(Preliminary)

1. GENERAL DESCRIPTION

The MK8A03BP device is an integrated micro-controller that includes an 8-bit RISC CPU core, 192-bytes SRAM, full speed USB interface and an 8K x 16 internal program OTP-ROM for USB general purpose application.

2. FEATURES

- USB 2.0 full speed
  - Fully compliant to USB 2.0 and USB1.1 full speed specification
  - Built-in USB Transceiver
  - Support USB Suspend and Resume
  - Built-in internal 3.3V regulator and USB pull-up resistor
  - One Control IN/OUT, two IN/OUT endpoint
  - Programmable endpoint number from 1 to 15
- Clock Mode
  - External 6MHz crystal oscillation
- RISC 8-bit microcontroller
  - CPU speed can be up to 12MHz
  - 3MHz or 6MHz instruction cycle depend on system clock.
- Internal memory
  - Built-in 192 bytes SRAM
  - Built-in 8Kx16 OTP (One Time Programming) ROM
- Reduction component
  - Built-in internal USB pull-up resistor and DP DM match resistor
- General-purpose I/O ports
  - Up to 40 General Purpose I/O (GPIO) pins
  - support open-drain (Port B) and pull-up (Port A Port B) resistor
  - Port B with pull-up
  - Port A and Port C with interrupt feature
- SPI serial communication
  - Master and slave operation
  - Supports half duplex single data line mode for optical sensors
- Two 8 bit timer capture which can cascade to be one 16 bit capture timer.
  Capture timer registers store both rising and falling edge times
- Two 8 bit auto reload timer interrupt which can cascade to be one 16 bit timer
- Internal low-power wake-up timer during suspend mode

6. Wakeup with no external components

- Watchdog timer(WDT)
- Built-in 4 or 8MHz (by option) clock output pin
- Built-in low voltage reset (LVR) function
- 48 pin package

