AX206

Digital Photo Frame SoC
Product Specification

Rev 1.1.0
Jun 2009

AppoTech Limited

Address	: Unit 705-707, 7/F, IC Development Centre, No. 6, Science Park West Ave. Hong Kong Science Park, Shatin, N.T. Hong Kong
Telephone	: (852) 2607 4090
Fax	: (852) 2607 4096

www.appotech.com

AX206 8-bit RISC Microcontroller

High Performance 8-bit RISC MCU

- DC-48MHz operation, 48MIPS (Max 48MHz)
- Compatible with 8051
- Fetch 1 instruction byte in 1 cycle

Program Memory and Data Memory

- 4K Bytes Mask-ROM program memory
- 3K Bytes internal data SRAM (Also use as IRAM and USB FIFO)

Interrupt Features

- External wakeup/interrupt capabilities on 3 GPIO and USB PHY
- 2-level interrupt priority

Flexible I/O

- 35 GPIO pins in 5 ports
- All GPIO pins can be individually programmable as input or output
- All GPIO pins are internal pull-up selectable
- TTL-level inputs with Schmitt Trigger
- All GPIO pins are 6mA current output driving except 4 GPIO pins are 12mA current output driving

Digital Peripheral Features

- One 8-bit timer: Timer 0
- Two multi-function 16-bit timers: Timer 1 and Timer 2, support PWM mode
- Real-time wake up
- Watchdog Timer with on-chip 16KHz RC Oscillator
- Full-speed USB 2.0 controller and PHY module with 2 endpoints (including endpoint 0). USB FIFO Share with IRAM
- One high-speed SPI, Maximum throughput 24Mb
- One UART
- 16-bit x 16-bit Multiplier for IDCT
- Bit-fetcher for Huffman Decode

Analog Peripheral Features

- External 24MHz Oscillator
- Internal 16KHz RC Oscillator
- Real-Time Clock Oscillator (32.768KHz)
- Frequency doubler
- 8 Channel 10-bit ADC
- Power-on Reset
- BOR Reset
- LDO

Packages

- 48-pin LQFP (7mm x 7mm)
- DIE form

AppoTech Limited

Address	:Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., Hong Kong
Telephone	:(852) 2607 4090
Fax	:(852) 2607 4096

www.appotech.com

Table of Contents

Chapter 1 Product Overview....6
1.1 Description....6
1.2 System Architecture....7
Chapter 2 Pin Information....8
2.1 Pin Assignment....8
2.1.1 LQFP48....8
2.1.2 Pin Description....9
Chapter 3 Memory Mapping and CPU Architecture....11
3.1 Basic Structure....11
3.2 Interrupt Entry Address Mapping....12
3.3 Data Memory – DRAM (DATA)....13
3.4 General Purpose Register (REGISTER)....13
3.5 Extended Data Memory – XRAM (XDATA)....13
3.6 CPU Instruction Set Summary....14
3.6.1 Basic Instruction Set....14
Chapter 4 Special Function Register (SFR)....18
4.1 SFR List....18
4.2 CPU and Interrupt SFR....20
Chapter 5 Interrupt Processing....23
5.1 Interrupt Sources and Vectors....24
5.2 Interrupt Priorities....25
5.3 Interrupt Latency....26
Chapter 6 Clocks and Reset Management....27
6.1 Clock System....27
6.1.1 Clock Source Control....28
6.1.2 Clock Gating Control....29
6.1.3 Oscillator configurations....30
6.1.4 PCB layout recommendation....30
6.1.5 RC Oscillator....30
6.2 Reset System....31
6.2.1 Reset Sequence....31
6.2.2 Master Reset....32
6.2.3 Power On Reset....32
6.2.4 Watchdog Timeout Reset....32
6.3 Power Management....33

6.3.1 Operating in 3.3V or 5V Systems....33
6.3.2 Idle Mode....34
6.3.3 Halt Mode....34
6.3.4 Power Down Mode....34

Chapter 7 Ports....36

7.1 Registers....37
7.2 Interrupt and Wakeup....43
7.3 Operation Guide....43

Chapter 8 Multi-Function Timer....44

8.1 Timer0....44
8.1.1 Timer0 Registers....45
8.1.2 Operation Modes....46
8.1.3 Interrupt....46
8.1.4 Operation Guide....46
8.2 Timer1....48
8.2.1 Timer1 Registers....49
8.2.2 Timer1 Operation Mode....51
8.2.3 Interrupt....51
8.2.4 Operation Guide....51
8.3 Timer2....52
8.3.1 Timer2 Register....53
8.3.2 Operation Mode....55
8.3.3 Interrupt....55
8.3.4 Operation Guide....55
8.4 Watchdog Timer with On-chip 16kHz RC oscillator....56
8.4.1 Watchdog Timer Register....56
8.4.2 Operation Mode....56
8.5 RTCC Timer....57
8.5.1 RTCC Timer Register....57
8.5.2 Operation Mode....58
8.5.3 Operation Guide....58

Chapter 9 Serial Protocol Interface (SPI)....59

9.4 SPI Registers....59
9.5 Baudrate Setting....60
9.6 Operation Mode....60
9.7 Operation Guide....61

Chapter 10 USB Device Controller and PHY....62

10.1 Registers....63
10.2 Operation Mode....64
10.2.1 Registers Read From/Writer to USB Controller....64
10.2.2 Control The USB Module....64

Chapter 11 UART......66
11.1 Registers......67
11.2 Operation Guide......68
11.2.1 UART TX Operation Flow......68
11.2.2 UART RX Operation Flow......68

Chapter 12 16-bit x 16-bit Multiplier (MUL)......70
12.1 Registers......71
12.2 Operation Guide......73

Chapter 13 Bit-Fetcher......74
13.1 Registers......74
13.2 Operation Guide......75

Chapter 14 ADC......76
14.1 ADC Architecture......76
14.2 Registers......78
14.3 Operation Guide......79

Chapter 15 Electrical Characteristics 80
15.1 Absolute Maximum Rating......80
15.2 DC Characteristics......80
15.3 AC Parameters......82

Chapter 16 Package dimensions......83

Appendix I Revision History......84

Appendix II USB Function Register......85

1 Product Overview

1.1 Description

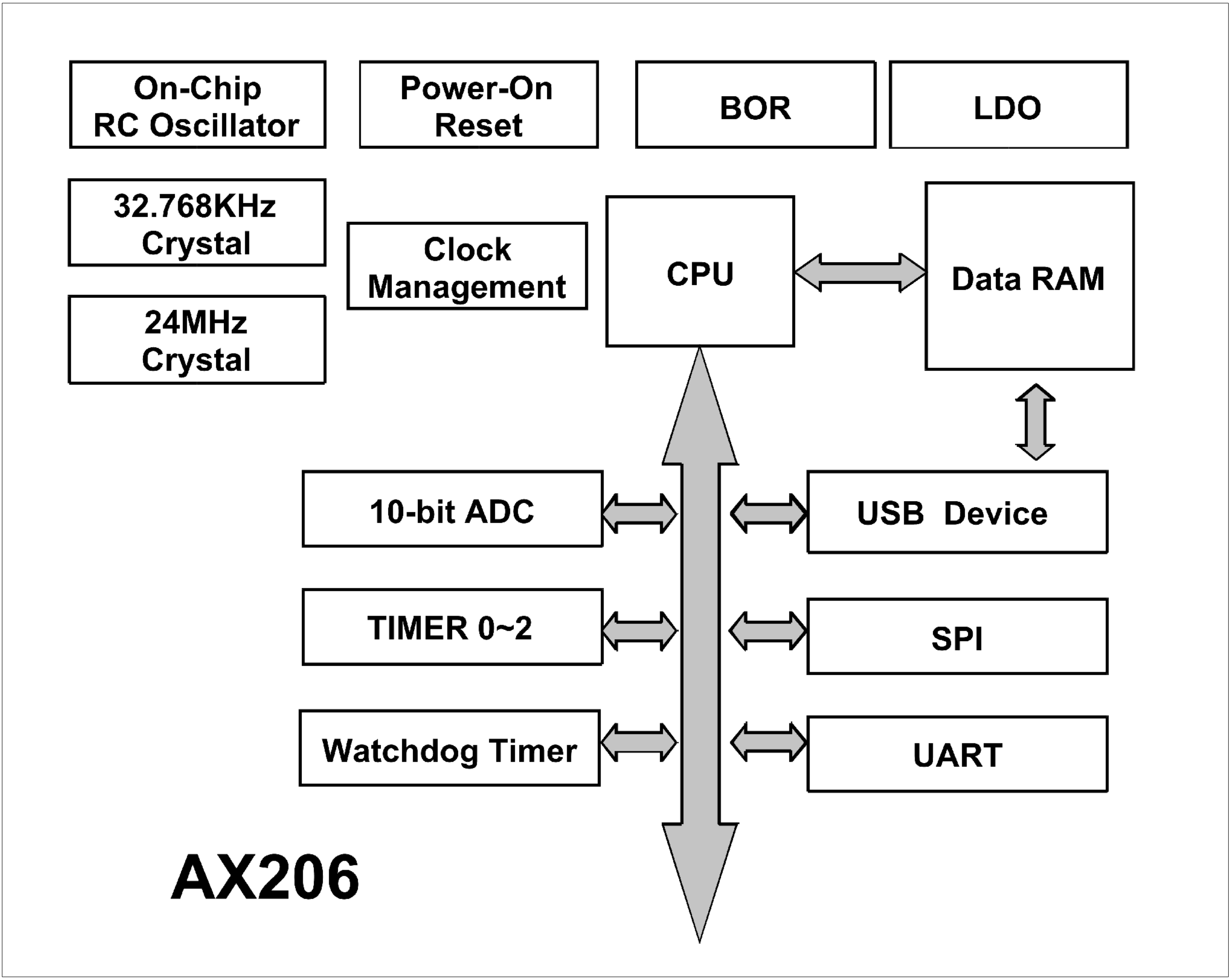
AX206 is an 8051 compatible high performance mixed signal 8-bit microcontroller. It integrates advanced digital and analog peripherals to suit for a variety of applications.

The AX206 has the following features:

- 8-bit RISC CPU compatible with 8051
- Fetch 1 instruction byte in 1 cycle
- CPU can run in RTC (32.768K), 12MHz, 24MHz or 48MHz. (External oscillator is 24MHz)
- 4K bytes Mask ROM Program Memory
- 3K bytes SRAM which can be used as XDATA and Program Memory (IRAM).
- External Wakeup/Interrupt capabilities on 3 GPIO and USB PHY
- 2-level interrupt priority
- 35 GPIO pins in 5 ports (P0, P1, P2, P3, P4)
- All GPIO pins can be individually programmable as input or output
- All GPIO pins are internally pull-up selectable
- ALL GPIO pins are 6mA current output driving except 4 GPIO pins are 12mA current output driving
- Full-Speed USB 2.0 controller and PHY module with 2 endpoints. USB shares part of SRAM
- A high-speed SPI and a high-speed UART
- 16-bit x 16-bit Multiplier for IDCT
- Bit-Fetcher for Huffman Decode
- 8 Channel 10-bit ADC
- Power-on Reset and BOR
- LDO

