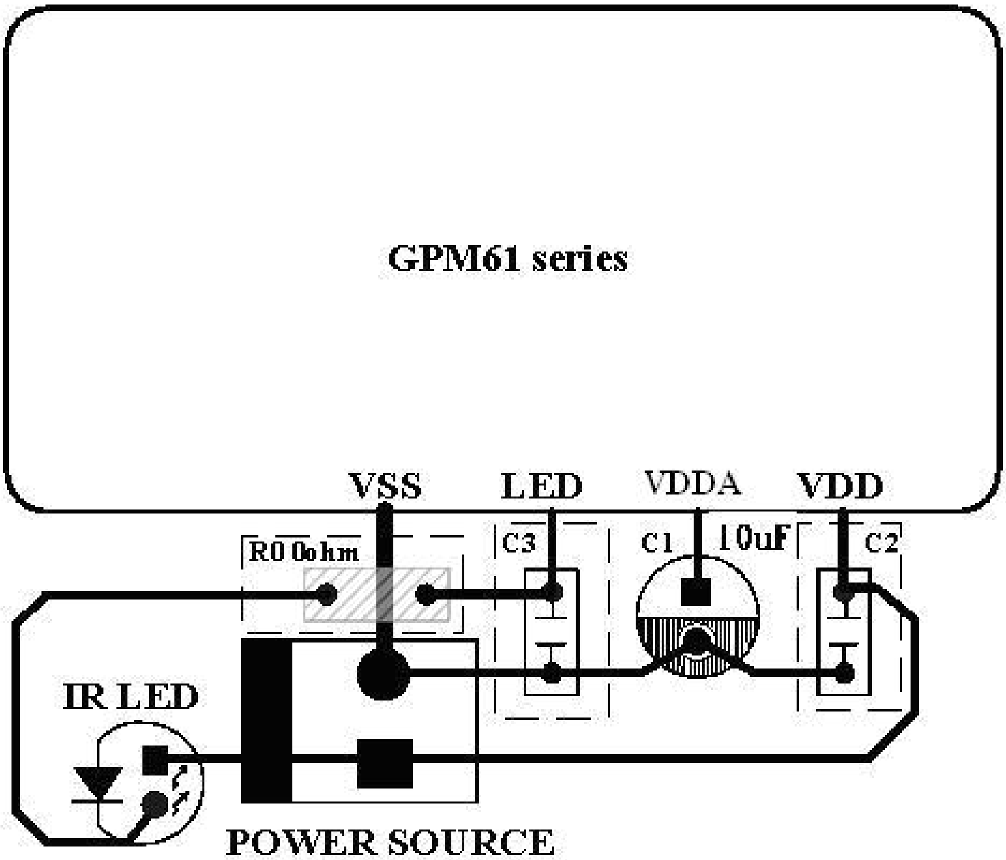
VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	MAX.
A	--	1.6
A1	0.05	0.15
A2	1.35	1.45
c1	0.09	0.16
D	9.00 BSC	
D1	7.00 BSC	
E	9.00 BSC	
E1	7.00 BSC	
e	0.5 BSC	
b	0.17	0.27
L	0.45	0.75
L1	1 REF	

NOTES:

1. JEDEC OUTLINE:MS-026 BBC
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS IMCLUDING MOLD MISMATCH.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.

9. PCB LAYOUT GUIDELINE



To avoid the unexpected noises to end up with abnormal CPU operations, the following cares must be exercised while designing a PCB layout:

1. Forbidden inserting jump zero-ohm resistor in the connect line between VSS pin and power source; this line should be as short as possible, and its width greater than 3mm is recommended.
2. The GND line of all these voltage stabilize intention capacitors should be pulled from power source separately divided from chip GND line.
3. C1 placed between VDDA and VSS must be as close as possible to IC itself, it is necessary for all GPM6P1816A application circuits for power stabilization and power down data protection.
4. C2 must be as close as possible to IC itself; it is a must for GPM6P1816A..
5. C3 can only be placed in some special application for IR LED power stabilization; its GND must be as close as possible to power source GND.
6. The power and GND routes between these devices should be as short and wide as possible; the width should keep greater than 1mm.

10. 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.

11. REVISION HISTORY

Date	Revision #	Description	Page
Mar 05, 2013	0.1	Preliminary revision	63