3. PIN ASSIGNMENT

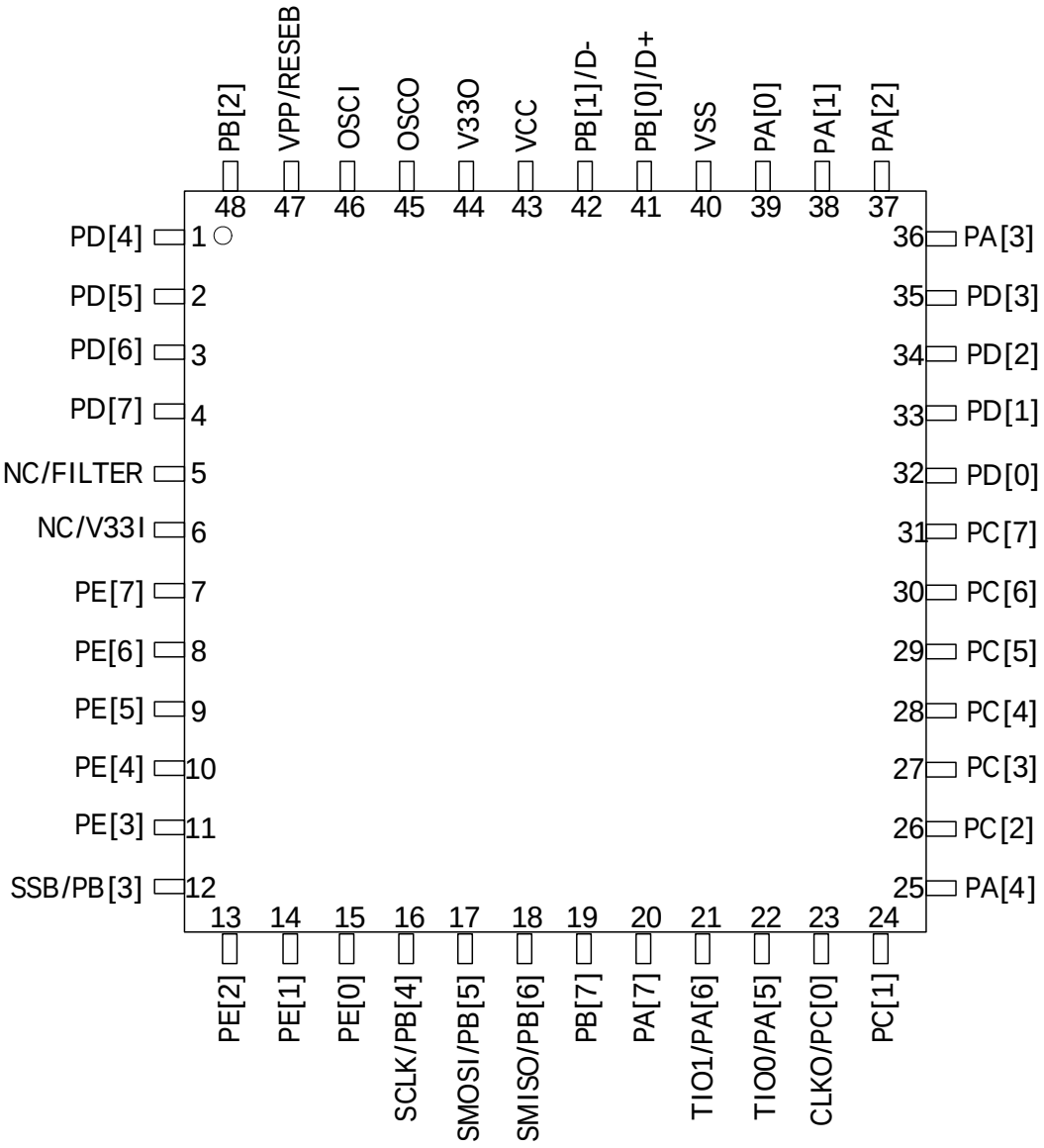


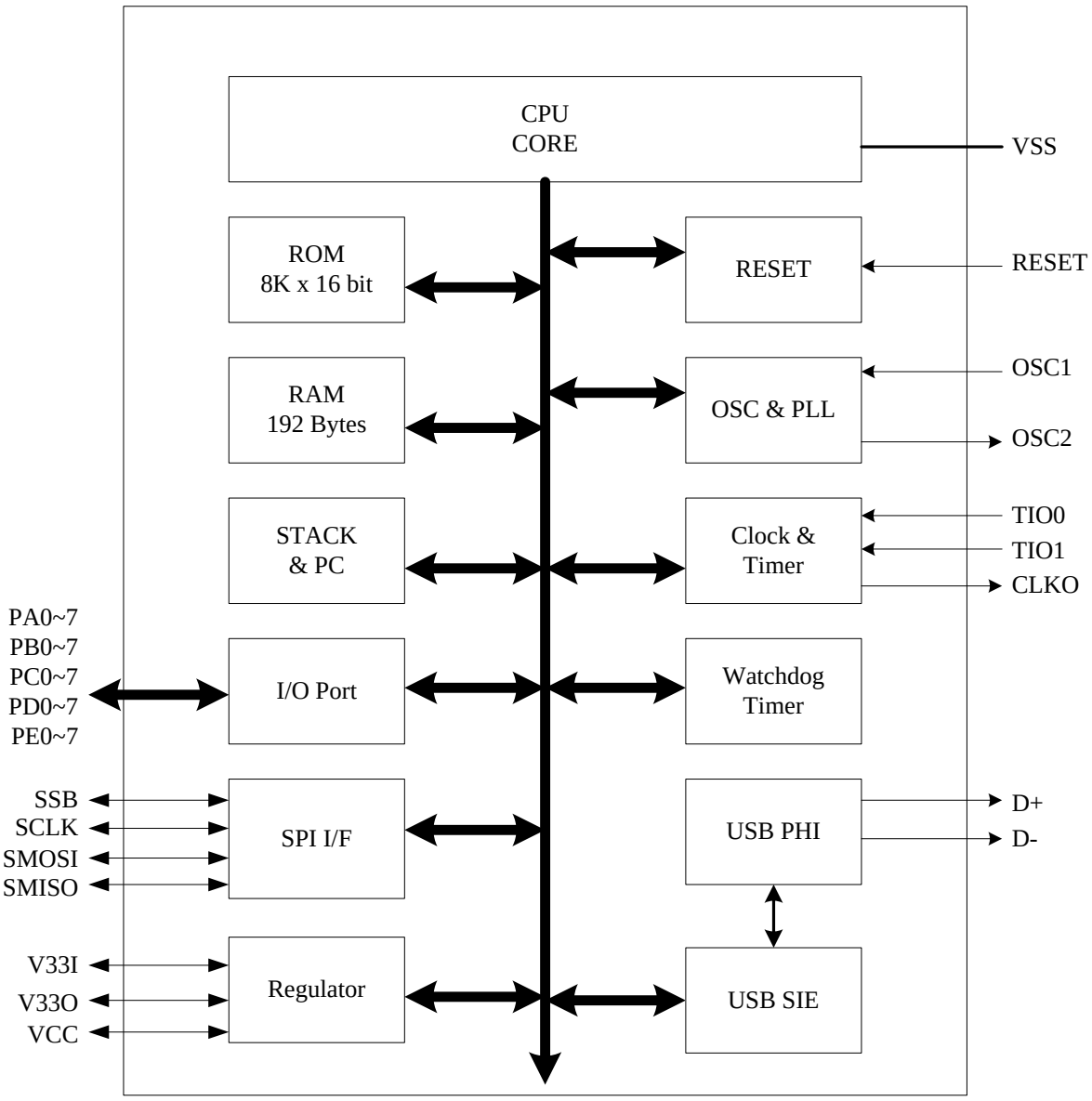
Fig. QFP48 Package

4. PIN DESCRIPTION

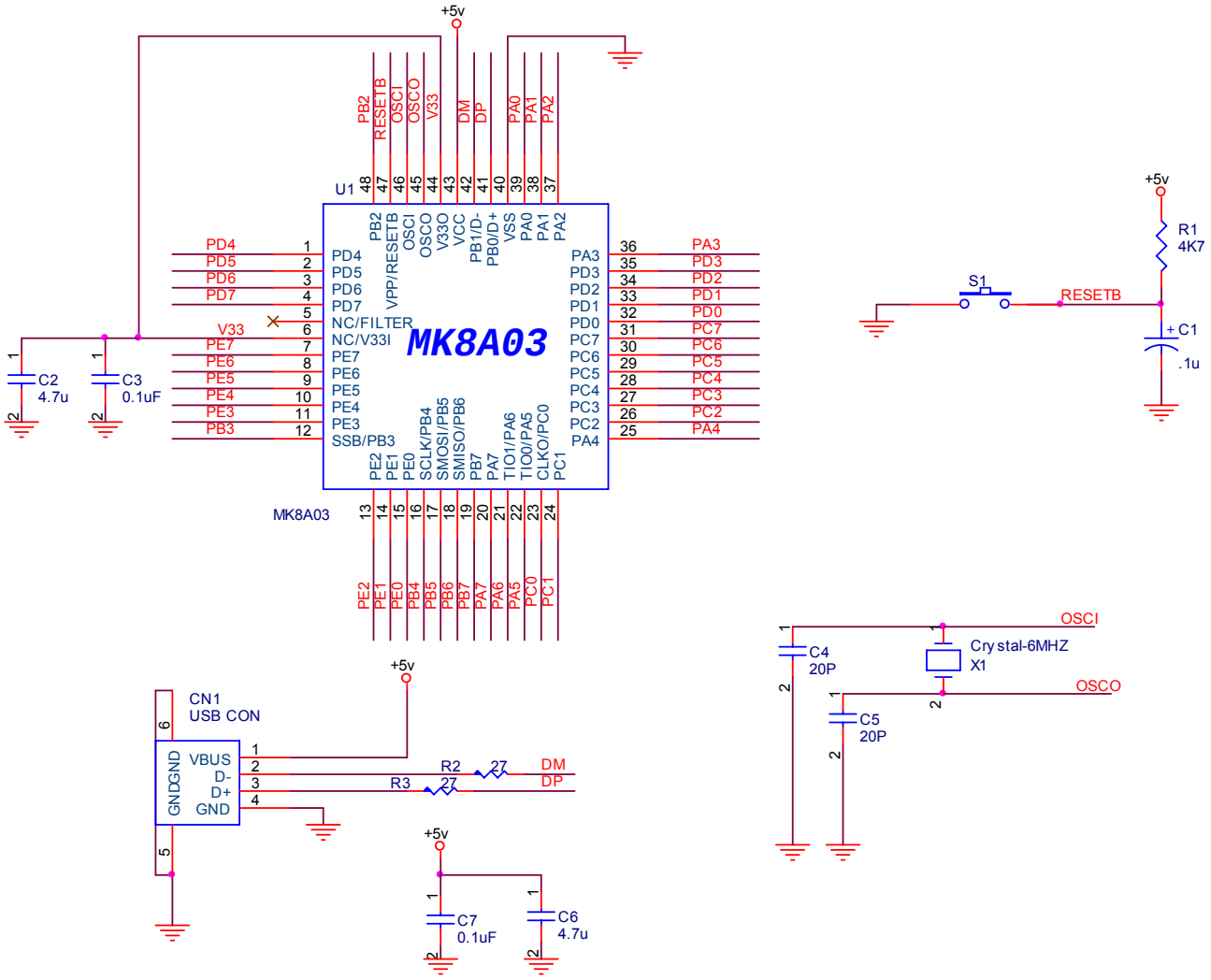
Pin	I/O	Description
VCC	P	Power(5V)
VSS	P	Ground
RESETB		RESET or Input IO
NC		No connect
OSCI OSCO	I O	Crystal in (3MHz/6MHz) Crystal out
V33O		3.3V regulator output
V33I		3.3V input
PA[0]	I/O	GPIO PORTA[0]
PA[1]	I/O	GPIO PORTA[1]
PA[2]	I/O	GPIO PORTA[2]
PA[3]	I/O	GPIO PORTA[3]
PA[4]	I/O	GPIO PORTA[4]
PA[5]/TIO0	I/O	1.GPIO PORTA[5] 2.Timer capture input TIO0
PA[6]/TIO1	I/O	1.GPIO PORTA[6] 2.Timer capture input TIO1
PA[7]	I/O	GPIO PORTA[7]
PB[0]/D+	I/O	1.GPIO PORTB[0] 2.USB D+ pin
PB[1]/D-	I/O	1.GPIO PORTB[1] 2.USB D- pin
PB[2]	I/O	GPIO PORTB[2]
PB[3]/SSB	I/O	1.GPIO PORTB[3] 2.SSB for SPI bus
PB[4]/SCLK	I/O	1.GPIO PORTB[4] 2.SCLK for SPI bus
PB[5]/SMOSI	I/O	1.GPIO PORTB[5] 2.SMOSI for SPI bus
PB[6]/SMISO	I/O	1.GPIO PORTB[6] 2.SMISO for SPI bus

PB[7]	I/O	GPIO PORTB[7]
PC[0]/CLKO	I/O	GPIO PORTC[0] 4Mhz or 8Mhz clock out
PC[7:1]	I/O	GPIO PORTC[7:1]
PD[7:0]	I/O	GPIO PORTD[7:0]
PE[7:0]	I/O	GPIO PORTD[7:0]

5. BLOCK DIAGRAM



6. APPLICATION CIRCUIT



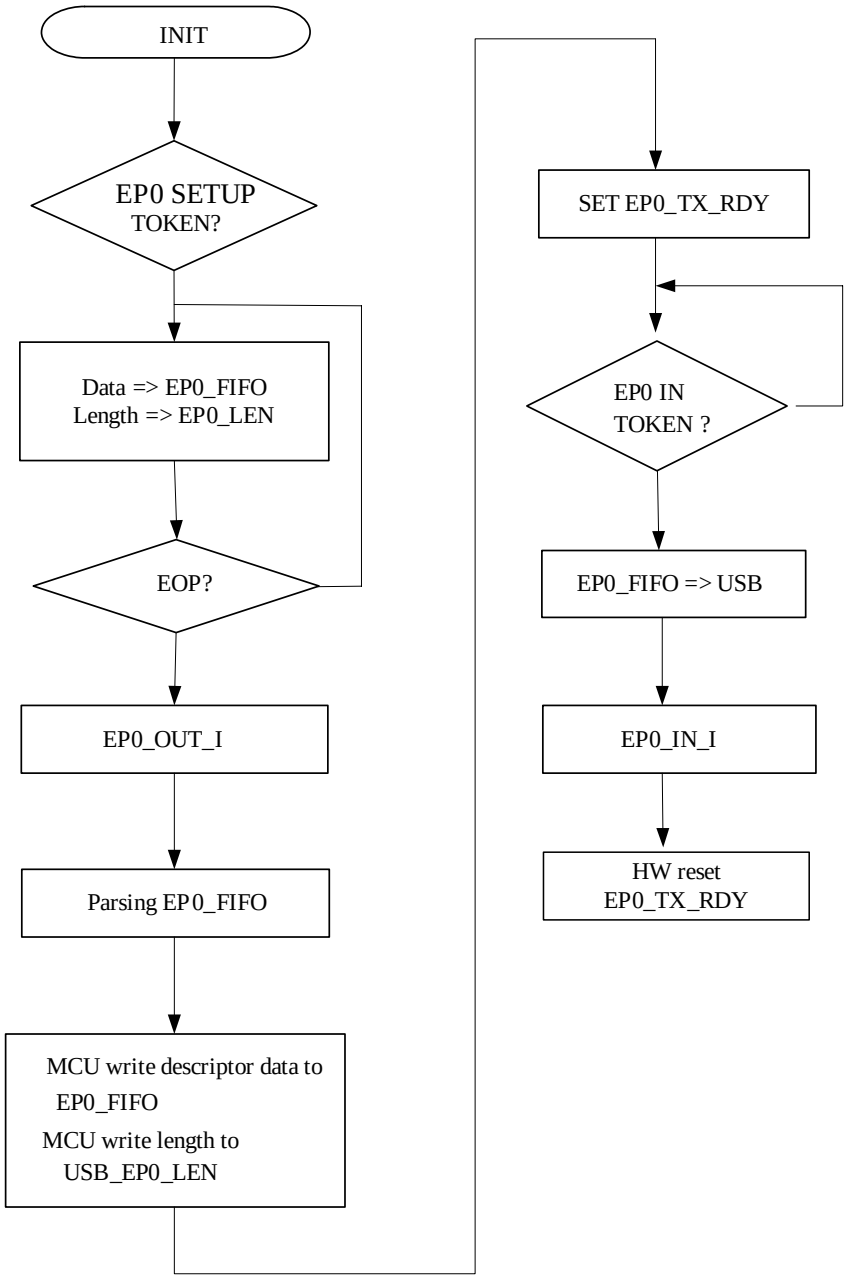
(Preliminary)

7. FUNCTION DESCRIPTION

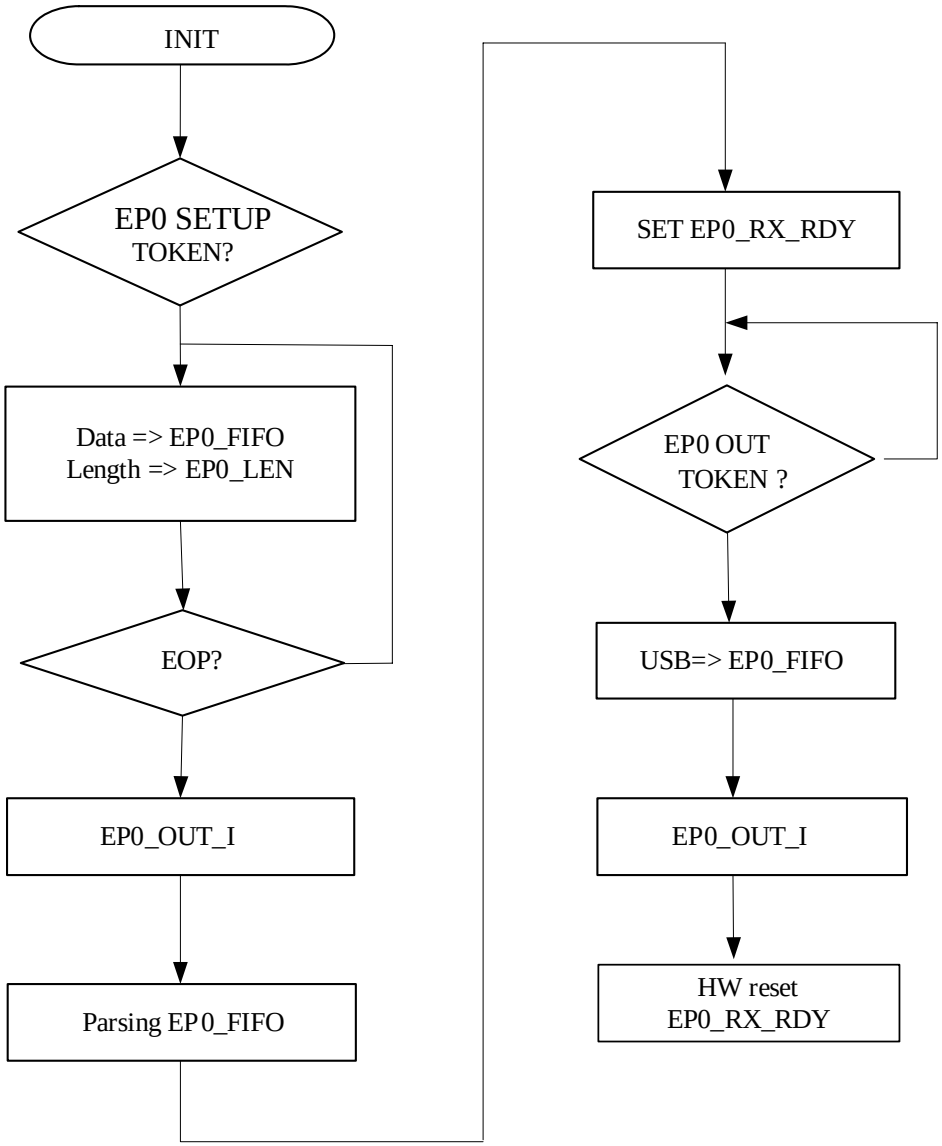
7.1 USB Serial Interface Engine (SIE)

- Fully compliant to USB 2.0 / Full Speed (12 Mbps) specification.
- Complete device configuration
- EP 0 for control IN/OUT. EP 0 have 8 bytes receive buffer and 8 bytes transmit buffer.
- Two for IN/OUT endpoint

CONTROL IN (EP0)



CONTROL OUT (EP0)



7.2 Timer Capture 0/1

- The timer counter contains two 8 bits capture timer and can be cascaded into an one 16 bits capture timer
- Separate registers for rising and falling edge capture
- Prescaler range can be set 12M/1, /8, /16, /32, /64, /128, /256
- Timer capture interrupt