1.2 System Architecture

Figure 1-1: System Architecture

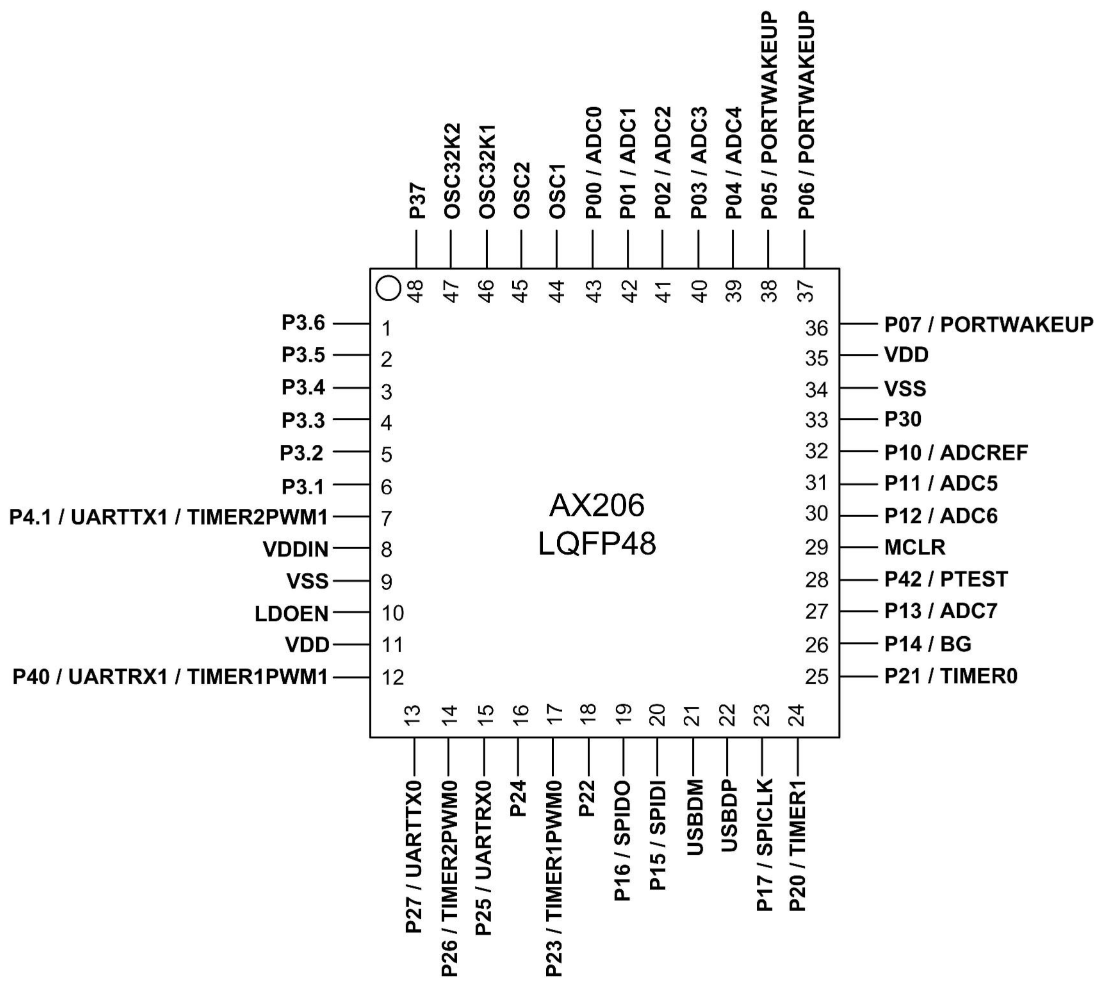


2 Pin Information

2.1 Pin Assignment

2.1.1 LQFP48

Figure 2-1: LQFP48 Pin Assignment



2.1.2 Pin Description

*Table 2-1: **Pin Description***

Pin	Symbol	Direction	Description
1	P36	I/O	Port 3 pin 6
2	P35	I/O	Port 3 pin 5
3	P34	I/O	Port 3 pin 4
4	P33	I/O	Port 3 pin 3
5	P32	I/O	Port 3 pin 2
6	P31	I/O	Port 3 pin 1
7	P41 / UARTTX1 / TIMER2PWM1	I/O O O	Port 4 pin 1 UART TX1 Timer2 PWM1 Output
8	VDDIN	Power	LDO 5.0V Input
9	VSS	Ground	VSS
10	LDOEN	I	LDO Enable
11	VDD	Power	3.3V
12	P40 / UARTRX1 / TIMER1PWM1	I/O I O	Port 4 pin 0 UART RX1 Timer1 PWM1 Output
13	P27 / UARTTX0	I/O O	Port 2 pin 7 UART TX0
14	P26 / TIMER2PWM0	I/O I/O	Port 2 pin 6 Timer2 PWM0 Output
15	P25 / UARTRX0	I/O I	Port 2 pin 5 UART RX0
16	P24	I/O	Port 2 pin 4
17	P23 / TIMER1PWM0	I/O O	Port 2 pin 3 Timer1 PWM0 Output
18	P22	I/O	Port 2 pin 2
19	P16 / SPIDO	I/O O	Port 1 pin 6 SPI DO
20	P15 / SPIDI	I/O I	Port 1 pin 5 SPI DI
21	USBDM	I/O	USB Negative Input/Output
22	USBDP	I/O	USB positive Input/Output
23	P17 / SPICLK	I/O	Port 1 pin 7
		I/O	SPI CLK
24	P20 / TIMER1	I/O I	Port 2 pin 0 Timer1's External Clock Input

25	P21 / TIMER0	I/O I	Port 2 pin 1 Timer0's External Clock Input
26	P14 / BG	I/O O	Port1 pin 4 Bandgap Output
27	P13 / ADC7	I/O I	Port 1 pin 3 ADC Channel 7
28	P42 / PTEST	I/O I	Port 4 pin 2 Test Mode
29	MCLR	I	Master Clear Reset Input
30	P12 / ADC6	I/O I	Port 1 pin 2 ADC Channel 6
31	P11 / ADC5	I/O I	Port 1 pin 1 ADC Channel 5
32	P10 / ADCREF	I/O I	Port 1 pin 0 ADC Voltage Reference Input
33	P30	I/O	Port 3 pin 0
34	VSS	Ground	Negative Power Supply
35	VDD	Power	Positive Power Supply
36	P07 / PORTWAKEUP	I/O I	Port 0 pin 7 Port Wakeup
37	P06 / PORTWAKEUP	I/O I	Port 0 pin 6 Port Wakeup
38	P05 / PORTWAKEUP	I/O I	Port 0 pin 5 Port Wakeup
39	P04 / ADC4	I/O I	Port 0 pin 4 ADC Channel 4
40	P03 / ADC3	I/O I	Port 0 pin 3 ADC Channel 3
41	P02 / ADC2	I/O I	Port 0 pin 2 ADC Channel 2
42	P01 / ADC1	I/O I	Port 0 pin 1 ADC Channel 1
43	P00 / ADC0	I/O I	Port 0 pin 0 ADC Channel 0
44	OSC1	I	Crystal Oscillator Input
45	OSC2	O	Crystal Oscillator Output
46	OSC32K1	I	32.768K Crystal Oscillator Input
47	OSC32K2	O	32.768K Crystal Oscillator Output
48	P37	I/O	Port 3 pin 7

3 Memory Mapping and CPU Architecture

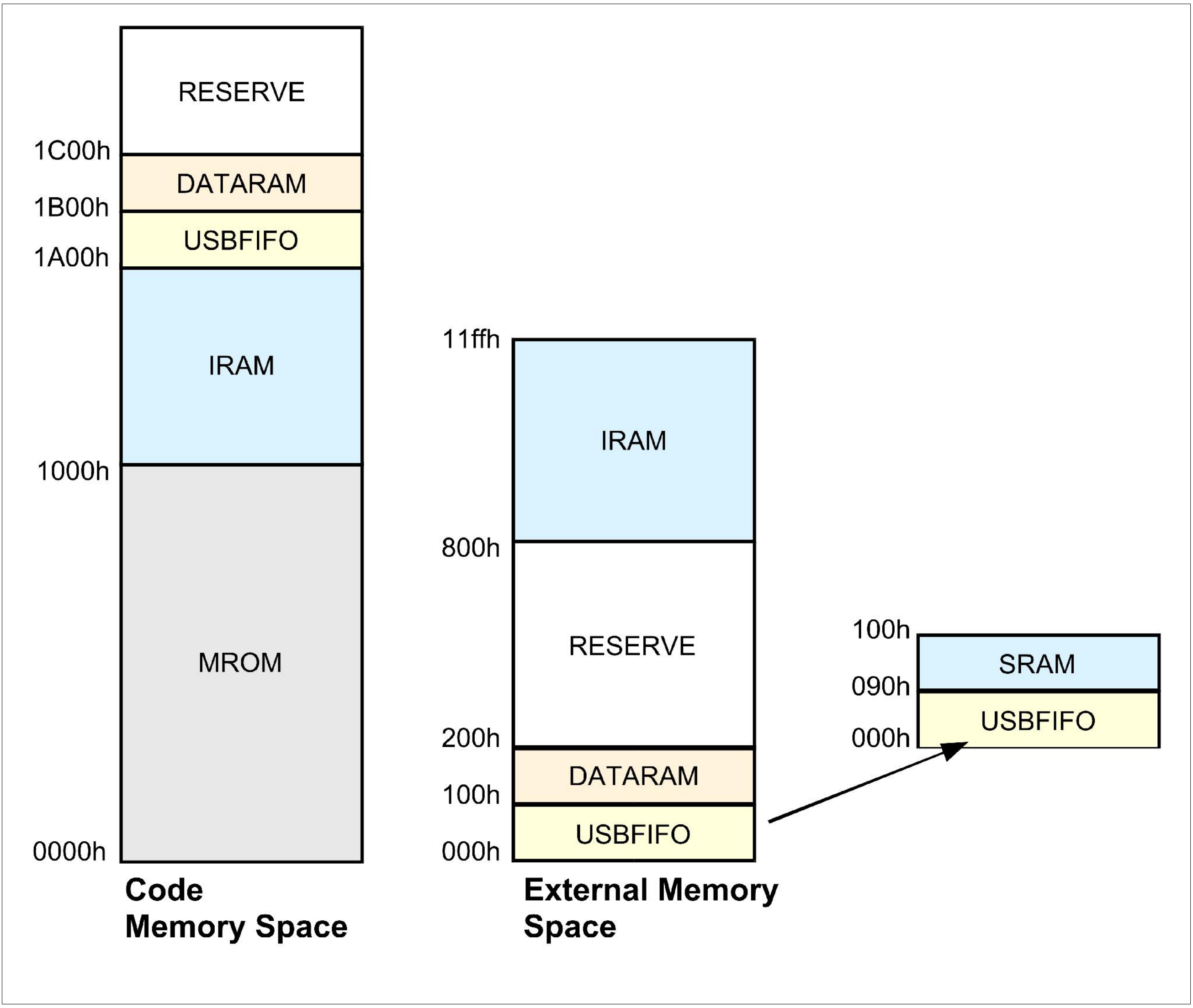
3.1 Basic Structure

AX206 CPU is modified and optimized from the standard 8051/8052 architecture. It supports most of the instructions in standard 8051/8052 except listed below:

DA

AX206 supports 1 Data Pointer Register (DPTR) for accessing the external memory space (XDATA). Also, the memory mapping is different, as shown in Figure 3-1.

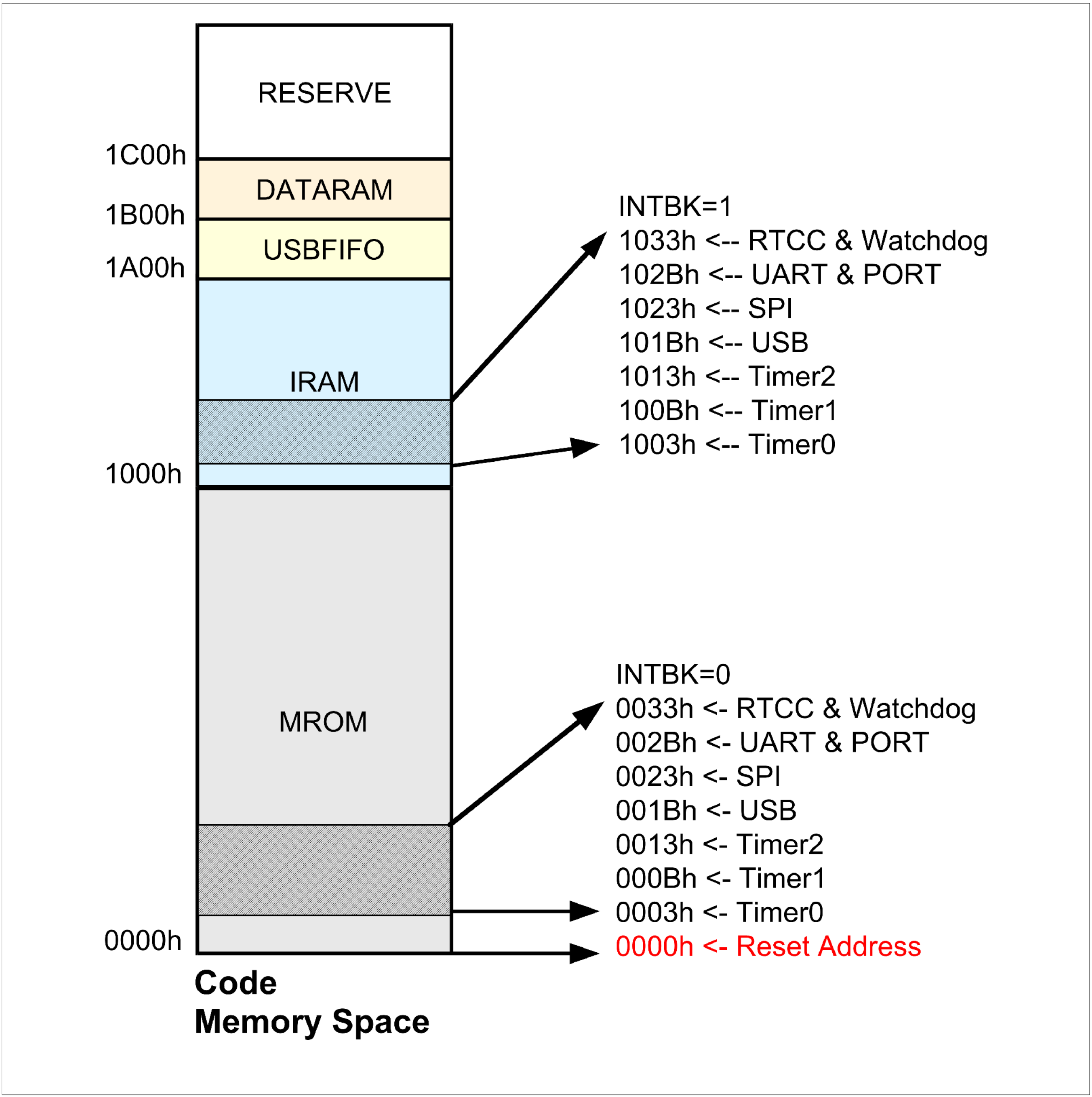
Figure 3-1: Memory mapping



3.2 Interrupt Entry Address Mapping

The first instruction starts from 0000h after Power Up. That means CPU will run the boot loader program stored in MASK ROM first, and also the interrupt address is remapping start from 0003h as a standard 8051/8052. By setting the interrupt entry bank address control bit **INTBK (DPCON.7)**, the interrupt entries address will be changed starting from 1003h.