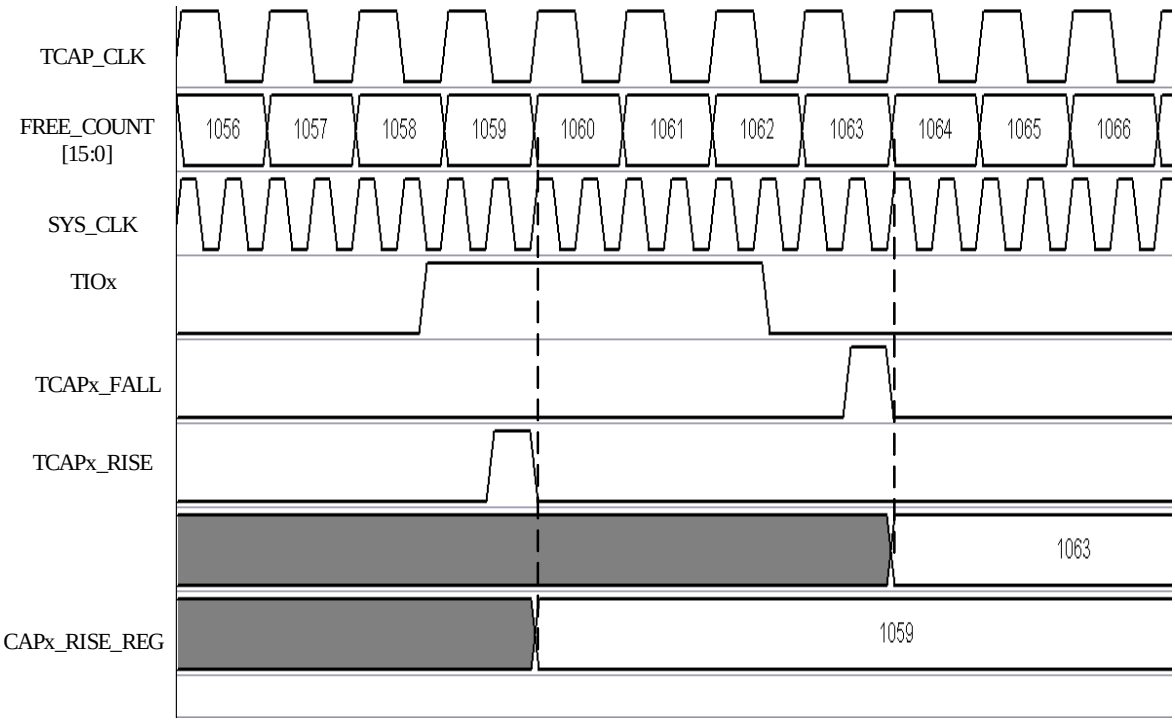
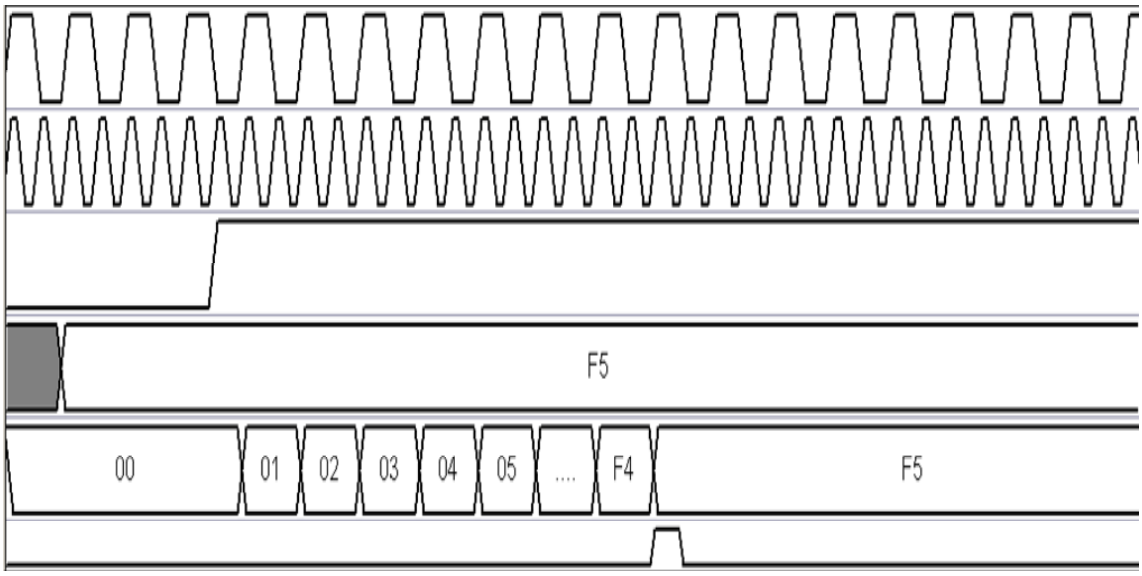


Figure: Timer Capture Timing Diagram

7.3 Timer 0/1

- The timer counter contains two 8 bits programmable one-shot/auto-reload count-up timer and can be cascaded into an one 16 bits timer
- Prescaler range can be set 12M/1, /8, /16, /32, /64, /128, /256
- Timer capture interrupt



SYS_CLK

Figure: Timer Counter Timing Diagram (8Bit, Single Shot)

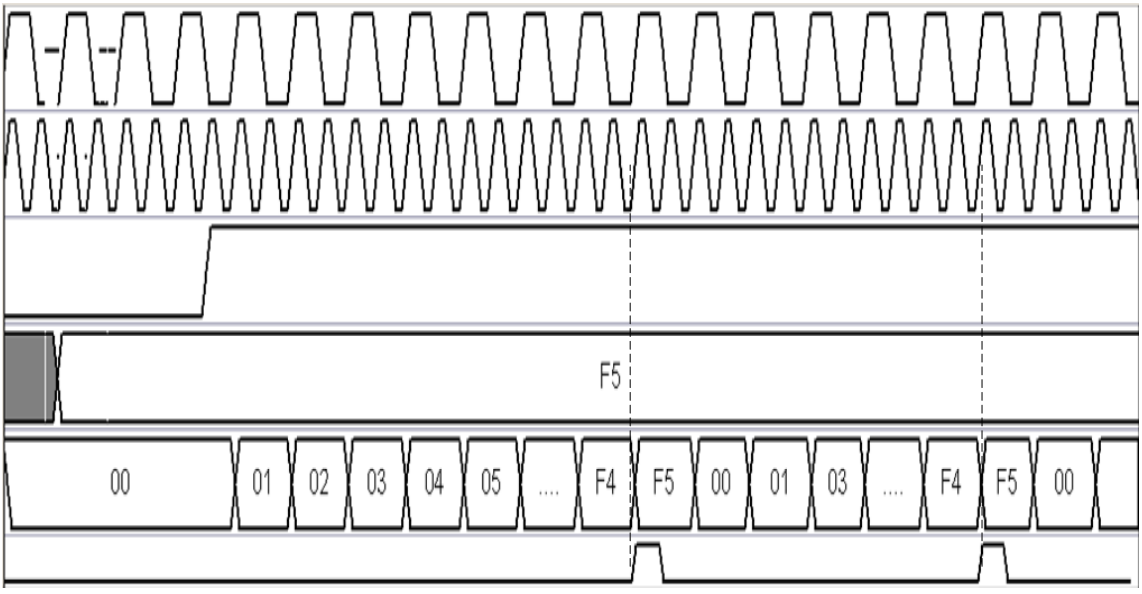


Figure: Timer Counter Timing Diagram (8Bit, Auto-reload)

TMx_I

TM_CLK

7.4 SPI

- Master operation
- Supports half duplex single data line mode for optical sensors
- Supports SPI four mode

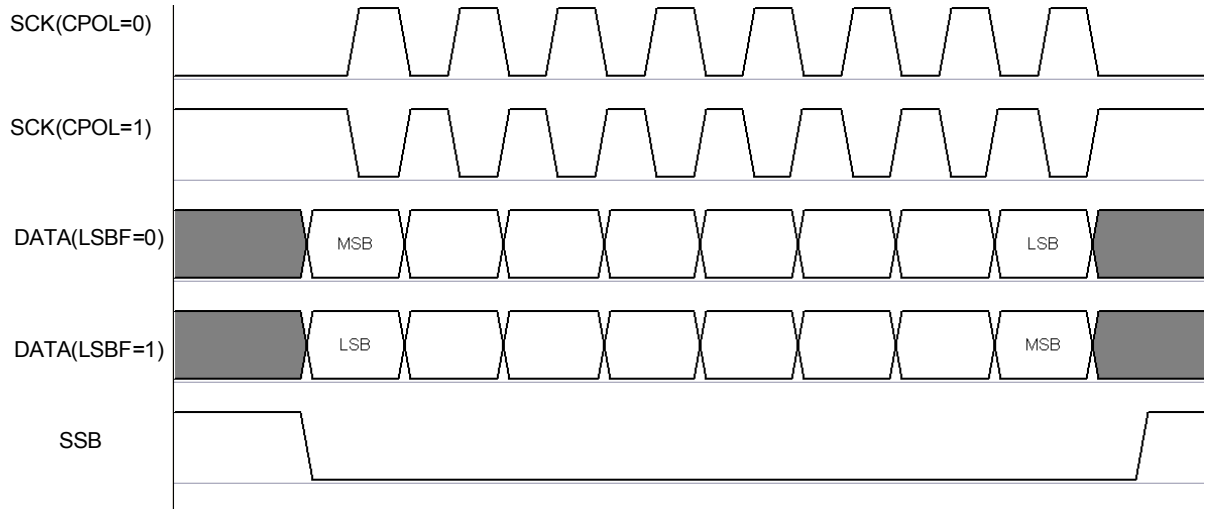


Figure: When SPI CPHA=0

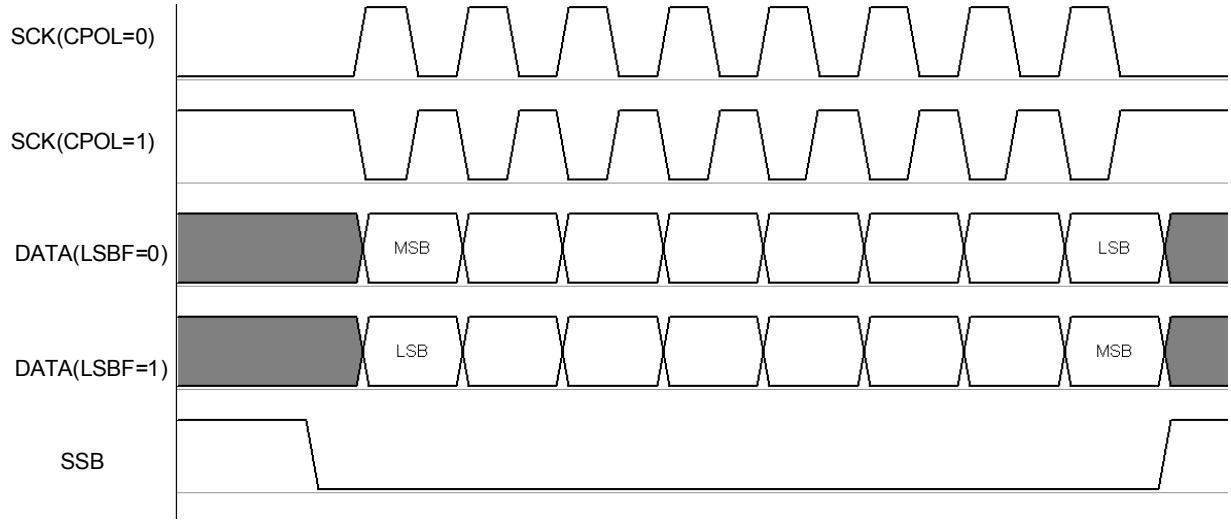
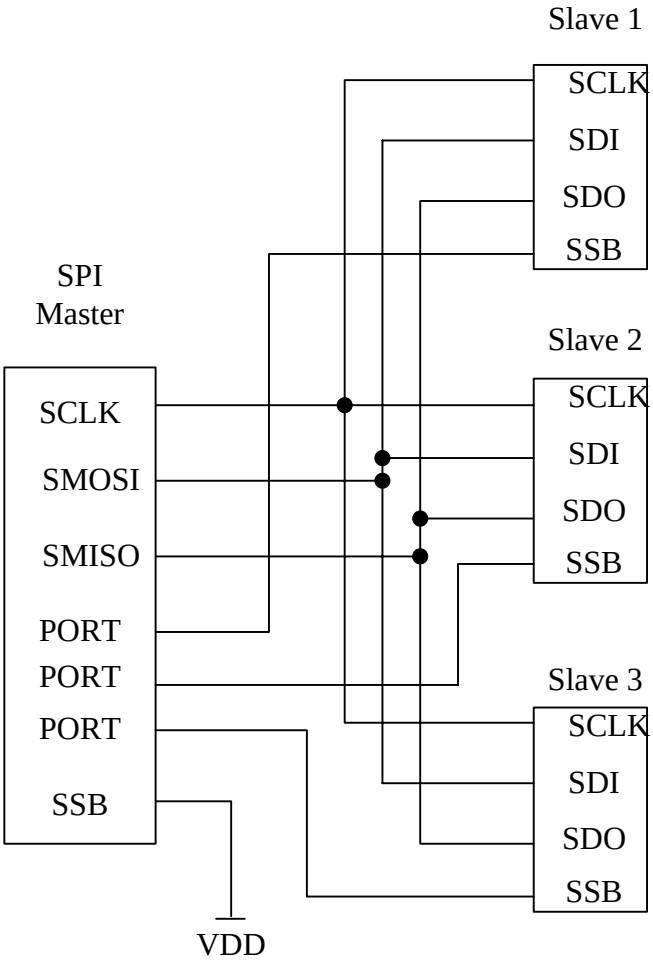
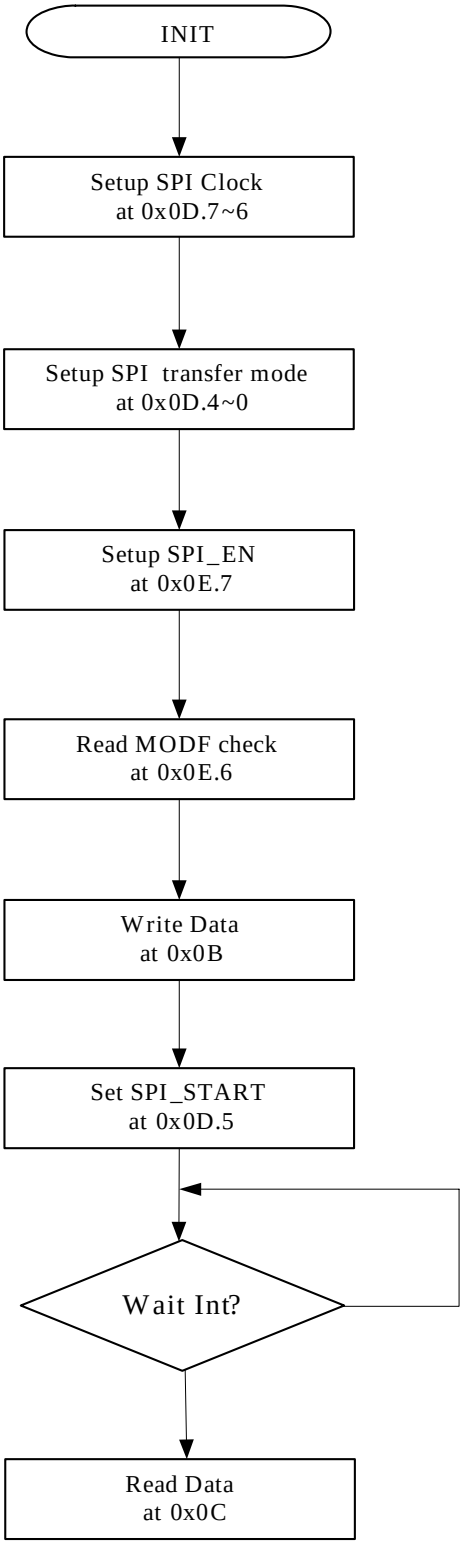


Figure: When SPI CPHA=1

7.4.1 When SPI is Master:



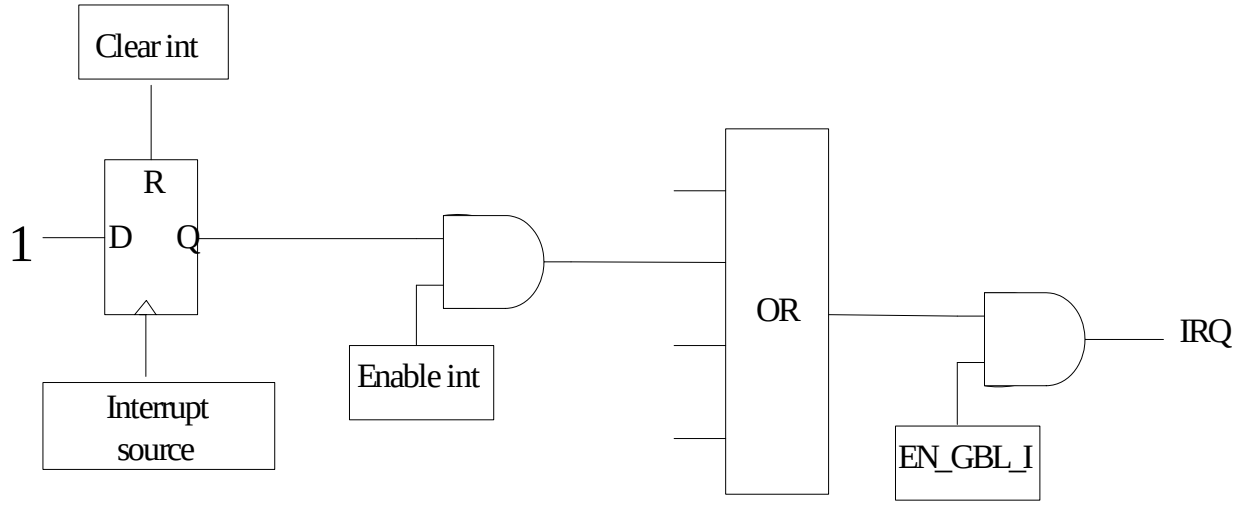
(Preliminary)



7.5 Interrupt

- Interrupt source
  - USB control OUT (EP0) interrupt
  - USB control IN (EP0) interrupt
  - USB (EP1) interrupt
  - USB (EP2) interrupt
  - USB reset interrupt
  - USB resume interrupt
  - USB suspend interrupt
  - SPI interrupt
  - Capture Timer0 interrupt
  - Capture Timer1 interrupt
  - Timer0 interrupt
  - Timer1 interrupt
  - Sleep Timer interrupt
  - GPIO interrupt
- Single interrupt vector
- Clear interrupt before enable it

- Function block



7.6 Power-down mode

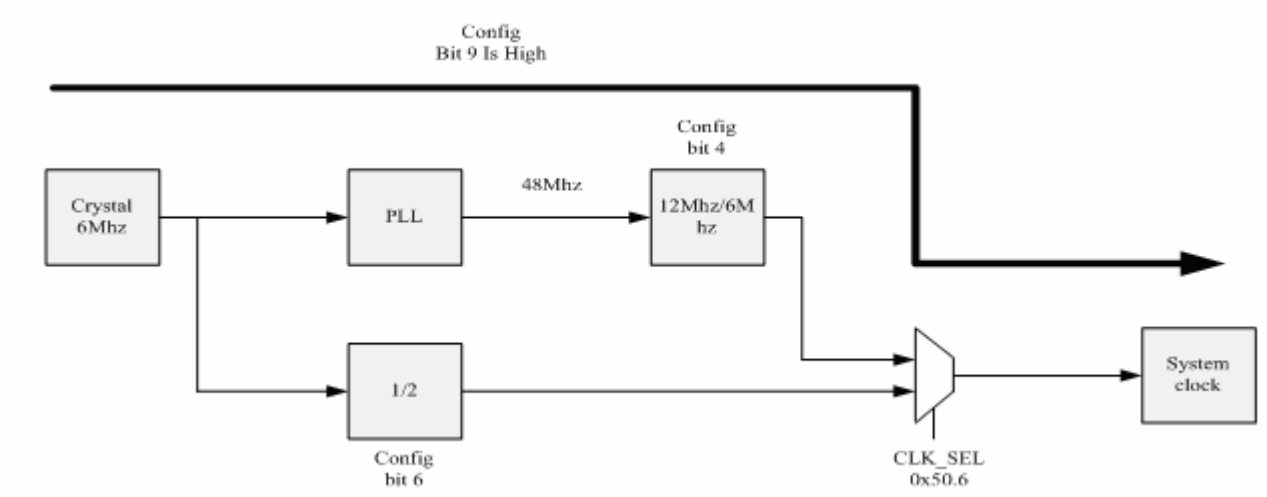
When USB host issue a suspend command or some reason need to save the power, F/W can use SLEEP instruction to enter the power-down mode. In this mode, the crystal oscillator is stopped. The MK8A03BP exits the power down mode using one of the following methods:

- USB bus resume
- USB bus reset
- Wake-up timer
- External enabled GPIO interrupt

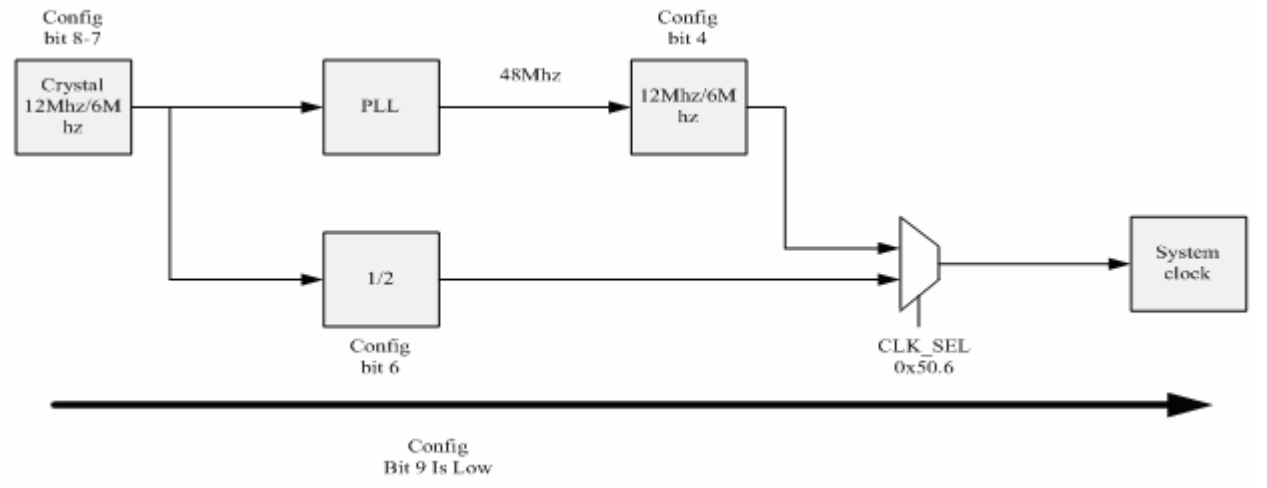
7.7 Configuration system clock environment flowchart