Figure 3-2: Interrupt Entry Address Mapping



3.3 Data Memory – DRAM (DATA)

Data memory space (DATA) contains 256 bytes sketchpad memory, special function registers for controlling peripherals. It is also a super set of REGISTER space. To access this complex space, several addressing modes (direct, indirect, register and bit addressing) are provided. For details, please refer to the 8052 standard.

3.4 General Purpose Register (REGISTER)

General purpose registers *R0* through *R7* link to 32 bytes of internal data memory in the way that allows quick and efficient access. For example, the instruction *MOV A, 00h* using two bytes of code can be replaced by shorthand notation instruction *MOV A, R0* that uses one byte of code only.

These 32 bytes of memory are put into 4 banks. Any one of them within a bank is selected by *R0* through *R7*. Desired register bank is selected using bits *RS1* and *RS0* in PSW, bit 4 and bit 3 respectively (please refer to Table 3-1 for setting description). This feature eliminates the effort required to backup the registers to stack memory during context switching, in addition provides more registers for complicated algorithms.

Table 3-1: Selection of Register Bank

RS1	***RS0***	**Bank**	**Mapping Addressing to DATA**
0	0	0	00h-07h
0	1	1	08h-0Fh
1	0	2	10h-17h
1	1	3	18h-1Fh

3.5 Extended Data Memory – XRAM (XDATA)

To provide a unified linear memory model, AX206 organizes internal data memory that out of the scope of DATA into extended data memory space (XDATA). Total 3K bytes of XDATA (XRAM, shared with IRAM) are addressed through instruction *MOVX* using a 16-bit data pointers, *DPTR* .

3.6 CPU Instruction Set Summary

3.6.1 Basic Instruction Set

Table 3-2 lists the basic instructions of AX206, these instruction are compatible with 8051/8052 ISA.

Table 3-2: Basic Instruction Set Summary

Mnemonics		Description
ADD	A, Rn	Add register to *ACC*
	A, direct	Add direct byte to *ACC*
	A, @Ri	Add indirect byte from DRAM to *ACC*
	A, #data	Add immediate to *ACC*
ADDC	A, Rn	Add register to *ACC* with carry
	A, direct	Add direct byte to *ACC* with carry
	A, @Ri	Add indirect byte from DRAM to *ACC* with carry
	A, #data	Add immediate to *ACC* with carry
SUBB	A, Rn	Subtract register to *ACC* with borrow
	A, direct	Subtract direct byte to *ACC* with borrow
	A, @Ri	Subtract indirect byte from DRAM to *ACC* with borrow
	A, #data	Subtract immediate to *ACC* with borrow
INC	A	Increment *ACC*
	Rn	Increment register
	direct	Increment direct byte
	DPTR	Increment the selected *DPTR*
	@Ri	Increment indirect byte of DRAM
DEC	A	Decrement *ACC*
	Rn	Decrement register
	direct	Decrement direct byte
	DPTR	Decrement the selected *DPTR*
	@Ri	Decrement indirect byte of DRAM
MUL	AB	Multiply *ACC* and *B*
DIV	AB	Divide *ACC* by *B*
ANL	A, Rn	AND register to *ACC*
	A, direct	AND direct byte to *ACC*
	A, @Ri	AND indirect DRAM to *ACC*
	A, #data	AND immediate to *ACC*
	direct, A	AND *ACC* to direct byte
	direct, #data	AND immediate to direct byte
	C, bit	AND direct bit to carry
	C, /bit	AND complement of direct bit to carry

ORL	A, Rn	OR register to ACC
	A, direct	OR direct byte to ACC
	A, @Ri	OR indirect DRAM to ACC
	A, #data	OR immediate to ACC
	direct, A	OR ACC to direct byte
	direct, #data	OR immediate to direct byte
	C, bit	OR direct bit to carry
	C, /bit	OR complement of direct bit to carry
XRL	A, Rn	XOR register to ACC
	A, direct	XOR direct byte to ACC
	A, @Ri	XOR indirect DRAM to ACC
	A, #data	XOR immediate to ACC
	direct, A	XOR ACC to direct byte
	direct, #data	XOR immediate to direct byte
CLR	A	Clear ACC
CPL	A	Complement ACC
RL	A	Rotate ACC left
RLC	A	Rotate ACC left through carry
RR	A	Rotate ACC right
RRC	A	Rotate ACC right through carry
SWAP	A	*Swap nibbles of ACC*
MOV	A, Rn	Move register to ACC
	A, direct	Move direct byte to ACC
	A, @Ri	Move indirect DRAM to ACC
	A, #data	Move immediate to ACC
	Rn, A	Move ACC to register
	Rn, direct	Move direct byte to register
	Rn, #data	Move immediate to register
	direct, A	Move ACC to direct byte
	direct, Rn	Move register to direct byte
	direct, direct	Move direct byte to direct byte
	direct, @Ri	Move indirect DRAM to direct byte
	direct, #data	Move immediate to direct byte
	@Ri, A	Move ACC to indirect DRAM
	@Ri, direct	Move direct byte to indirect DRAM
	@Ri, #data	Move immediate to indirect DRAM
	DPTR, #data	Load DPTR with 16-bit constant
	C, bit	Move direct bit to Carry
	bit, C	Move Carry to direct bit
MOVC	A, @A+DPTR	Move code byte relative DPTR to ACC
	A, @A+PC	Move code byte relative PC to ACC

MOVX	A, @DPTR	Move external data (16-bit address) to ACC
	@DPTR, A	Move ACC to external data (16-bit address)
	MOVX A,@Ri	Move external data (Ri as Pointer) to ACC
	MOVX @Ri, A	Move ACC to external data (Ri as Pointer)
PUSH	direct	Push direct byte onto stack
POP	direct	Pop direct byte from stack
XCH	A, Rn	Exchange register with ACC
	A, direct	Exchange direct byte with ACC
	A, @Ri	Exchange indirect DRAM with ACC
XCHD	A, @Ri	Exchange low nibble of indirect DRAM with ACC
CLR	C	Clear carry
	bit	Clear direct bit
SETB	C	Set carry
	bit	Set direct bit
CPL	C	Complement carry
	bit	Complement direct bit
JC	rel code	Jump if carry is set
JNC	rel code	Jump if carry is not set
JB	bit, rel code	Jump if direct bit is set
JNB	bit, rel code	Jump if direct bit is not set
JBC	bit, rel code	Jump if direct bit is set and clear bit
ACALL	page code	Absolute subroutine call
LCALL	long code	Long subroutine call
RET		Return from subroutine
RETI		Return from interrupt
AJMP	page code	Absolute jump
LJMP	long code	Long jump
SJMP	rel addr	Short jump (relative address)
JMP	@A+DPTR	Jump indirect relative to DPTR
JZ	rel code	Jump if ACC equals zero
JNZ	rel code	Jump if ACC does not equal zero
CJNE	A, direct, rel code	Compare direct byte to ACC and jump if not equal
	A, #data, rel code	Compare immediate to ACC and jump if not equal
	Rn, #data, rel code	Compare immediate to Register and jump if not Equal
	@Ri, #data, rel code	Compare immediate to indirect and jump if not Equal
DJNZ	Rn, rel code	Decrement Register and jump if not zero
	direct, rel code	Decrement direct byte and jump if not zero
NOP		No operation

***Notes**:*
***Rn**: Register R0-R7 of the currently selected register bank.*
***@Ri**: Data RAM location addressed indirectly through R0 or R1.*

rel code*: 8-bit, signed (2s complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.*

direct*: 8-bit internal data location address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).*

#data*: 8-bit immediate data*

#data16*: 16-bit immediate data*

bit*: Direct-accessed bit in Data RAM or SFR*

page code*: 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-byte page of program memory as the first byte of the following instruction.*

long code*: 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64KByte program memory space.*

4 Special Function Register (SFR)

4.1 SFR List

The unimplemented bits are labelled '-', never write value other than its reset value to it, otherwise unpredictable effects will be resulted. Some registers have undetermined reset value, it is labelled 'x'.

Table 4-1: SFR List

Func.	Name	Addr	Des.	Reset Value							
CPU & INT	SP	81h	Stack Pointer	0	0	0	0	0	1	1	1
	DPL	82h	DPTR0 Low byte	0	0	0	0	0	0	0	0
	DPH	83h	DPTR0 High byte	0	0	0	0	0	0	0	0
	DPCON	86h	DPTR Control Register	0	0	0	0	0	0	0	0
	IE	A8h	Interrupt Enable Register	0	0	0	0	0	0	0	0
	IP	B8h	Interrupt Priority Register	0	0	0	0	0	0	0	0
	PSW	D0h	Program Status Register	0	0	0	0	0	0	0	0
	ACC	E0h	Accumulator	0	0	0	0	0	0	0	0
	EIF0	E8h	Interrupt Flag	0	0	0	0	0	0	0	0
	B	F0h	Register B	0	0	0	0	0	0	0	0
PORT	P0	80h	Port 0 Register	x	x	x	x	x	x	x	x
	P0DIR	E9h	Port 0 Direction Register	1	1	1	1	1	1	1	1
	P0UP	F9h	Port 0 Pull-Up Register	0	0	0	0	0	0	0	0
	P1	90h	Port 1 Register	x	x	x	x	x	x	x	x
	P1DIR	EAh	Port 1 Direction Register	1	1	1	1	1	1	1	1
	P1PU	FAh	Port 1 Pull-Up Register	0	0	0	0	0	0	0	0
	P2	A0h	Port 2 Register	x	x	x	x	x	x	x	x
	P2DIR	EBh	Port 2 Direction Register	1	1	1	1	1	1	1	1
	P2PU	FBh	Port 2 Pull-Up Register	0	0	0	0	0	0	0	0
	P3	B0h	Port 3 Register	x	x	x	x	x	x	x	x
	P3DIR	ECh	Port 3 Direction Register	1	1	1	1	1	1	1	1
	P3PU	FCh	Port 3 Pull-Up Register	0	0	0	0	0	0	0	0
	P4	C0h	Port 4 Register	-	-	-	-	-	x	x	x
	P4DIR	EDh	Port 4 Direction Register	-	-	-	-	-	1	1	1
	P4PU	FDh	Port 4 Pull-Up Register and PIE bit 9-8	1	1	-	-	-	0	0	0
	PIE	A4h	Port Interrupt Enable Flag	1	1	1	1	1	1	1	1
	WKPND	9Ah	Port Wakeup Pending Register	x	x	x	x	x	x	x	x
	WKEN	9Bh	Port Wakeup Enable Register	1	1	1	1	1	1	1	1
	WKEDG	9Ch	Port Wakeup Edge Register	x	x	x	x	x	x	x	x
SPI	SPICON	D8h	SPI Control Register	0	0	0	0	-	0	0	0
	SPIBAUD	D6h	SPI Baud rate Register	x	x	x	x	x	x	x	x
	SPIBUF	D7h	SPI Buffer Register	x	x	x	x	x	x	x	x