It can be between A version or B version By configuration Bit 9

Default state is high, Crystal always is 6Mhz. System clock is from PLL,

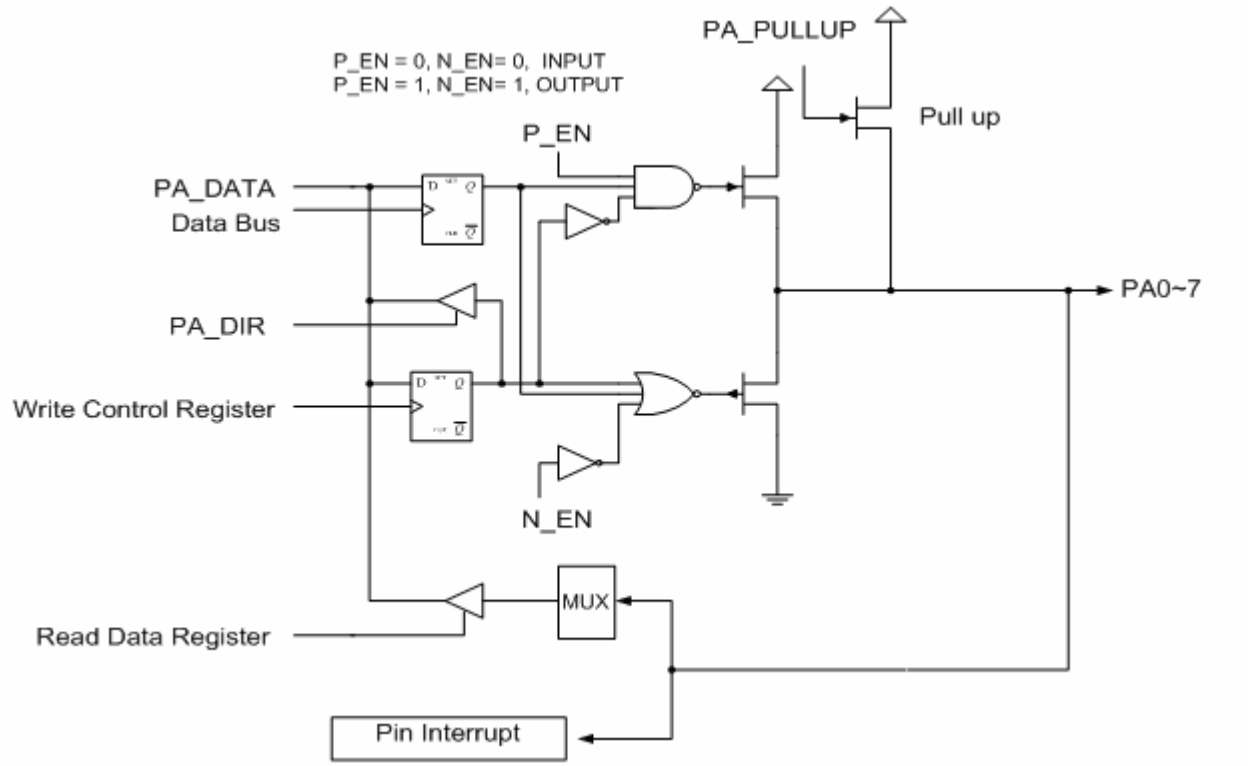


When Configuration Bit 8 is low, system clock is from Crystal

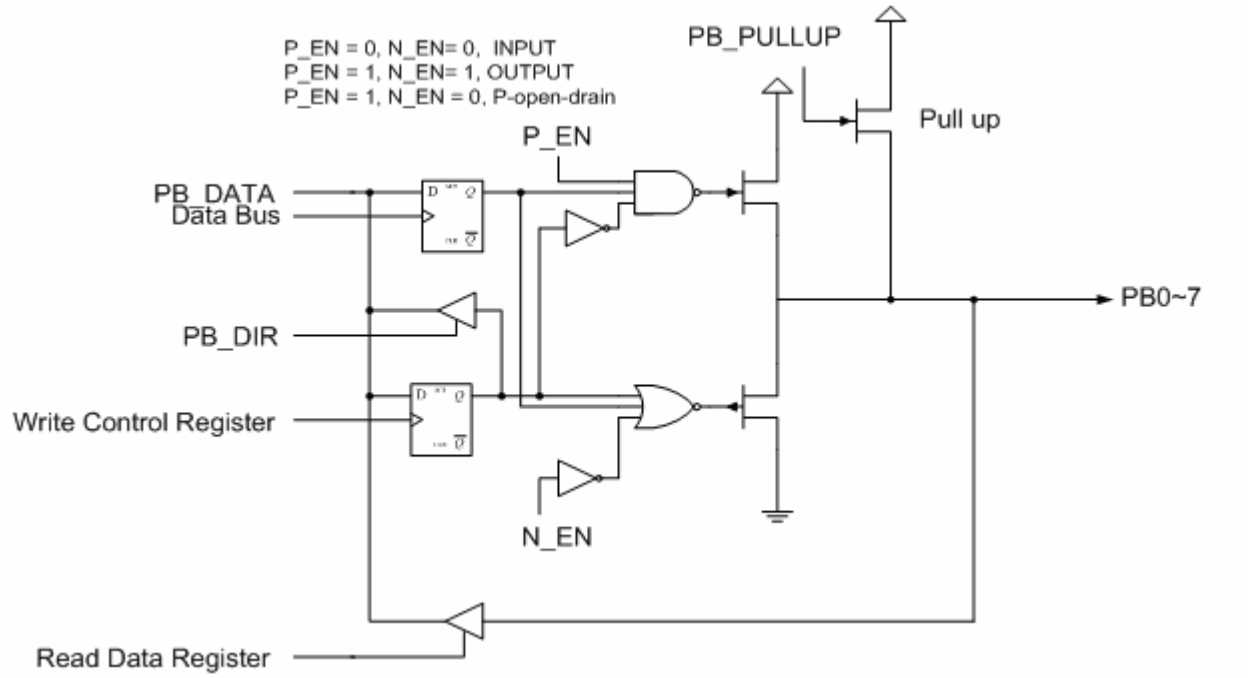


(Preliminary)

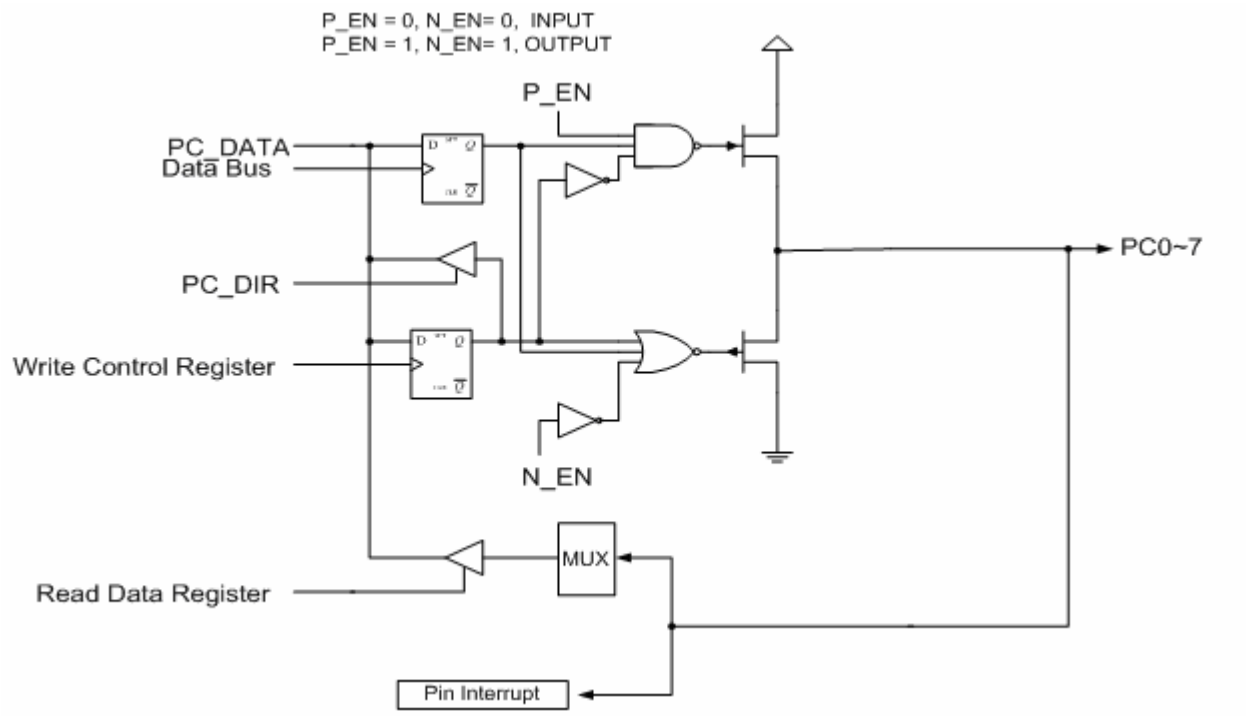
7.8 General-Purpose I/O (GPIO) Ports



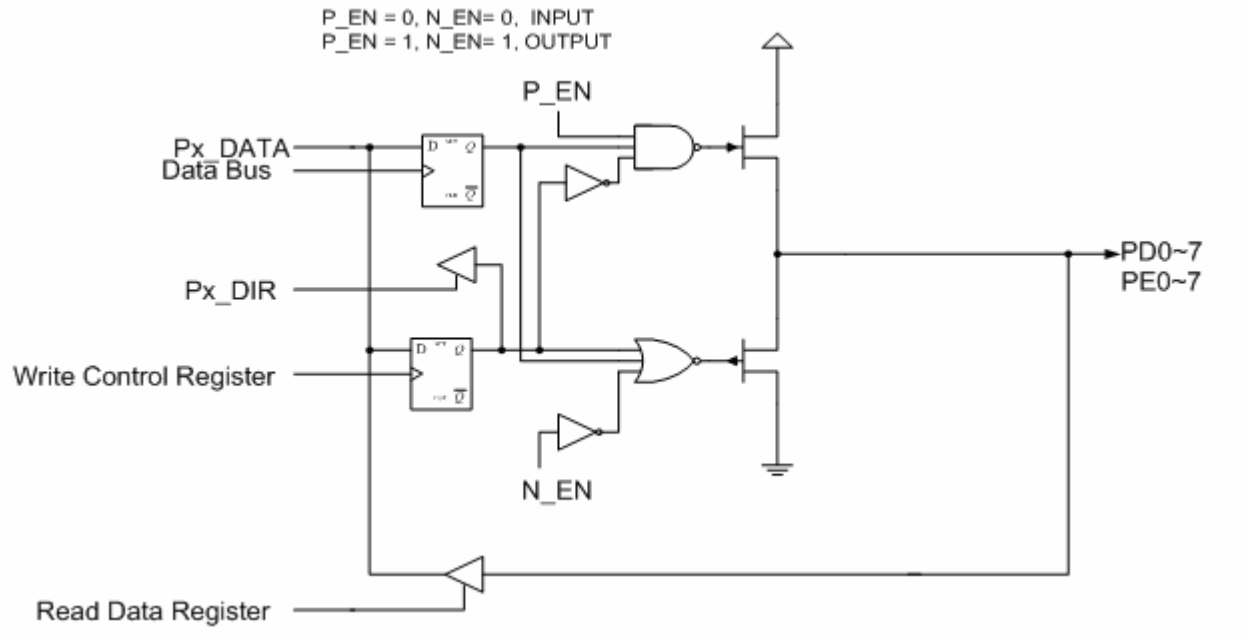
Port A structure



Port B structure



Port C structure



Port D , Port E structure

8. MEMORY MAP

Program address

0x0000	Reset vector
0x0004	Interrupt vector
0x0008 0x1FFF	8K program ROM

Register/SRAM address

0x00 0x5F	Register		
0x60 0x7F	32 bytes SRAM		
0x80 0x9F	128 bytes SRAM bank = 00	32 bytes SRAM bank = 01	
    0xFF			

(Preliminary)

9. REGISTER

Register name	Addr	R/W	Rst	Description
CONFIG	0x3FFF	R	FF	Bit9: B_c_ver 0: MK8A03BPc's PLL 1: Crystal always is 6Mhz MK8A03BPb's PLL Bit8~7: Crystal_for_pll ~~00: crystal 12Mhz for PLL~~ 01: crystal 3Mhz for PLL 11: crystal 6Mhz for PLL Bit6: Crystal to system clock 0: system clock for crystal /2 1: system clock for crystal Bit5: fix_4m8m 0: 0x4F.3~2 fix 4/8M output 1: 0x4F.2 is regulator enable control 0x4f.3 is PLL enable control Bit4: System clock 0 6Mhz 1 12Mhz Bit3: Low Reset 0 disable reset 1 2v reset Bit1: 0: code protect 1: no code protect Bit0: 0: RESET pin is a normal function input pin 1: RESET pin is an external reset pin
INDF	0x00	R/W	00	Addressing this location uses contents of FSR to address data memory
PCL	0x01	R/W	00	Program counter (PC) Low
PCH	0x02	R/W	00	Program counter (PC) High

STATUS	0x03	R/W	18	Bit6:5 RB1:RB0 , SRAM bank select 00: SRAM bank 00 01: SRAM block 01 1x: don't use
		R		Bit4: TO , Time-out; Watchdog overflow 1:After power-up or by the CLRWDT, or SLEEP 0:WDT timeout occurred
		R		Bit3: PD ,Power-down bit 1: after supply have been turned on or by The CLRWDT 0: executing SLEEP instruction
		R/W		Bit2: ALUZ , Zero bit 1: The result of an arithmetic or logic operation is zero 0: The result of an arithmetic or logic operation isn't zero
		R/W		Bit1: ALUDC , Digit carry/borrow bit 1: A carry-out from the 4th low order bit of the result occurred 0: No carry-out from the 4th low order bit of the result
		R/W		Bit0: ALUC , Carry/Borrow bit 1: A carry-out from the most significant bit of the result occurred 0: No carry-out from the most significant bit of The result occurred
FSR	0x04	R/W		Indirect Data Memory Address Pointer
Interrupt				
USB_INT (0x05)				
EN_EP0_IN_I	0x05.7	R/W	0	Enable EP0 IN interrupt. (control transfers) 0: disable 1: enable
EN_EP0_OUT_I	0x05.6	R/W	0	Enable EP0 OUT interrupt. (control transfers) 0: disable 1: enable

EN_EP1 _I	0x05.5	R/W	0	Enable EP1 interrupt 0: disable 1: enable
EN_EP2 _I	0x05.4	R/W	0	Enable EP2 interrupt 0: disable 1: enable
EN_RST_I	0x05.3	R/W	0	Enable USB bus reset interrupt. 0: disable 1: enable
EN_RSM_I	0x05.2	R/W	0	Enable USB resume interrupt. 0: disable 1: enable
EN_SUSP_I	0x05.1	R/W	0	Enable USB suspend interrupt. 0: disable 1: enable
MISC_INT_EN(0x06)				
EN_SPI_I	0x06.7	R/W	0	Enable SPI interrupt 0: disable 1: enable
EN_TCAP0_I	0x06.6	R/W	0	Enable 8 bits Timer capture 0 interrupt 0: disable 1: enable
EN_TCAP1_I	0x06.5	R/W	0	Enable 8 bits capture or 16 bits Timer capture 1 interrupt 0: disable 1: enable
EN_TM0_I	0x06.4	R/W	0	Enable 8 bits Timer0 interrupt 0: disable 1: enable
EN_TM1_I	0x06.3	R/W	0	Enable 8 bits or 16 bitsTimer1 interrupt 0: disable 1: enable
EN_STM_I	0x06.2	R/W	0	Enable Sleep Timer interrupt 0: disable 1: enable
EN_WAKEUP_I	0x06.1	R/W	0	Enable Wake up interrupt 0: disable

				1: enable
GPIO_INT_EN(0x07)				
EN_GPIO_I	0x07.7	R/W	0	Enable GPIO interrupt Use PA_I_E and PC_I_E to choose which GPIO is selected to be an external interrupt 0: disable 1: enable
EN_GBL_I	0x07.6	R/W	0	Enable Global interrupt 0: disable 1: enable
USB_INT_FLAG(0x08)				
EP0_IN_I	0x08.7	R/W	0	EP0 IN interrupt flag, write 0 to clear (control transfers)
EP0_OUT_I	0x08.6	R/W	0	EP0 OUT interrupt flag, write 0 to clear (control transfers)
EP1_I	0x08.5	R/W	0	EP1 interrupt flag, write 0 to clear
EP2 _I	0x08.4	R/W	0	EP2 interrupt flag, write 0 to clear
RST_I	0x08.3	R/W	0	USB bus reset interrupt flag, write 0 to clear
RSM_I	0x08.2	R/W	0	USB resume interrupt flag, write 0 to clear
SUSP_I	0x08.1	R/W	0	USB suspend interrupt flag, write 0 to clear
MISC_INT_FLAG(0x09)				
SPI_I	0x09.7	R/W	0	SPI interrupt flag, write 0 to clear
TCAP0_I	0x09.6	R/W	0	8 bits Timer capture 0 interrupt flag, write 0 to clear
TCAP1_I	0x09.5	R/W	0	8 bits or 16 bits Timer capture 1 interrupt flag, write 0 to clear
TM0_I	0x09.4	R/W	0	8 bits Time0 interrupt flag, write 0 to clear
TM1_I	0x09.3	R/W	0	8 bits or 16 bits Timer 1 interrupt flag, write 0 to clear
STM_I	0x09.2	R/W	0	Sleep time interrupt flag, write 0 to clear
WAKEUP_I	0x09.1	R/W	0	Wake up interrupt flag, write 0 to clear
GPIO_INT_FLAG(0X0A)				
GPIO_I	0x0A.7	R/W	0	GPIO interrupt flag, write 0 to clear
SPI				
SPI_TX_DATA	0x0B	W	0	SPI transmit buffer
SPI_RX_DATA	0x0C	R	0	SPI receive buffer
SPI_CTRL(0x0D)				

SPI_CLK_SEL	0x0D.7 ~ 0x0D.6	R/W	0	SPI clock output select 00: System Clock /2 01: System Clock /16 10: System Clock /32 11: System Clock /128
SPI_START	0x0D.5	R/W	0	SPI transmit/receive start. Write 1 to start the transfer. Reset by H/W after transmission completed. When SPI master mode enable this, SPI slave don't
CPOL	0x0D.4	R/W	0	SPI clock polarity
CPHA	0x0D.3	R/W	0	SPI clock phase
SWAP	0x0D.2	R/W	0	Swap 0: Swap function disabled 1: SPI swap its use of SMOSI and SMISO. This is useful in Single wire SPI communications
LSBF	0x0D.1	R/W	0	LSB first 0: The SPI transmits and receives the MSB first 1: The SPI transmits and receives the LSB first
MSMODE	0x0D.0	R/W	1	Master mode 0: SPI slave mode 1: SPI master mode
SPI_MODE(0x0E)				
SPI_EN	0x0E.7	R/W	0	SPI enable 0: disable SPI 1: enable SPI <Note>: SPI_EN enable, can't setup SPI_CLK_SEL, CPOL, CPHA, SWAP, LSBF, MSMODE
MODF	0x0E.6	R	0	Mode Fault This bit set if SSB is become low while the SPI is configured as a master 0: Mode fault has not occurred 1: Mode fault has occurred
Timer Capture 0/1				
TCAP_CTRL (0x0F)				
TCAP_DIV	0x0F.7 ~	R/W	0	Timer Capture 0 1 timer divider 000: 8 bits timer 0 1 or cascaded 16 bits timer stop

	0x0F.5			001: System Clock 010: System Clock /8 011: System Clock /16 100: System Clock /32 101: System Clock /64 110: System Clock /128 111: System Clock /256
TCAP_PRE	0x0F.4 ~ 0x0f.2	R/W	0	Timer Capture Prescale 000: Capture timer[7:0] 001: Capture timer[8:1] 010: Capture timer[9:2] 011: Capture timer[10:3] 100: Capture timer[11:4] 101: Capture timer[12:5] 110: Capture timer[13:6] 111: Capture timer[14:7]
FIRST_HOLD	0x0F.1	R/W	0	First Hold 0: The time hold last happen 1: The time hold fist happen until the data read
TCAP_CASCADE	0x0F.0	R/W	0	Timer Capture 0 ,1 cascade select 0: individual 8 bits timer 0 and timer 1 1: cascaded 16 bits timer
TCAP_MODE(0x10)				
TCAP0_FALL_EN	0x10.7	R/W	0	Timer Capture 0 Fall enable 0: disable capture 0 falling edge interrupt 1: enable capture 0 falling edge interrupt
TCAP0_RISE_EN	0x10.6	R/W	0	Timer Capture 0 Rise enable 0: disable capture 0 rising edge interrupt 1: enable capture 0 rising edge interrupt
TCAP1_FALL_EN	0x10.5	R/W	0	Timer Capture 1 Fall enable 0: disable capture 1 falling edge interrupt 1: enable capture 1 falling edge interrupt
TCAP1_RISE_EN	0x10.4	R/W	0	Timer Capture 1 Rise enable 0: disable capture 1 rising edge interrupt 1: enable capture 1 rising edge interrupt

TCAP0_FALL_H	0x10.3	R/W	0	Timer Capture 0 Fall happen 0: No happen 1: capture 0 falling edge happen, write 0 to clear
TCAP0_RISE_H	0x10.2	R/W	0	Timer Capture 0 Rise happen 0: No happen 1: capture 0 rising edge happen, write 0 to clear
TCAP1_FALL_H	0x10.1	R/W	0	Timer Capture 1 Fall happen 0: No happen 1: capture 1 falling edge happen, write 0 to clear
TCAP1_RISE_H	0x10.0	R/W	0	Timer Capture 1 Rise happen 0: No happen 1: capture 1 rising edge happen, write 0 to clear
FCNT_L	0x11	R	0	Free running Timer count low byte
FCNT_H	0x12	R	0	Free running Timer count high byte
TCAP0_FALL	0x13	R/W	0	Timer capture 0 when the last falling edge happen
TCAP0_RISE	0x14	R/W	0	Timer capture 0 when the last rising edge happen
TCAP1_FALL	0x15	R/W	0	Timer capture 1 when the last falling edge happen
TCAP1_RISE	0x16	R/W	0	Timer capture 1 when the last rising edge happen
TIMER 0/1				
TM0_CTRL(0x17)				
TM0_DIV	0x17.7 ~ 0x17.5	R/W	0	8 bits Timer 0 or 16 bits cascaded timer divider 000: 8 bits timer 0 or cascaded 16 bits timer stop 001: System Clock 010: System Clock /8 011: System Clock /16 100: System Clock /32 101: System Clock /64 110: System Clock /128 111: System Clock /256
TM0_AUTO	0x17.4	R/W	0	8 bits timer 0 or 16 bits cascaded timer auto reload select 0: auto-reload 1: single shot
TM0_EN	0x17.3	R/W	0	8 bits timer 0 or 16 bits cascaded timer enable. After divider and auto-reload setting finished, enable this flag to start the timer.