Func.	Name	Addr	Des.	Reset Value							
UART	UARTSTA	F1h	UART Status Register	0	0	0	0	-	-	-	0
	UARTCON	F2h	UART Control Register	0	1	0	0	0	0	0	0
	UARTBAUD	F3h	UART Baud Rate Register	x	x	x	x	x	x	x	x
	UARTBUF	F4h	UART Buffer Register	x	x	x	x	x	x	x	x
Timer	TMR0CON	B1h	Timer0 Control Register	0	-	-	-	0	0	0	0
	TMR0CNT	B3h	Timer0 Counter Register	x	x	x	x	x	x	x	x
	TMR0PR	B4h	Timer0 Period Register	1	1	1	1	1	1	1	1
	TMR0PSR	B5h	Timer0 Prescaler Register	-	-	-	-	-	x	x	x
	TMR1CON	E1h	Timer1 Control Register	0	0	0	0	0	0	0	0
	TMR1CNTL	E2h	Timer1 Counter Low Byte Register	x	x	x	x	x	x	x	x
	TMR1CNTH	E3h	Timer1 Counter High Byte Register	x	x	x	x	x	x	x	x
	TMR1PERL	E4h	Timer1 Period Low Byte Register	x	x	x	x	x	x	x	x
	TMR1PERH	E5h	Timer1 Period High Byte Register	x	x	x	x	x	x	x	x
	TMR1PWML	E6h	Timer1 PWM Low Byte Register	x	x	x	x	x	x	x	x
	TMR1PWMH	E7h	Timer1 PWM High Byte Register	x	x	x	x	x	x	x	x
	TMR2CON	C1h	Timer2 Control Register	0	0	0	0	0	0	0	0
	TMR2CNTL	C2h	Timer2 Counter Low Byte Register	x	x	x	x	x	x	x	x
	TMR2CNTH	C3h	Timer2 Counter High Byte Register	x	x	x	x	x	x	x	x
	TMR2PERL	C4h	Timer2 Period Low Byte Register	x	x	x	x	x	x	x	x
	TMR2PERH	C5h	Timer2 Period High Byte Register	x	x	x	x	x	x	x	x
	TMR2PWML	C6h	Timer2 PWM Low Byte Register	x	x	x	x	x	x	x	x
	TMR2PWMH	C7h	Timer2 PWM High Byte Register	x	x	x	x	x	x	x	x
	TMR3CON	F8h	Timer3 (RTCC) Control Register	0	0	0	0	-	0	0	0
	RTCNT	D1h	Timer3 (RTCC) Counter Register	x	x	x	x	x	x	x	x
	WDTCON	BBh	Watchdog Control Register	0	0	0	0	0	0	0	0
MUL	MULXL	CBh	Multiplier Operand X Low Byte Register	x	x	x	x	x	x	x	x
	MULXH	CCh	Multiplier Operand X High Byte Register	x	x	x	x	x	x	x	x
	MULYL	CDh	Multiplier Operand Y Low Byte Register	x	x	x	x	x	x	x	x
	MULYH	CEh	Multiplier Operand Y High Byte Register	x	x	x	x	x	x	x	x
	MULCON	98h	Multiplier Control Register	-	-	0	0	0	0	0	0
	MULRES0	91h	Multiplier Result's 0 Register	x	x	x	x	x	x	x	x
	MULRES1	92h	Multiplier Result's 1 Register	x	x	x	x	x	x	x	x
	MULRES2	93h	Multiplier Result's 2 Register	x	x	x	x	x	x	x	x
	MULRES3	94h	Multiplier Result's 3 Register	x	x	x	x	x	x	x	x
Bit-Fetcher	BFCNT	BCh	Bit-Fetcher Counter Register	-	-	-	-	x	x	x	x
	BFBUF0	BDh	Bit-Fetcher Buffer Low Byte Register	x	x	x	x	x	x	x	x
	BFBUF1	BEh	Bit-Fetcher Buffer Medium Byte Register	x	x	x	x	x	x	x	x
	BFBUF2	BFh	Bit-Fetcher Buffer High Byte Register	x	x	x	x	x	x	x	x

Func.	Name	Addr	Des.	Reset Value							
USB	USBCON	C8h	USB Control 0 Register	0	0	0	0	0	0	0	0
	USBBUF	C9h	USB Buffer Register	x	x	x	x	x	x	x	x
	USBADR	CAh	USB Address Register	x	x	x	x	x	x	x	x
ADC	ADCCON	D2h	ADC Control Register	0	0	0	0	0	0	0	0
	ADCBAUD	D3h	ADC Baud Rate Register	0	0	x	x	x	x	x	x
	ADCBUFH	D4h	ADC Buffer High Byte Register	x	x	x	x	x	x	x	x
	ADCBUFL	DCh	ADC Buffer Low Byte Register	x	x	x	x	x	x	x	x
Clock	PCON	87h	Power Control Register	0	0	0	0	0	0	0	0
	CKCON	A5h	Clock Control0 Register	0	0	0	0	0	0	0	0

Notes:
🟧: bit accessible

4.2 CPU and Interrupt SFR

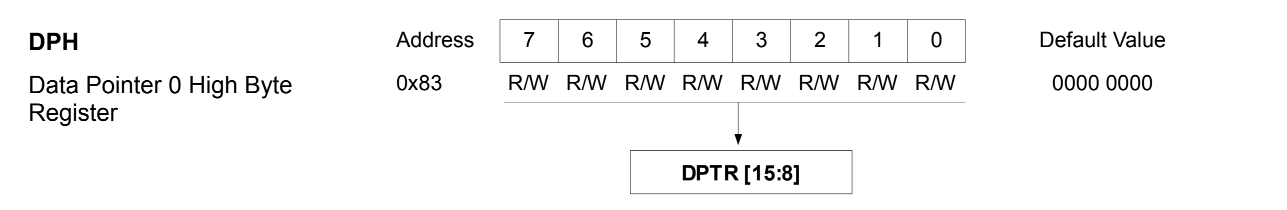
Register 4-1: SP – Stack Pointer Register

SP	Address	7	6	5	4	3	2	1	0	Default Value
Stack Pointer Register	0x81	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0000 0111
		SP [7:0]								

Register 4-2: DPL - Data Pointer 0 Low Byte Register

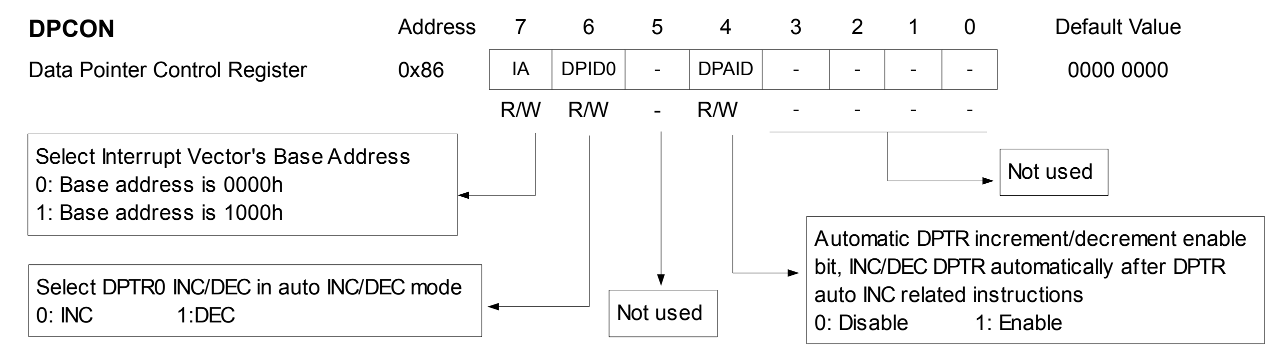
DPL	Address	7	6	5	4	3	2	1	0	Default Value
Data Pointer 0 Low Byte Register	0x82	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0000 0000
		DPTR [7:0]								

Register 4-3: DPH - Data Pointer 0 High Byte Register



DPH	Address	7	6	5	4	3	2	1	0	Default Value
Data Pointer 0 High Byte Register	0x83	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	0000 0000

Register 4-4: DPCON – Data Pointer Control Register



DPCON	Address	7	6	5	4	3	2	1	0	Default Value
Data Pointer Control Register	0x86	IA	DPID0	-	DPAID	-	-	-	-	0000 0000
		R/W	R/W	-	R/W	-	-	-	-	

Select Interrupt Vector's Base Address
0: Base address is 0000h
1: Base address is 1000h

Select DPTR0 INC/DEC in auto INC/DEC mode
0: INC 1:DEC

Not used

Automatic DPTR increment/decrement enable bit, INC/DEC DPTR automatically after DPTR auto INC related instructions
0: Disable 1: Enable

Not used

Notes:

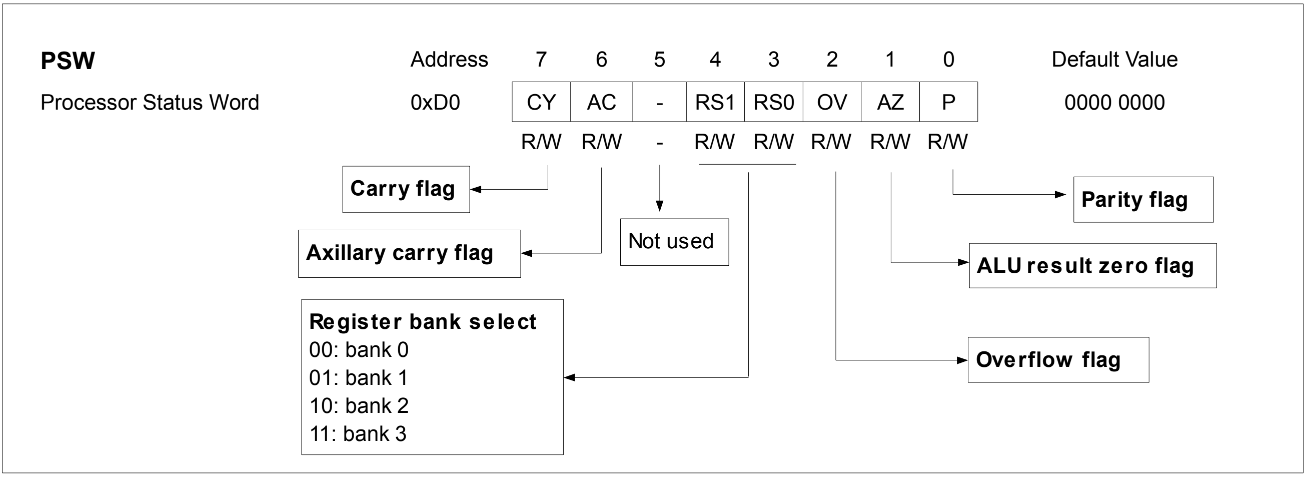
DPTR Toggle related instructions are:

- *MOVC A, @A+DPTR*
- *MOVX A, @DPTR*
- *MOVX @DPTR, A*
- *INC DPTR*
- *MOV DPTR, #data16*

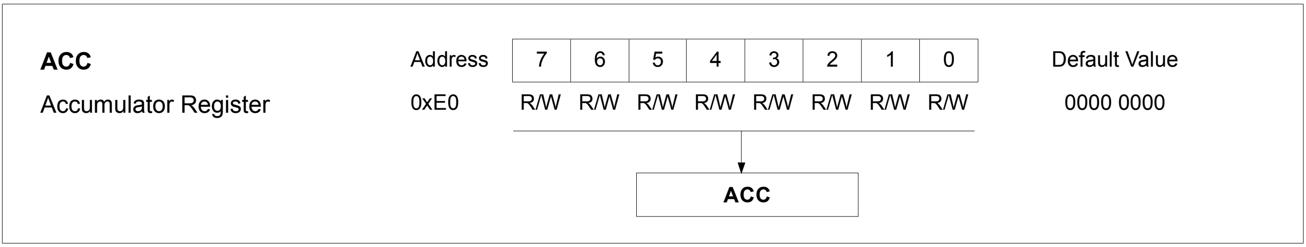
DPTR Auto INC related instructions are:

- *MOVC A, @A+DPTR*
- *MOVX A, @DPTR*
- *MOVX @DPTR, A*

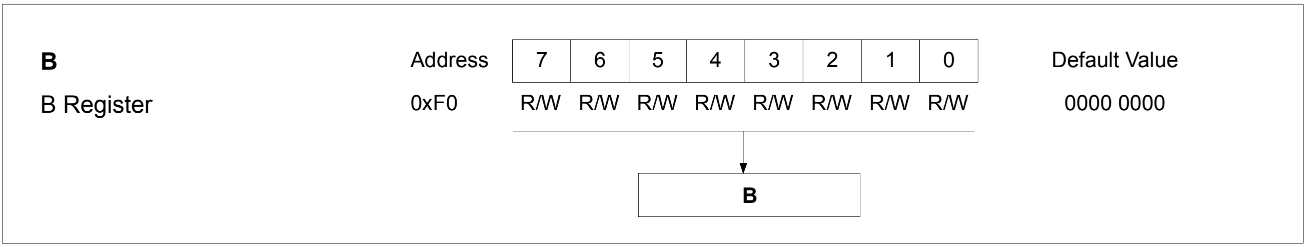
Register 4-5: PSW - Processor Status Word