				0: disable 1: enable
TM_CASCADE	0x17.2	R/W	0	Timer 0 ,1 cascade select 0: individual 8 bits timer 0 and timer 1 1: cascaded 16 bits timer
TM1_CTRL(0x18)				
TM1_DIV	0x18.7 ~ 0x18.5	R/W	0	8 bits Timer 1 timer divider 000 : 8 bits timer 0 or cascaded 16 bits timer stop 001 : System Clock 010 : System Clock /8 011 : System Clock /16 100 : System Clock /32 101 : System Clock /64 110 : System Clock /128 111 : System Clock 256
TM1_AUTO	0x18.4	R/W	0	8 bits timer 1 auto reload select 0: auto-reload 1: single shot
TM1_EN	0x18.3	R/W	0	8 bits timer 1 enable. After divider and auto-reload setting finished, enable this flag to start the timer. 0: disable 1: enable
TM0_PREL	0x19	W	FF	Timer 0 pre-load register, don't write "0"
TM0_CNT	0x1A	R/W	0	Timer 0 counter value. Only "0" can be written to clear the counter.
TM1_PREL	0x1B	W	FF	Timer 1 pre-load register, don't write "0"
TM1_CNT	0x1C	R/W	0	Timer 1 counter value. Only "0" can be written to clear the counter.
WATCHDOG				
WDT_CTRL(0x1D)				
WDT_EN	0x1D.7	W	0	Watch dog reset enable 0: disable 1: enable Note: Watchdog don't enable at sleep mode. Enable watchdog at 12Mhz / 6Mhz, then sleep

				Watchdog is ok.
WDT_PRE	0x1D.6 ~ 0x1D.4	W	111	Watch dog timer select 010 : 80 ms 011 : 160 ms 100 : 320 ms 101 : 640 ms 110 : 1.28 sec 111 : 2.56 sec
USB				
EP0_FIFO	0x1E ~ 0x25	R/W	0	EP0 data FIFO (8 bytes) (control transfers) 0x1E = data0, 0x25 = data7
EP1 _FIFO	0x26 ~ 0x2D	R/W	0	EP1 data FIFO (8 bytes) 0x26 = data0 , 0x2D = data 7
EP2 _FIFO	0x2E ~ 0x35	R/W	0	EP2 data FIFO (8 bytes) 0x2E = data0 , 0x35 = data 7
USB_ADR(0x36)				
USB_EN	0x36.7	R/W	0	USB enable This bit must be enable by firmware before SIE will respond to USB host. When this bit is cleared, the USB transceiver enters power-down state. User's firmware should clear this Bit to entering sleep mode to Save power 0: Disable USB into power-down 1: Enable USB and into normal operation mode
USB_ADR	0x36.6 ~ 0x36.0	R/W	0	USB device address, cleared while chip reset or USB bus reset
USB_CTRL_STATUS(0x37)				
TOKEN	0x37.7 ~ 0x37.6	R	0	EP0 received token 00: SETUP token 01: IN token

				10: OUT token
DATA_PKT	0x37.5	R	0	USB received DATA packet 0: DATA0 packet 1: DATA1 packet
DATA_ERR	0x37.4	R	0	USB received data error. 0: no error 1: error
EP1_DIR	0x37.3	R/W	0	EP1 Data Direction 0: IN endpoint 1: OUT endpoint
EP2_DIR	0x37.2	R/W	0	EP2 Data Direction 0: IN endpoint 1: OUT endpoint
SUSPND	0x37.1	R/W	0	USB device suspend mode enable write “1” or “0” to enable or disable the suspend mode
RESUME	0x37.0	R/W	0	Force USB device send RESUME signal to USB host while in suspend mode. F/W should set this flag for 10~15ms to send RESUME signal.
EP0_LEN(0x38)				
EP0_LEN	0x38.7 ~ 0x38.4	R/W	0	EP0 data length. 0 will send an zero length data packet. (control transfers) For IN transaction, firmware write the number of bytes For SETUP or OUT transaction, hardware set receive the Number (0~8) of bytes
EP1&2_LEN(0x39)				
EP1_LEN	0x39.7 ~ 0x39.4	R/W	0	EP1 data length. 0 will send an zero length data packet. (interrupt transfers) For IN transaction, firmware write the number of bytes For OUT transaction, hardware set receive the Number(0~8) of bytes When EP1_DIR is 0, EP1_LEN only read EP1_DIR is 1, EP1_LEN only write
EP2_LEN	0x39.3 ~ 0x39.0	R/W	0	EP2 data length. 0 will send an zero length data packet. (interrupt transfers) For IN transaction, firmware write the number of bytes

				For OUT transaction, hardware set receive the number (0~8) of bytes When EP2_DIR is 0, EP2_LEN only read EP2_DIR is 1, EP2_LEN only write
USB_RDY_CTRL(0x3A)				
EP0_RX_RDY	0x3A.7	R/W	0	EP0 is ready to receive data. F/W should set this flag every time the device is ready to receive the data and H/W will reset this flag after transmission completed(EP0_OUT_I). If not ready, H/W will respond with NAK command once IN/OUT token received. 0: not ready 1: ready
EP0_TX_RDY	0x3A.6	R/W	0	EP0 is ready to transmit data. F/W should set this flag every time the device is ready to transmit the data and H/W will reset this flag after transmission completed(EP0_IN_I). If not ready, H/W will respond with NAK command once IN/OUT token received. 0: not ready 1: ready
EP1_RDY	0x3A.5	R/W	0	EP1 is ready to transmit data. F/W should set this flag every time the device is ready to transmit the data and H/W will reset this flag after transmission completed(EP1_IN_I). If not ready, H/W will respond with NAK command once IN/OUT token received. 0: not ready 1: ready
EP2_RDY	0x3A.4	R/W	0	EP2 is ready to transmit data. F/W should set this flag every time the device is ready to transmit the data and H/W will reset this flag after transmission completed(EP2_IN_I). If not ready, H/W will respond with NAK command once IN/OUT token received. 0: not ready 1: ready
EP1_CFG	0x3A.3	R/W	0	Indicate the EP1 is a configured end point. After SET_CNFIGURATION request received, F/W should set this flag or reset this flag when received

				SET_CONFIGURATION with a value of 0
EP2_CFG	0x3A.2	R/W	0	Indicate the EP2 is a configured end point. After SET_CNFIGURATION request received, F/W should set this flag or reset this flag when received SET_CONFIGURATION with a value of 0
USB_PULLUP_EN	0x3A.1	R/W	0	USB pull-up enable 0: disable the pull-up resistor on D+ 1: Enable the pull-up resistor on D+
USB_RESPOND(0x3B)				
EP0_STALL	0x3B.7	R/W	0	EP0 stalled. This is used for received an unknown command from host or unable to transmit or receive data 0: not stall 1: stall
EP1_STALL	0x3B.6	R/W	0	EP1 stalled. This is used for received an unknown command from host or unable to transmit or receive data 0: not stall 1: stall
EP2_STALL	0x3B.5	R/W	0	EP2 stalled. This is used for received an unknown command from host or unable to transmit or receive data 0: not stall 1: stall
EP0_TGL	0x3B.4	R/W	0	Read this flag will read out the received toggle bit of EP0. Write this flag to select DATA0 or DATA1 packet to transmit via endpoint 0 Note: don't use instruction BS & BC setup this bit
EP1_TGL	0x3B.3	R/W	0	Read this flag will read out the received toggle bit of EP1. Write this flag to select DATA0 or DATA1 packet to transmit via endpoint 0 Note: don't use instruction BS & BC setup this bit
EP2_TGL	0x3B.2	R/W	0	Read this flag will read out the received toggle bit of EP2. Write this flag to select DATA0 or DATA1 packet to transmit via endpoint 0 Note: don't use instruction BS & BC setup this bit

EP1&2_NO(0x3C)				
EP1_NO	0x3C.7 ~ 0x3C.4	R/W	1	EP1 determine endpoint NO. (1~15)
EP2_NO	0x3C.3 ~ 0x3C.0	R/W	2	EP2 determine endpoint NO. (1~15)
GPIO				
PA_DATA	0x3D	R/W	0	Port A [7:0] data read/write When using BS, BC instruction at output direct It read PORTA_DATA, then set bit
PA_DIR	0x3E	R/W	FF	Port A [7:0] input/output direction. 0: output 1: input
PA_PULLUP	0x3F	R/W	0	Port A [7:0] pull-up resistor enable 0: disable 1: enable
PB_DATA	0x40	R/W	0	Port B [7:0] data read/write When using BS, BC instruction at output direct It read PORTB_DATA, then set bit
PB_DIR	0x41	R/W	FF	Port B [7:0] input/output direction 0: output 1: input
PB_PULLUP	0x42	R/W	00	Port B [7:0] pull-up resistor enable 0: disable 1: enable Note: PB[0] : when usb_pullup_en = 1, PB_PULLUP[0] always disable PB[1] : when usb_pullup_en = 1, PB_PULLUP[1] always disable
PB_DRAIN	0x43	R/W	0	Port B [7:0] n-open drain enable N-open drain has high priority than PB_DIR Don't enable open drain and PB_PULLUP 0 : disable 1 : enable

PC_DATA	0x45	R/W	0	Port C [7:0] data read/write When using BS, BC instruction at output direct It read PORTC_DATA, then set bit
PC_DIR	0x46	R/W	FF	Port C [7:0] input/output direction 0: output 1: input
PD_DATA	0x47	R/W	0	Port D [7:0] data read/write When using BS, BC instruction at output direct It read PORTD_DATA, then set bit
PD_DIR	0x48	R/W	FF	Port D input/output direction 0: output 1: input
PE_DATA	0x49	R/W	0	Port E [7:0] data read/write When using BS, BC instruction at output direct It read PORTE_DATA, then set bit
PE_DIR	0x4A	R/W	FF	Port E input/output direction 0: output 1: input
PA_I_E	0x4B	R/W	0	Port A [7:0] interrupt enable
PA_I_EDGE	0x4C	R/W	0	Port A [7:0] interrupt edge select 0: rising edge 1: falling edge
PC_I_E	0x4D	R/W	0	Port C [7:0] interrupt enable
PC_I_EDGE	0x4E	R/W	0	Port C [7:0] interrupt edge select 0: rising edge 1: falling edge
GPIO_SEL(0x4F)				
GPIO_SPI_SEL	0x4F.7	R/W	0	GPIO SPI select 0: GPIO 1: SPI
GPIO_TCAP0_SEL	0x4F.6	R/W	0	GPIO TIME capture 0 select 0: GPIO 1: Time0
GPIO_TCAP1_SEL	0x4F.5	R/W	0	GPIO TIME capture 1 select 0: GPIO

				1: Time1
RST_PIN	0x4F.4	R	1	When RESET pin is configured as normal function input pin, Read it from this flag.
FIX_4M8M_SEL	0x4F.3 ~ 0x4F.2	R/W	0	When CONFIG_L Bit6 is 0 PortC[0] GPIO or fixed 4/8 Mhz output select 00 : GPIO 01 : fixed 4 Mhz output 1X : fixed 8 Mhz output When CONFIG_L Bit9 is low 0x4F.2 : 0: regulator enable 1: regulator disable 0x4F.3 : 0: PLL disable 1: PLL enable
GPIO_USB_SEL	0x4F.1	R/W	0	GPIO USB IO select 0: GPIO 1: USB, when set USB , GPIO always input
MISC				
MISC(0x50)				
WAKE_UP	0x50.7	W	0	When 48Mhz is stop, generate interrupt to wake up Only “0” can be written to generate Please use movla 0x00 movam MISC(0x50)
CLK_SEL	0x50.6	R/W	0	Clock select Configuration bit9 is high 0: from PLL clock 1: from Crystal clock Configuration bit9 is low 0: from crystal clock 1: from PLL clock When 0x4F.3 PLL enable, switch to PLL 12Mhz. Firmware wait for PLL stable, then switch to PLL 12Mhz
BANK_SET(0x51)				
TAB_BANK	0x51.7	W	0	Table read instruction bank select.