Register 4-6: ACC - Accumulator Register



Register 4-7: B – B Register



5 Interrupt Processing

AX206 extends the interrupt system to support totally 10 interrupt sources allocating in 7 interrupt vectors with two priority levels. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

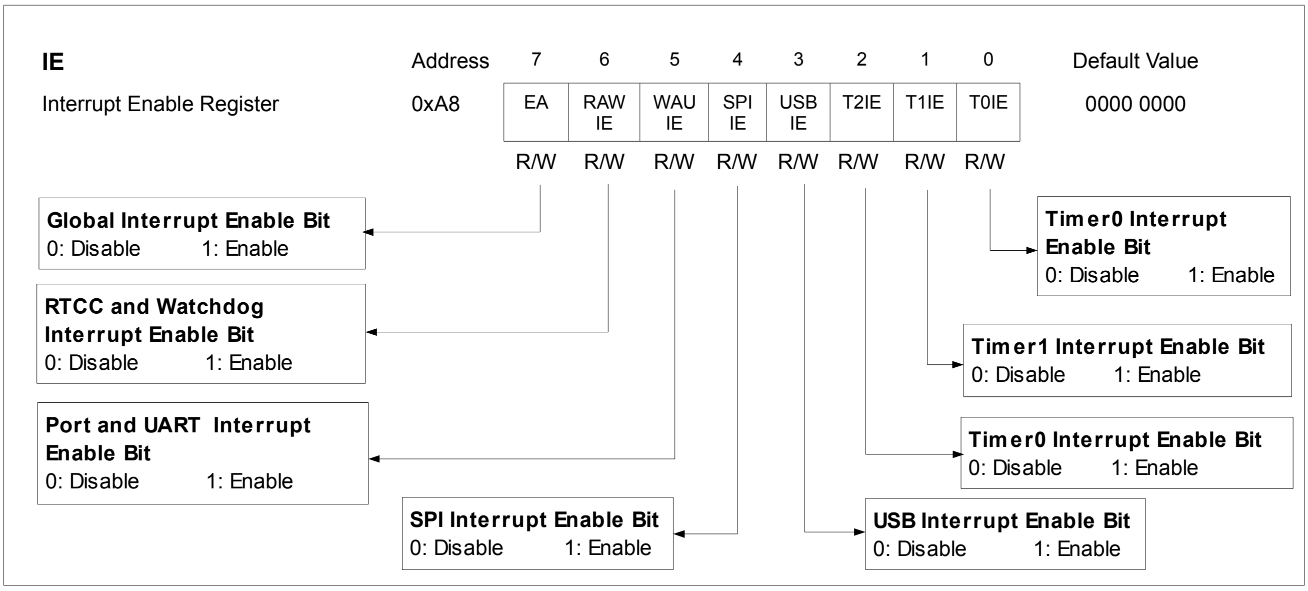
If interrupt is enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU backs up the location of the next instruction to STACK and then begins execution of an interrupt service routine (ISR). Each ISR must end with an *RETI* instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the *EA* bit (*IE.7*) to logic 1 before the individual interrupt enables are recognized.

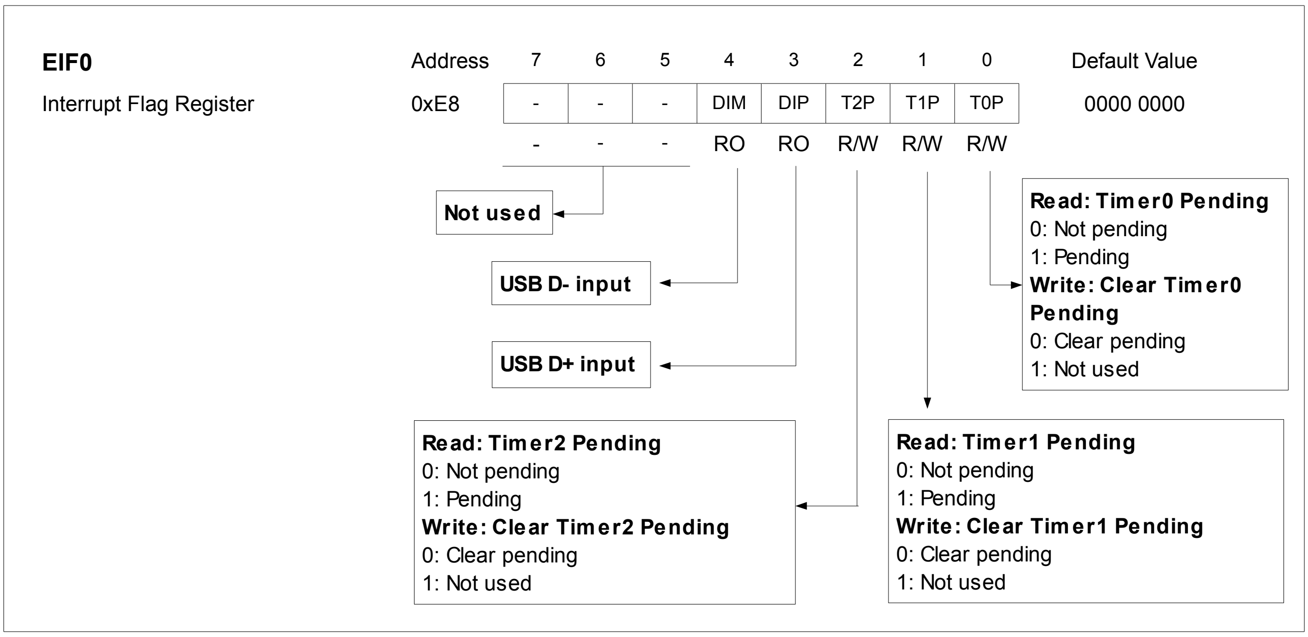
Setting the *EA* bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the *EA* bit is set to logic 0 will be held in a pending state, and will not be serviced until the *EA* bit is set back to logic 1.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (*RETI*) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

Register 5-1: IE - Interrupt Enable Register



Register 5-2: EIF0 - Interrupt Flag Register



5.1 Interrupt Sources and Vectors

The CPU supports 10 interrupt sources. Some interrupt sources share the same interrupt vector. Software can simulate an interrupt by setting some interrupt-pending flags to ‘1’. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order, and control bits are summarized in Table 5-1. Please refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

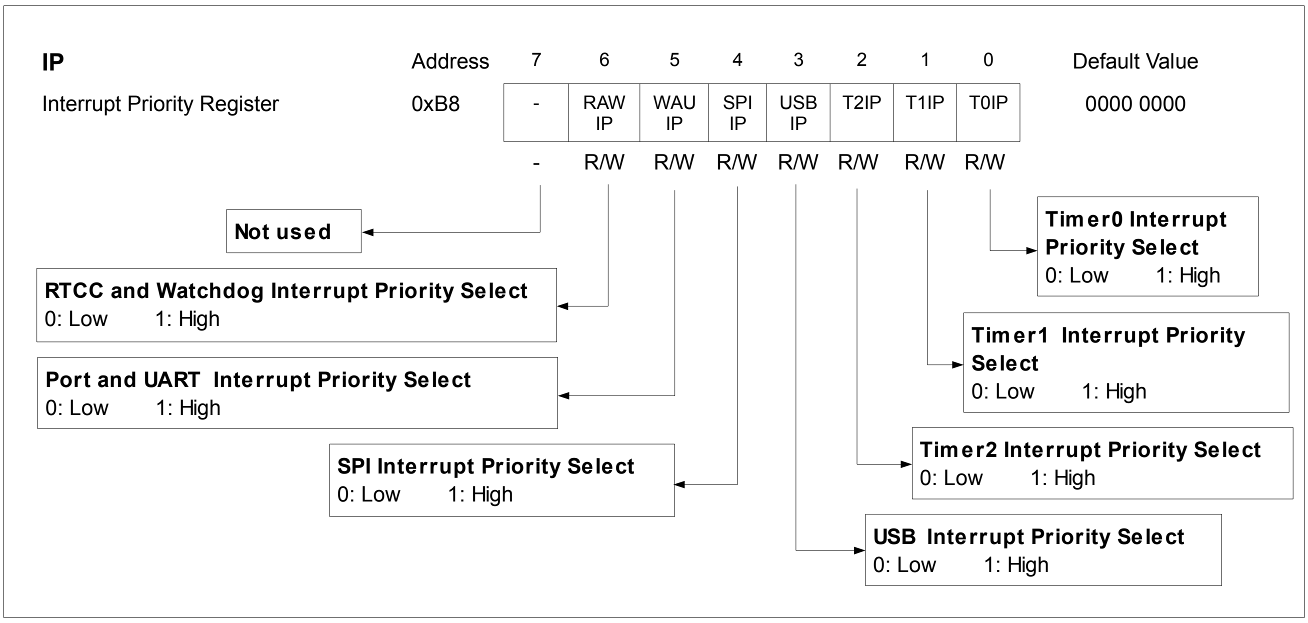
5.2 Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be pre-empted by a high priority interrupt. A high priority interrupt cannot be pre-empted. Each interrupt has an associated interrupt priority bit in an SFR (*IP*) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 5-1.

Table 5-1: Summary of Interrupts

Interrupt Source	Vector	Priority	Pending Flag	Enable Bit	Enable Bit 2	Priority Bit
Timer0	0003h	1 (Highest)	*EIF.0*	*IE.0*	-	*IP.0*
Timer1	000Bh	2	*EIF.1*	*IE.1*	-	*IP.1*
Timer2	0013h	3	*EIF.2*	*IE.2*	-	*IP.2*
USB SOF	001Bh	4	*USBCON.7*	*IE.3*	-	*IP.3*
USB CTL			*N/A*			
SPI	0023h	5	*SPICON.7*	*IE.4*	-	*IP.4*
Port Wakeup	002Bh	6	*WKPND*	*IE.5*	*WKEN*	*IP.5*
UART			*UARTSTA.[7:6]*		*UARTCON[1:0]*	
Watchdog	0033h	7 (Lowest)	*WDTCON.5*	*IE6*	*WDTCON.4/IP.7*	*IP.6*
RTCC			*T3CON.3*		*TMR3CON.2*	

Register 5-3: IP - Interrupt Priority Register


IP
Interrupt Priority Register
Address
0xB8
7
6
5
4
3
2
1
0
-
RAW IP
WAU IP
SPI IP
USB IP
T2IP
T1IP
T0IP
Default Value
0000 0000
-
R/W
R/W
R/W
R/W
R/W
R/W
R/W
Not used
RTCC and Watchdog Interrupt Priority Select
0: Low 1: High
Port and UART Interrupt Priority Select
0: Low 1: High
SPI Interrupt Priority Select
0: Low 1: High
Timer0 Interrupt Priority Select
0: Low 1: High
Timer1 Interrupt Priority Select
0: Low 1: High
Timer2 Interrupt Priority Select
0: Low 1: High
USB Interrupt Priority Select
0: Low 1: High