	~ 0x51.2			When use TABRDH/TABRDL instruction to read the data from ROM, assign these 6 bits flags first to address the corresponding address of ROM code 000000: 0x0000 ~ 0x00FF 000001: 0x0100 ~ 0x01FF 000010: 0x0200 ~ 0x02FF ~ 111101: 0x3D00 ~ 0x3DFF 111110: 0x3E00 ~ 0x3EFF 111111: 0x3F00 ~ 0x3FFF

10. INSTRUCTION TABLE

JUMP INSTRUCTION				
LCALL I	Call subroutine. However, LCALL can addressing 16K address	2	None	01ii iiii iiii iiii
LGOTO I	Go branch to any address	2	None	00ii iiii iiii iiii
LOGIC				
AND M, a	(M) · (acc) → (acc)	1	Z	1010 1000 MMMM MMMM
AND M, m	(M) · (acc) → (M)	1	Z	1010 1001 MMMM MMMM
ANDLA I	Immediate · (acc) → (acc)	1	Z	1111 1000 iiii iiii
COM M, a	~(M) → (acc)	1	Z	1010 0100 MMMM MMMM
COM M, m	~(M) → (M)	1	Z	1010 0101 MMMM MMMM
IOR M, a	(M) or (acc) → (acc)	1	Z	1011 1110 MMMM MMMM
IOR M, m	(M) or (acc) → (M)	1	Z	1011 1111 MMMM MMMM
IORLA I	Immediate or (acc) → (acc)	1	Z	1111 0010 iiii iiii
RL M, a	Rotate left from m to acc m[6:0]→acc[7:1] m[7]→ acc[0]	1	None	1110 0000 MMMM MMMM
RL M, m	Rotate left from m to itself m[6:0]→m[7:1] m[7]→ m[0]	1	None	1110 0001 MMMM MMMM
RLC M, a	Rotate left from m to acc m[7]→c m[6:0]→acc[7:1] c→acc[0]	1	C	1110 0010 MMMM MMMM
RLC M, m	Rotate left from m to itself m[7]→c	1	C	1110 0011 MMMM MMMM

	m[6:0]→m[7:1] c→m[0]			
SL0 M, a	Shift left from m to acc m[6:0]→acc[7:1] 0→acc[0]	1	None	1110 0100 MMMM MMMM
SL0 M, m	Rotate left from m to itself m[6:0]→m[7:1] 0→m[0]	1	None	1110 0101 MMMM MMMM
SL1 M, a	Shift left from m to acc m[6:0]→acc[7:1] 1→acc[0]	1	None	1110 0110 MMMM MMMM
SL1 M, m	Rotate left from m to itself m[6:0]→m[7:1] 1→m[0]	1	None	1110 0111 MMMM MMMM
RR M, a	Rotate right from m to acc 0→acc[7] m[7:1]→acc[6:0]	1	None	1110 1000 MMMM MMMM
RR M, m	Rotate right from m to itself M[0]→m[7] m[7:1]→m[6:0]	1	None	1110 1001 MMMM MMMM
RRC M, a	Rotate right from m to acc m[0]→c, c→acc[7] m[7:1]→acc[6:0]	1	C	1110 1010 MMMM MMMM
RRC M, m	Rotate right from m to itself m[0]→c, c→m[7] m[7:1]→m[6:0]	1	C	1110 1011 MMMM MMMM
SR0 M, a	Rotate right from m to acc 0→acc[7] m[7:1]→acc[6:0]	1	None	1110 1100 MMMM MMMM
SR0 M, m	Rotate right from m to itself 0→m[7] m[7:1]→m[6:0]	1	None	1110 1101 MMMM MMMM
SR1 M, a	Rotate right from m to acc	1	None	1110 1110 MMMM MMMM

	1→acc[7] m[7:1]→acc[6:0]			
SR1 M, m	Rotate right from m to itself 1→m[7] m[7:1]→m[6:0]	1	None	1110 1111 MMMM MMMM
SWAP M, a	m[7:4] → acc[3:0] m[3:0] → acc[7:4]	1	None	1011 1100 MMMM MMMM
SWAP M, m	m[7:4] ←→ m[3:0]	1	None	1011 1101 MMMM MMMM
XOR M, a	(M) xor (acc) → (acc)	1	Z	1011 0110 MMMM MMMM
XOR M, m	(M) xor (acc) → (M)	1	Z	1011 0111 MMMM MMMM
XORLA I	Immediate xor (acc) → (acc)	1	Z	1111 1001 iiii iiii
MATHEMATICS				
ADD M, a	(M)+(acc) → (acc)	1	C, DC, Z	1010 1010 MMMM MMMM
ADD M, m	(M)+(acc) → (M)	1	C, DC, Z	1010 1011 MMMM MMMM
ADDC M,a	(M)+(acc) + (carry) → (acc)	1	C, DC, Z	1011 1010 MMMM MMMM
ADDC M,m	(M)+(acc) + (carry) → (M)	1	C, DC, Z	1011 1011 MMMM MMMM
ADDLA I	Immediate + (acc) → (acc)	1	C, DC, Z	1111 1010 MMMM MMMM
BC M, bn	Clear bit n of (M)	1	None	1001 1bbb MMMM MMMM
BS M, bn	Set bit n of (M)	1	None	1001 0bbb MMMM MMMM
CLRA	Clear accumulator	1	Z	1010 0010 0000 0000
CLR M	Clear memory M	1	Z	1010 0011 MMMM MMMM
TABRDL M	Read low byte ROM table to (acc) ROM table address={TB_BNK,index of M }	2	None	1101 1000 MMMM MMMM
TABRDH M	Read high byte ROM table to (acc) ROM table address={TB_BNK,index of M }	2	None	1101 1001 MMMM MMMM
DAA M, a	Decimal Adjust M to ACC If ACC[3:0] > 9 or DC=1 Then ACC[3:0]←ACC[3:0]+6, DC1=~DC else ACC[3:0] ←ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then ACC[7:4]←ACC[7:4]+6+DC1, C=1 else ACC[7:4] ←ACC[7:4]+DC1, C=C	1	C	1101 0110 MMMM MMMM
DAA M, m	Decimal Adjust M to memory If ACC[3:0] > 9 or DC=1 Then M[3:0]←ACC[3:0]+6, DC1=~DC else M[3:0] ←ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then	1	C	1101 0111 MMMM MMMM

	M[7:4]←ACC[7:4]+6+DC1, C=1 else M[7:4] ←ACC[7:4]+DC1, C=C			

DAS M, a	Decimal Adjust M to ACC
	If ACC[3:0] > 9 or DC=0
	Then
	ACC[3:0]←ACC[3:0]-6,
	DC1=~DC
	Else
	ACC[3:0]←ACC[3:0], DC1=1
	If ACC[7:4] –DC1> 9 or C=0
	Then
	ACC[7:4]←ACC[7:4]-6-DC1,
	C=0
	else
	ACC[7:4] ←ACC[7:4]-DC1,
	C=~C
		1	C	1101 1110 MMMM MMMM

DAS M, m	Decimal Adjust M to memory If ACC[3:0] > 9 or DC=0 Then M[3:0]←ACC[3:0]-6, DC1=~DC else M[3:0] ←ACC[3:0], DC1=1 If ACC[7:4]-DC1 > 9 or C=0 Then M[7:4]←ACC[7:4]-6, C=1 else M[7:4] ←ACC[7:4], C=C	1	C	1101 1111 MMMM MMMM
DEC M, a	(M) - 1 → (acc)	1	Z	1010 1100 MMMM MMMM
DEC M, m	(M) - 1 → (M)	1	Z	1010 1101 MMMM MMMM
INC M, a	(M) + 1 → (acc)	1	Z	1011 0000 MMMM MMMM
INC M, m	(M) + 1 → (M)	1	Z	1011 0001 MMMM MMMM
MOVAM m	(acc) → (M)	1	None	1010 0001 MMMM MMMM
MOV M, a	(M) → (acc)	1	Z	1010 0110 MMMM MMMM
MOV M, m	(M) → (M)	1	Z	1010 0111 MMMM MMMM
MOV2 M, a	(M) → (acc)	1	None	1111 0110 MMMM MMMM
MOV2 M, m	(M) → (M)	1	None	1111 0111 MMMM MMMM
MOVLA I	Immediate data → acc	1	None	1111 0000 iiii iiii
SUBLA I	(immediate data)-(Acc)→(Acc)	1	C, DC, Z	1111 0100 iiii iiii
SUB M, m	(M)–(acc) → (M)	1	C, DC, Z	1011 0101 MMMM MMMM
SUB M, a	(M) –(acc) → (acc)	1	C, DC, Z	1011 0100 MMMM MMMM
OTHER OPERATION				
NOP	No operation	1	None	1111 1111 1111 1111
CLRWDT	Clear watch-dog register	1	$\overline{TO}$, $\overline{PD}$	1111 1111 1111 0000
RET	Return (for lcall instruction)	2	None	1111 1111 1111 0001

IRETI	Return and enable INTM(for IRQ)	2	None	1111 1111 1111 0010
IRET	Return (for IRQ)	2	None	1111 1111 1111 0011
SLEEP	Enter sleep (saving) mode	1	$\overline{TO}$, $\overline{PD}$	1111 1111 1111 0100
CONDITION OPERATION				
BTSC M, bn	If (bit n of (M))=0, skip next instruction	1 or 2	None	1000 1bbb MMMM MMMM
BTSS M, bn	If (bit n of (M))=1, skip next instruction	1 or 2	None	1000 0bbb MMMM MMMM
DECSZ M, a	(M) - 1 →(acc), skip if (acc) = 0	1 or 2	None	1010 1110 MMMM MMMM
DECSZ M, m	(M) - 1 → (M), skip if (M) = 0	1 or 2	None	1010 1111 MMMM MMMM
INCSZ M, a	(M) + 1 →(acc), skip if (acc) = 0	1 or 2	None	1011 0010 MMMM MMMM
INCSZ M, m	(M) + 1 → (M), skip if (M) = 0	1 or 2	None	1011 0011 MMMM MMMM
TMSS	If (acc) =0, skip next instruction	1 or 2	None	1011 1000 XXXX XXXX
TMSC M	If (M) = 0, skip next instruction	1 or 2	None	1011 1001 MMMM MMMM

11. Electrical Characteristics

11.1 Absolute Maximum Rating

Symbol	Min	Max	Unit
Temperature on bias	-50	+125	℃
Storage temperature	0	+70	℃
Supply Voltage	-0.3	5.5	V
Input voltage	-0.3	VDD+0.3	V
Output voltage	-0.3	VDD+0.3	V

11.2 DC Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
VDD	Operating Voltage	5V	System clock 12Mhz,don't use USB	3.1	5		V
			System clock 6Mhz,don't use USB	2.6	5		
V_{IH}	Input High Voltage	5V	V33I is 3.3V		1.5		V
		3.3	V33I is 3.3V		1.5		
V_{IL}	Input Low Voltage	5V	V33I is 3.3V		1.2		V
		3.3V	V33I is 3.3V		1.2		
I_{DD}	Operating current	5V	System clock 12Mhz from PLL		30		mA
			System clock 6Mhz from PLL		27		
			PLL off, regulator on System clock from crystal		4.4		
I_{DD1}	Standby current	5V	PLL off, regulator on, only 32k RC		144		uA
			PLL off, regulator off, only 32k RC		3		
		3.3V	PLL off, regulator on, only 32k RC		52		
			PLL off, regulator off, only 32k RC		0.5		
USB_{PUHI}	USB Pull HI Resistor	3.3V	Set PortB0(DP) Pull_Hi		1.5		KΩ
R_{PUHI}	Pull_Hi Pin Resister	5V	Set PortB input pin and Pull_Hi		60		KΩ
		3V	Set PortB input pin and Pull_Hi		100		

3.

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
f_{rc}	RC clock	5V	RC mode +20% -20%		30		Khz
		3.3V	RC mode +20% -20%		29		