5.3 Interrupt Latency

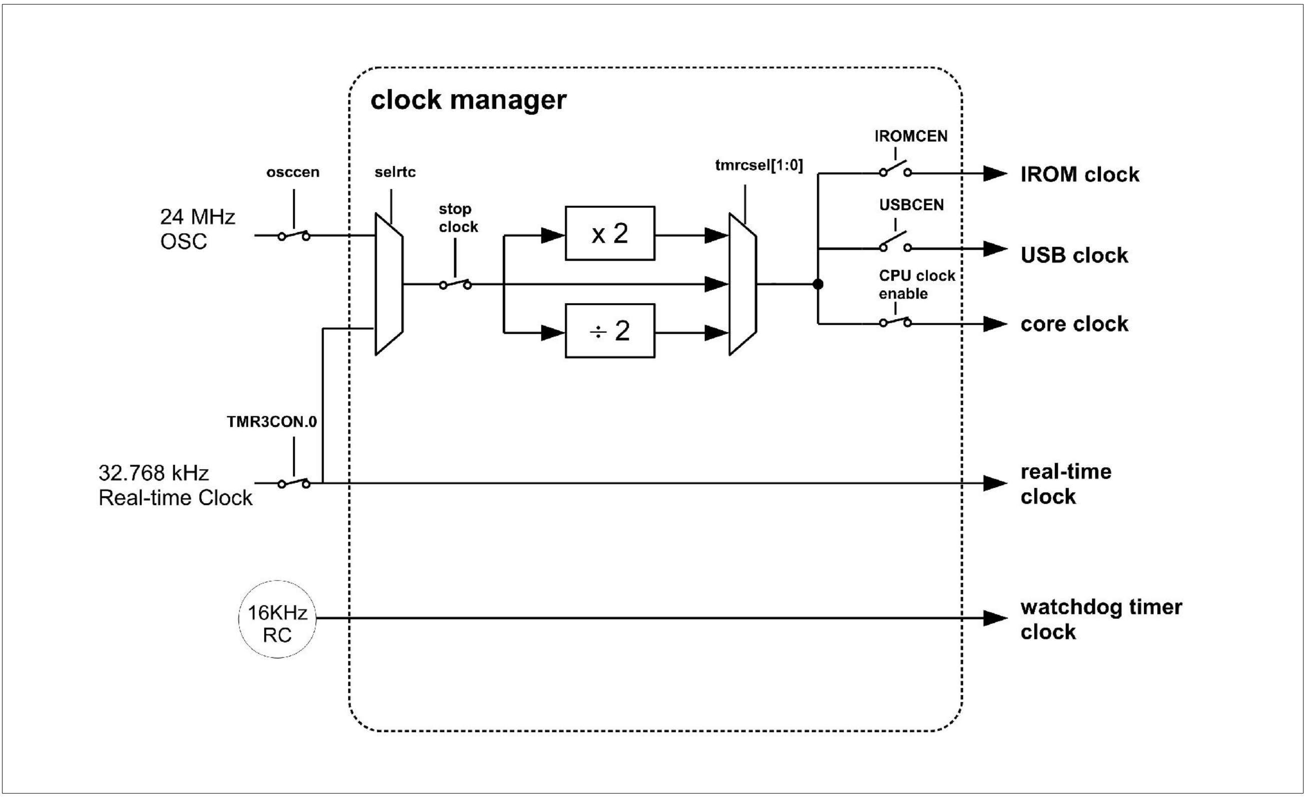
Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the response time is 4–6 system clock cycles: 1 clock cycle to detect the interrupt, 2 clock cycles to back up the location of next instruction, and 1–3 clock cycles to complete the fetch and to execute the first instruction of ISR, depending on the instruction length. If an interrupt is pending when a *RETI* is executed, a single instruction is executed before serving the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an *RETI* instruction followed by a 3-byte instruction as the next instruction. In this case, the response time is 10-12 system clock cycles: 1 clock cycle to detect the interrupt, 3 clock cycles to execute the RETI, 3 clock cycles to fetch and complete the following 3-byte instruction, 2 clock cycles to back up the location of next instruction and 1–3 clock cycles to complete the fetch the first instruction of ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the *RETI* and following instruction.

6 Clocks and Reset Management

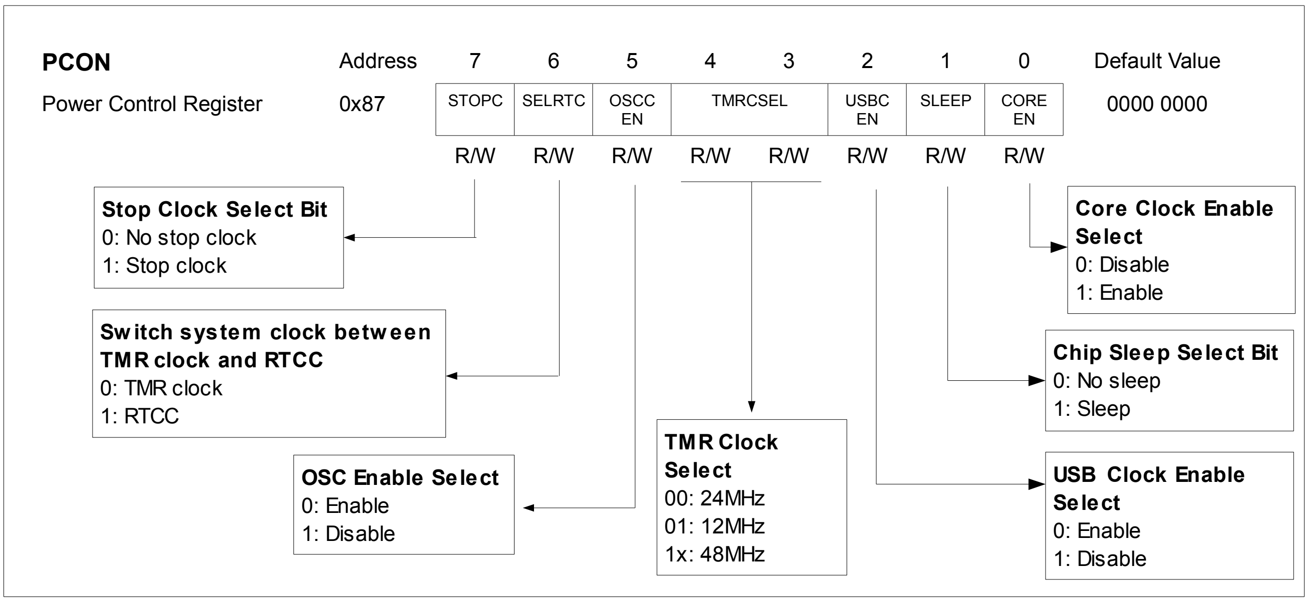
6.1 Clock System

AX206 possesses a configurable clock system aiming to provide balance between performance and power consumption in different applications. It provides fine-grain clock gating to shutdown unused part completely.

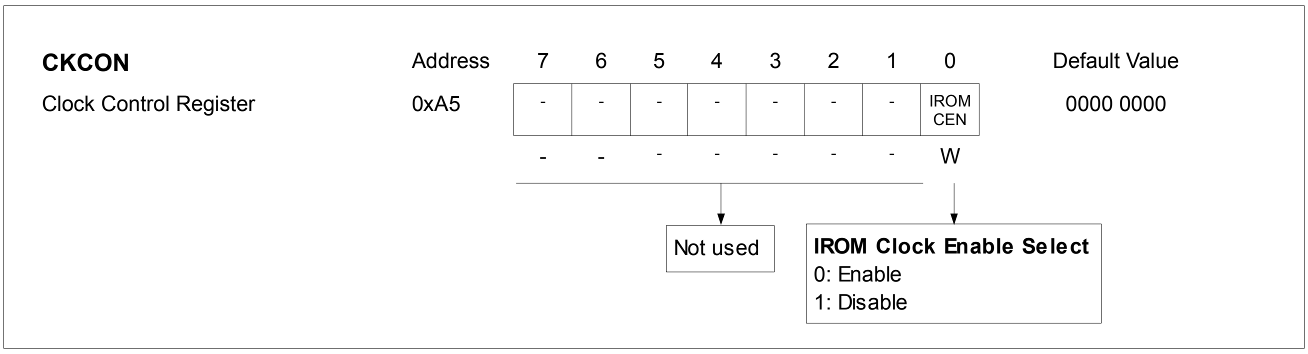
Figure 6-1: Clock System Block Diagram



Register 6-1: PCON – Power Control Register



Register 6-2: CKCON – Clock Control Register



6.1.1 Clock Source Control

Two clock domains are implemented in AX206, system clock domain and USB clock domain.

The major clock source is a high speed crystal oscillator that suits for 24MHz crystals. This clock is used directly, or is doubled, or is divided down to generate system clock for CPU and peripherals. There is another optional clock source, a 32,768 Hz crystal oscillator primarily for real-time clock, user can also choose this low speed clock source for power saving.

The selection of clock sources is controlled by setting register *SELRTC* (PCON.6). During clock switching the system clock suspends for 2 periods of the slower clock source. User must make sure the desired clock source is running before switching, otherwise the system stops which can be recovered by reset only.

System Clock Domain:

System clock is from either RT oscillator (32.768K clock output) or oscillator (24MHz), selected by *SELRTC*

(PCON.6). By default, *SELRTC* = 0 after power up and oscillator is selected for the system clock source as shown in Figure 6-1. System will run in 24MHz. By setting *TMRCSEL[1]*, the built-in x2 PLL will work to supply the clock with double frequency (48MHz) to system if needs, this example operation codes are shown as below:

```
//enable the x2 PLL

ORL PCON, #0x10

//insert 1 cycles delay

NOP
```

USB Clock Domain:
USB is inverted with System Clock. USB is enabled by default. After POR (power on reset), USB Clock run at 24MHz as well as System Clock. While connecting with USB Host, setting *TMRSEL[1]* to '1' will speed up both System Clock and USB Clock with double frequency (48MHz), setting *USBCEN* to '1' will disable USB Clock while disconnecting with USB host for saving power.

6.1.2 Clock Gating Control

There are 6 bits in *PCON* and 1 bit in *CKCON* to control the clock gating for the 6 main parts in AX206, there are *COREEN (PCON.0)* for CPU, *IROMCEN (CKCON.0)* for IROM, *OSCCEN (PCON.5)* for 24MHz oscillator, *USBCEN (PCON.2)* for USB, *RTCEN (TMR3CON.0)* for RTC and *WDTEN (WDTCON.4)* for RC.

Idle Mode:
Setting *COREEN (PCON.0)* will force the CPU into a idle mode, CPU stops running instruction until any interrupts happen (*EA* = 1 is a must). The interrupt will wake up the CPU and CPU will go to the ISR of this interrupt source.

Stop-Clock Mode:
Setting STOPC will force the CPU into Stop-Clock mode, CPU and Peripheral will stop running until RTCC and port wakeup interrupts happen (EA = 1 is a must). The interrupt will wake up the CPU and CPU will go to the ISR of this interrupt source.

6.1.3 Oscillator configurations

Table 6-1 illustrates recommended configuration of crystal/resonator oscillator at different operating frequency. R_F is the motional resistance of the crystal/resonator and can be found in crystal/resonator vendor's datasheet. C_1 and C_2 represents the two external loading parallel capacitors C_1 and C_2.

The desired output frequency of the crystal/resonator can be fine tuned by adjusting loading capacitors C_1 and C_2. The tuning range is highly dependent on crystal/resonator and users need to consult the crystal/resonator vendor for details.

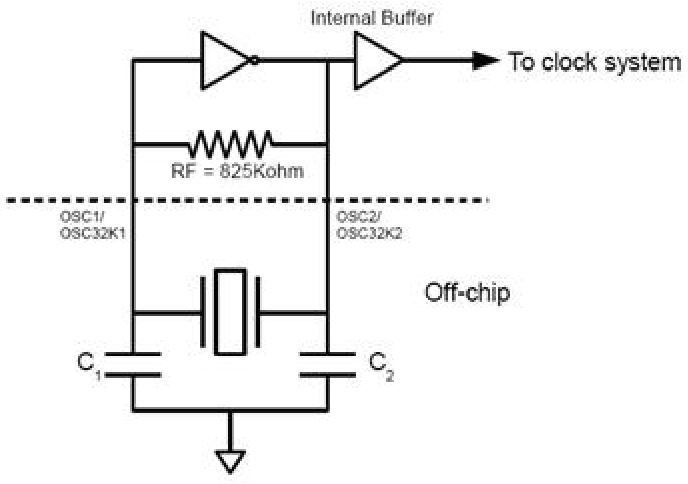
Table 6-1: Use of R_M and C_1, C_2

Crystal	Maximum R_F (ohm)	Loading capacitor C_1, C_2 (pF)
32KHz	50K	30
24MHz	40	30

6.1.4 PCB layout recommendation

Precaution should be taken when drawing printed-circuit board (PCB) layout for crystal. The crystal/resonator and the loading capacitor C1//C2 should be placed closest to AX206 OSC1/OSC2 pins and OSC32K1/OSC32K2 pins. If space is allowed, a grounded-ring surrounding the crystal and loading capacitors are always recommended in order to reduce coupling and noise from the near environment.

Figure 6-2: Crystal oscillator connection diagram



6.1.5 RC Oscillator

The 16MHz RC oscillator is an on-chip device to supply clock for watchdog, reset circuit and oscillator stabilization circuit. It is enabled automatically when (1) watchdog is enabled; (2) CPU in reset state and (3) an oscillator is re-activated. However, RC oscillator cannot be clock source for other modules.

6.2 Reset System

AX206 has several different reset sources. They are grouped into 2 classifications: normal resets and induced resets. Normal resets present in typical MCUs. They are:

1. Master clear (External reset through pin /MCLR)
2. Power on reset
3. Watchdog timeout reset

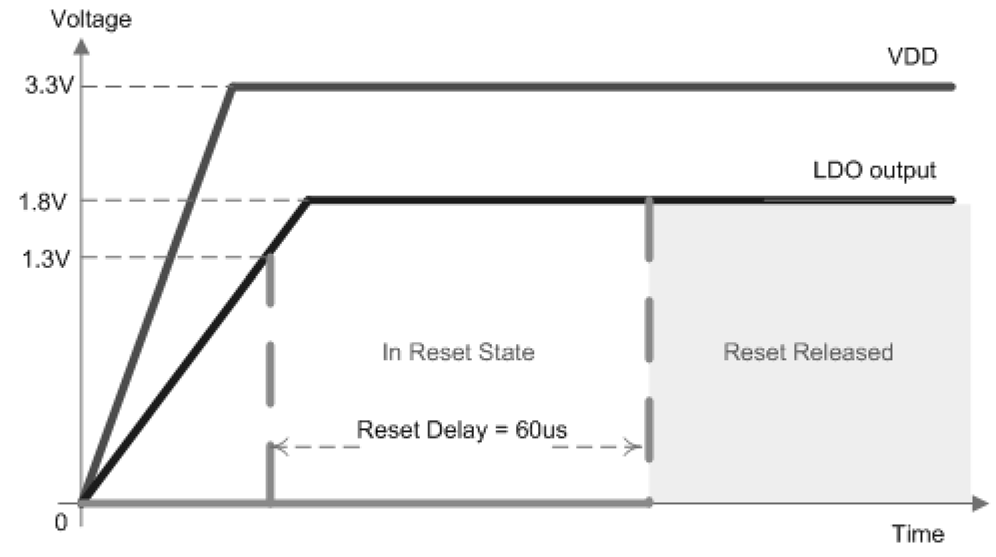
The other type of resets is namely induced, because these are not reset source in normal operation. They reset the system due to recovery from power down mode when wakeup. The induced resets are:

1. Port wakeup in power down mode
2. Real-time wakeup in power down mode

Although the cause and condition of these 2 kinds of resets are different, they force the system to initial state in the same way. However, some registers have different reset value under different reset sources, and this is going to be discussed in the following section.

Only POR (power on reset) will reset the whole system of AX206. It's illustrated in Figure 6-3.

Figure 6-3: Reset Timing



6.2.1 Reset Sequence

When the reset happens, AX206 falls back to initial state and is held. At this moment, the high speed oscillator and the RC oscillator are running. Once the reset is released, the oscillator stabilization counter starts counting. It counts for 16 ms to the oscillator becomes stable. After that, the CPU resumes and executes the first instruction located at program counter 0x0000.

6.2.2 Master Reset

AX206 has a noise filter in master reset path. The filter blocks the small pulses (shorter than 8 ms) appearing at pin /MCLR.

6.2.3 Power On Reset

AX206 provides an on-chip Power-on Reset (POR) circuit to detect power-on and to reset internal logic before VDD reaches the pre-determined POR threshold voltage. Under VDD=3.3V, the POR threshold voltage is set to be about 2.2V.

Sometimes, when the VDD is powered off and quickly powered on again, there might be cases that the POR will work improperly and internal reset might not be generated. For this reason, AX206 POR circuit incorporates an internal self-reset module to discharge PORB output during power-off to ensure each power cycle will work properly.

6.2.4 Watchdog Timeout Reset

The watchdog timer (WDT) subsystem protects the microcontroller system from incorrect code execution over a long period of time by causing a system reset when the watchdog timer overflows as a result of a failure of software to feed the timer prior to the timer reaching its terminal count.

For the operation of watchdog timer, please refer to Section 8.4.

6.3 Power Management

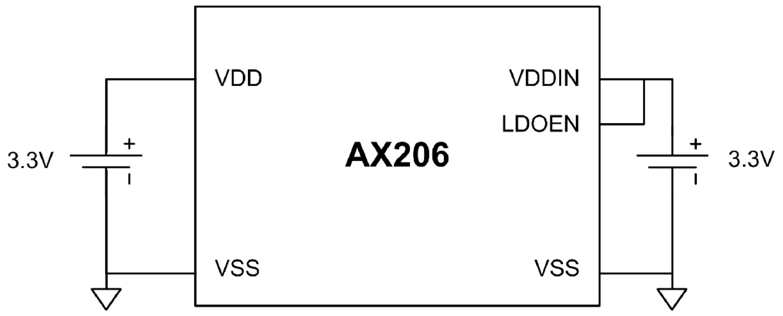
6.3.1 Operating in 3.3V or 5V Systems

AX206 is designed to operate in 3.3V or 5V system. It has on-chip regulator and separate supply to IO pin to guarantee seamless interfacing with 3.3V or 5V off-chip peripherals.

Operating in 3.3V System

In a 3.3V system, the on-chip regulator should be turned off by tying LDOEN to VDD. 3.3V power should be connected to both VDD and VDDIN.

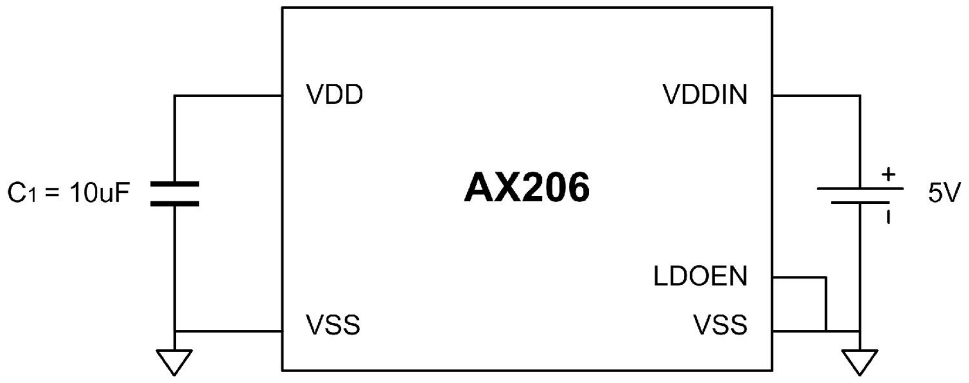
Figure 6-4: Topology of supplying 3.3V to AX206



Operating in 5V System

In a 5V system, the on-chip regulator should be turned on by tying LDOEN to VSS. 5V power should be connected to VDDIN only. A capacitor should be attached to VDD and VSS to ensure good quality of on-chip regulator output.

Figure 6-5: Topology of supplying 5V to AX206



Note:
1. The recommended value for C1 is 10uF.
2. C1 should be placed closely to the chip.

6.3.2 Idle Mode

Idle mode is the first level of power saving mode. The CPU clock stops but the rest of the clocks remains. Idle mode is activated by setting COREEN, bit 0 of PCON to '1'. Any enabled interrupt and reset sources can resume the core clock and deactivate idle mode.

Wakeup by enabled interrupts

Action of CPU:	Resumes and serves the interrupt request.
Wakeup sources:	All enabled interrupts

Wakeup by resets

Action of CPU:	Resets and serves the reset routine at program counter 0x0000
Wakeup sources:	All resets

6.3.3 Halt Mode

Halt mode aggressively shuts down the whole system clock from the "stop clock" switch showing in Figure Error: Reference source not found. At this moment, there is no activity in any module. Halt mode is activated by setting STOPC, bit 7 of PCON to '1'. Only selected interrupt and reset sources can resume the clocks and deactivate halt mode.

Wakeup by enabled interrupts

Action of CPU:	Resumes and serves the interrupt request.
Wakeup sources:	Enabled watchdog interrupt, port wakeup interrupt and real-time wakeup interrupt

Wakeup by disabled interrupts

Action of CPU:	Resumes and executes the next instruction before halt
Wakeup sources:	Disabled watchdog interrupt, port wakeup interrupt and real-time wakeup interrupt (corresponding interrupt enable bit is cleared)

Wakeup by resets

Action of CPU:	Resets and serves the reset routine at program counter 0x0000
Wakeup sources:	All resets

6.3.4 Power Down Mode

The definition of power down mode is that the whole system clock is shut down by turning off the source crystal oscillator. As AX206 has dual crystal oscillators, the condition of power down is tabulated in Table 6-2. The procedures of stopping crystal oscillators are shown in Section 6.1.2. It is recommended to disable all peripherals and also DPLL in power down mode. Only selected interrupt and reset sources can resume the clocks and deactivate power down mode.

Table 6-2: Power down condition

System clock source	Stop high-speed oscillator	Stop 32,768 Hz oscillator
High-speed oscillator	Power Down mode	Normal mode
Real-time oscillator	Normal mode	Power Down mode

Wakeup by enabled interrupts

Action of CPU:	Resets and serves the reset routine at program counter 0x0000
Wakeup sources:	Enabled port wakeup interrupt and real-time wakeup interrupt

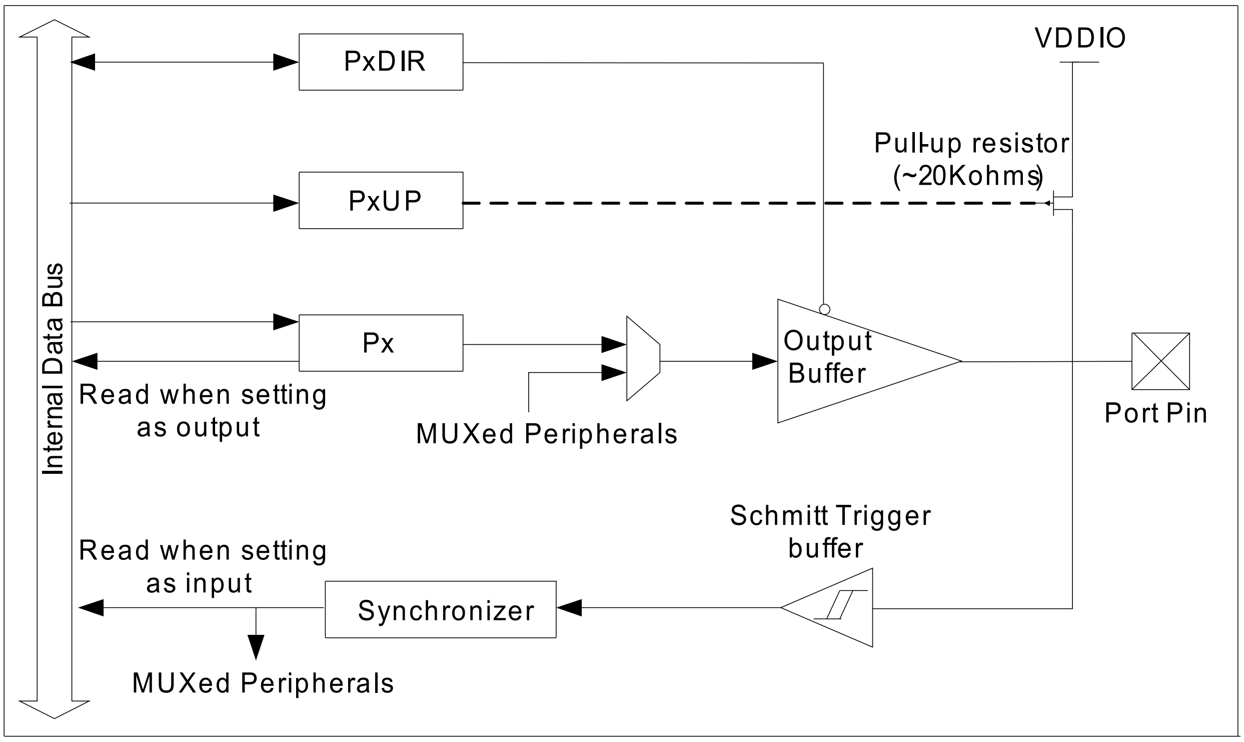
Wakeup by resets

Action of CPU:	Resets and serves the reset routine at program counter 0x0000
Wakeup sources:	All resets

7 Ports

AX206 provides 4 full ports (Port0, 1, 2, 3) and a half (Port4) GPIO pins for user to develop applications. It has a total of 35 GPIO pins. Inputs are all Schmitt triggered with about 400-500mV hysteresis level to filter input voltage fluctuations. Each port pin can be independently set as input or output. Output source/sink driving strength is 8mA. Most of the port pins are built-in slew-rate controlled to reduce output bouncing noise. There is also an internally 20KΩ pull-up resistor selectable for each output port pin. Figure 7-1 shows the structure of the GPIO.

Figure 7-1: Structure of IO



7.1 Registers

Register 7-1: P0 - Port 0 Register

P0	Address	7	6	5	4	3	2	1	0	Default Value
Port 0 Register	0x80	**P0[7:0]**								xxxx xxxx
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Port 0 Data Register

Register 7-2: P0DIR - Port 0 Direction Control Register

P0DIR	Address	7	6	5	4	3	2	1	0	Default Value
Port 0 Direction Control Register	0xE9	**P0DIR[7:0]**								1111 1111
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Port 0 pin 7/6/5/4/3/2/1/0 direction control bit
0: Output 1: Input

Register 7-3: P0PU - Port 0 Pull-up Register

P0PU	Address	7	6	5	4	3	2	1	0	Default Value
Port 0 Pull-up Register	0xF9	**P0PU[7:0]**								0000 0000
		W	W	W	W	W	W	W	W	

Port 0 pin 7/6/5/4/3/2/1/0 Pull-up Enable Bit
0: Disable 1: Enable

Register 7-4: P1 - Port 1 Register

	Address	7	6	5	4	3	2	1	0	Default Value
P1										
Port 1 Register	0x90	**P1[7:0]**								xxxx xxxx
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
		Port 1 Data Register								

Register 7-5: P1DIR - Port 1 Direction Control Register

	Address	7	6	5	4	3	2	1	0	Default Value
P1DIR										
Port 1 Direction Control Register	0xEA	**P1DIR[7:0]**								1111 1111
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
		Port 1 pin 7/6/5/4/3/2/1/0 direction control bit 0: Output 1: Input								

Register 7-6: P1PU - Port 1 Pull-up Register

	Address	7	6	5	4	3	2	1	0	Default Value
P1PU										
Port 1 Pull-up Register	0xFA	**P1PU[7:0]**								0000 0000
		W	W	W	W	W	W	W	W	
		Port 1 pin 7/6/5/4/3/2/1/0 Pull-up Enable Bit 0: Disable 1: Enable								

Register 7-7: P2 - Port 2 Register

	Address	7	6	5	4	3	2	1	0	Default Value
P2										
Port 2 Register	0xA0	**P2[7:0]**								xxxx xxxx
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
		Port 2 Data Register								

Register 7-8: P2DIR - Port 2 Direction Control Register

P2DIR	Address	7	6	5	4	3	2	1	0	Default Value
Port 2 Direction Control Register	0xEB	**P2DIR[7:0]**								1111 1111
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Port 2 pin 7/6/5/4/3/2/1/0 direction control bit
0: Output 1: Input

Register 7-9: P2PU - Port 2 Pull-up Control Register

P2PU	Address	7	6	5	4	3	2	1	0	Default Value
Port 2 Pull-up Control Register	0xFB	**P2PU[7:0]**								0000 0000
		W	W	W	W	W	W	W	W	

Port 2 pin 7/6/5/4/3/2/1/0 Pull-up Enable Bit
0: Disable 1: Enable

Register 7-10: P3 - Port 3 Register

P3	Address	7	6	5	4	3	2	1	0	Default Value
Port 3 Register	0xB0	**P3[7:0]**								xxxx xxxx
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

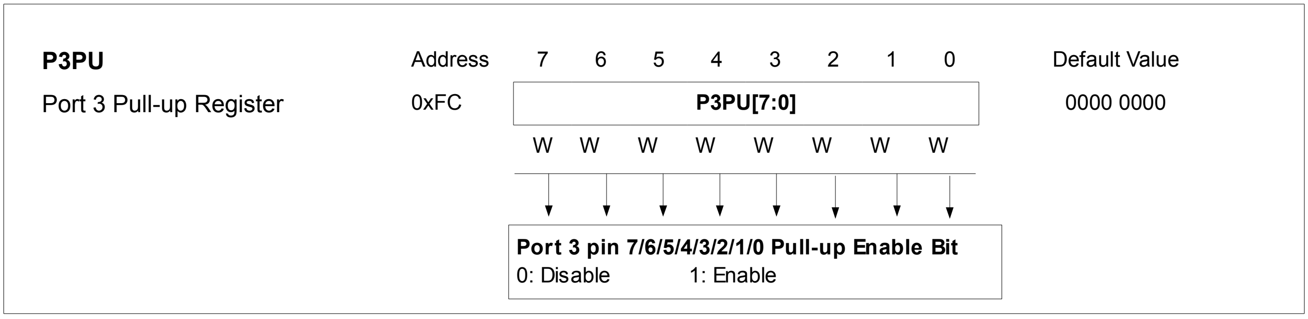
Port 3 Data Register

Register 7-11: P3DIR - Port 3 Direction Control Register

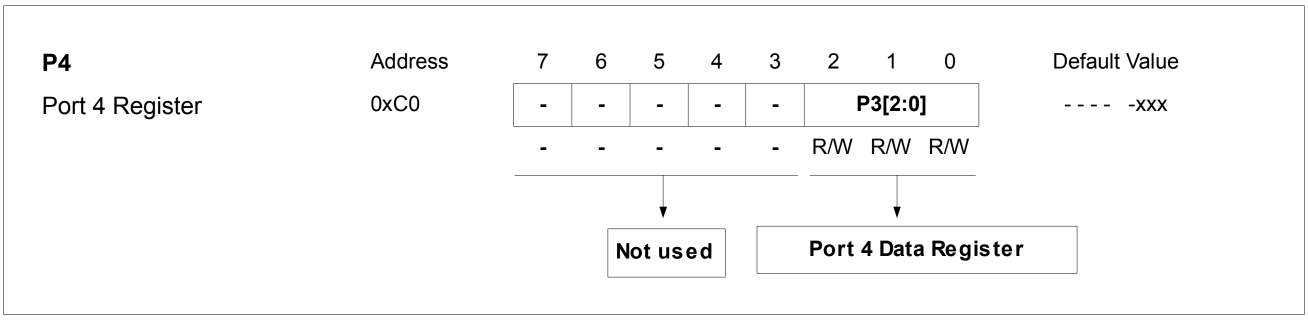
P3DIR	Address	7	6	5	4	3	2	1	0	Default Value
Port 3 Direction Control Register	0xEC	**P3DIR[7:0]**								1111 1111
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Port 3 pin 7/6/5/4/3/2/1/0 direction control bit
0: Output 1: Input

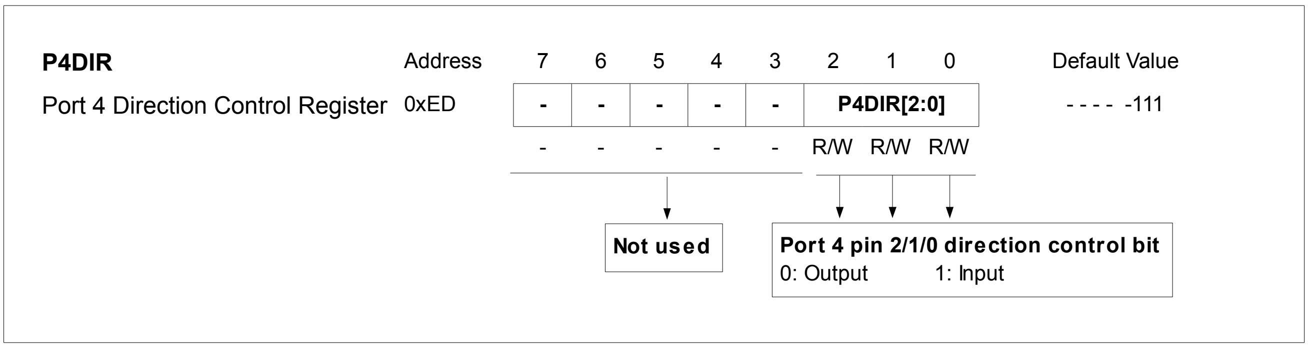
Register 7-12: P3PU - Port 3 Pull-up Register



Register 7-13: P4 - Port 4 Register



Register 7-14: P4DIR - Port 4 Direction Control Register



Register 7-15: P4PU – Port 4 Pull-up Register

P4PU
Port 4 Pull-up Register

Address	7	6	5	4	3	2	1	0	Default Value
0xFD	PIE9	PIE8	-	-	-	P4PU[2:0]			11- - -000
	R/W	R/W	-	-	-	R/W	R/W	R/W	

Port 1.4 input enable
0: Disable
1: Enable

Port 1.3 input enable
0: Disable
1: Enable

Not used

Port 4 pin 2/1/0 Pull-up Enable Bit
0: Disable 1: Enable

Register 7-16: PIE – Port Input Enable Register

PIE
Port Input Enable Register

Address	7	6	5	4	3	2	1	0	Default Value
0xA4	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0	1111 1111
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Port 1.2 Input Enable
0: Disable 1: Enable

Port 1.1 Input Enable
0: Disable 1: Enable

Port 1.0 Input Enable
0: Disable 1: Enable

Port 0.4 Input Enable
0: Disable 1: Enable

Port 0.3 Input Enable
0: Disable 1: Enable

Port 0.2 Input Enable
0: Disable 1: Enable

Port 0.1 Input Enable
0: Disable 1: Enable

Port 0.0 Input Enable
0: Disable 1: Enable

Register 7-17: WKPND – Port Wakeup Pending Register

WKPND
Port Wakeup Pending Register

Address	7	6	5	4	3	2	1	0	Default Value
0x9A	-	-	-	-	WPD P	WPP 07	WPP 06	WPP 05	xxxx xxxx
	-	-	-	-	R/W	R/W	R/W	R/W	

Not used

USBDP Wakeup Pending
0: No w akeup / Clear pending
1: Pending

Port0.7 Wakeup Pending
0: No w akeup / Clear pending
1: Pending

Port0.6 Wakeup Pending
0: No w akeup / Clear pending
1: Pending

Port0.5 Wakeup Pending
0:No w akeup / Clear pending
1: Pending

Register 7-18: WKEN – Port Wakeup Enable Register

WKEN	Address	7	6	5	4	3	2	1	0	Default Value
Port Wakeup Enable Register	0x9B	-	-	-	-	WE DP	WEP 07	WEP 06	WEP 05	- - - - 1111
		-	-	-	-	R/W	R/W	R/W	R/W	

Not used

USBDP Wakeup Enable
0: Enable 1: Disable

Port0.7 Wakeup Enable
0: Enable 1: Disable

Port0.6 Wakeup Enable
0: Enable 1: Disable

Port0.5 Wakeup Enable
0: Enable 1: Disable

Register 7-19: WKEDG – Port Wakeup Edge Register

WKEDG	Address	7	6	5	4	3	2	1	0	Default Value
Port Wakeup Edge Register	0x9C	-	-	-	-	WED DP	WEP 07	WEP 06	WEP 05	- - - - 1111
		-	-	-	-	R/W	R/W	R/W	R/W	

Not used

USBDP Wakeup Edge
0:Rising edge
1:Falling edge

Port0.7 Wakeup Edge
0:Rising edge
1:Falling edge

Port0.6 Wakeup Edge
0:Rising edge
1:Falling edge

Port0.5 Wakeup Edge
0:Rising edge
1:Falling edge

7.2 Interrupt and Wakeup

There are 4 dedicated circuits associating with P0.5 – P0.7 and WDDP for capturing transitions at the pad. When an interested transition (defined in WKEDG) appears at the pad, these circuits generate a port wakeup interrupt pending. If the corresponding interrupt is enabled (IP[5] = 1), the interrupt request is asserted immediately to notify CPU to take action. All these circuits share the same interrupt service entry, therefore interpretation inside ISR is needed to determine the triggered circuit.

As these circuits are designed to work without clock, they are used not only for wakeup signal of idle mode, but also wakeup signal of halt mode and power down mode. For details, please refer to Section 6.3 Power Management.

These circuits are controlled by port wakeup control register, WKEN. The detecting event is defined in WKEDG. The corresponding interrupt pending flags are gathered in register WKPND.

To enable the circuit, there is an initialization procedure. User has to disable the interrupt first, and then do the setting. Before enabling the interrupt, the interrupt pending must be cleared, because false triggering is possibly happened when changing the circuit setting. Initialization must be redone in every change of setting.

7.3 Operation Guide

1. Setting Port Direction

The direction of the ports is defined in registers PxDIR. When a pin is set as output, its pull-up resistor is disabled automatically to avoid leakage current. Wrong operations like: (1) reading from an output pin results in reading out '0' only; and (2) writing to an input pin does not change the voltage level of the pad, only the corresponding port data register is changed

2. Reading data from and Writing data to Port

Each port of AX206 has one set of registers to manage output and input of the port. When the port is set to be output, the value of the output register Px will be reflected by the logic level of the pad.

When the port is set to be input, the logic level of the pad will be fed to the port synchronizers, and the synchronized value will be available to be read out from the input register.

3. Using Pull-up Registers

Each port pin associates with a 20Kohm pull-up resistor. The pull-up is disabled by default and it is enabled through register PxPU. To get rid of current leaking through the pull-up resistors, the pull-up is disabled automatically when the pin is set as output in push-pull mode and the pin outputs a low voltage level in open-drain mode. When the disabling condition is over, the pull-up setting is recovered.

4. Configure for Analog Input

To configure a port pin for analog peripherals, the first step is to disable the digital buffers in order to reduce interference to the analog input.

1. Set the pin to input mode

After that, the digital buffers of the pin are completely off. At this moment, enable the analog channel by:

2. Write '0' to particular analog channel enable bit in PIE

And the the channel will be isolated from internal logic at the same time, which gets rid of current leaking.

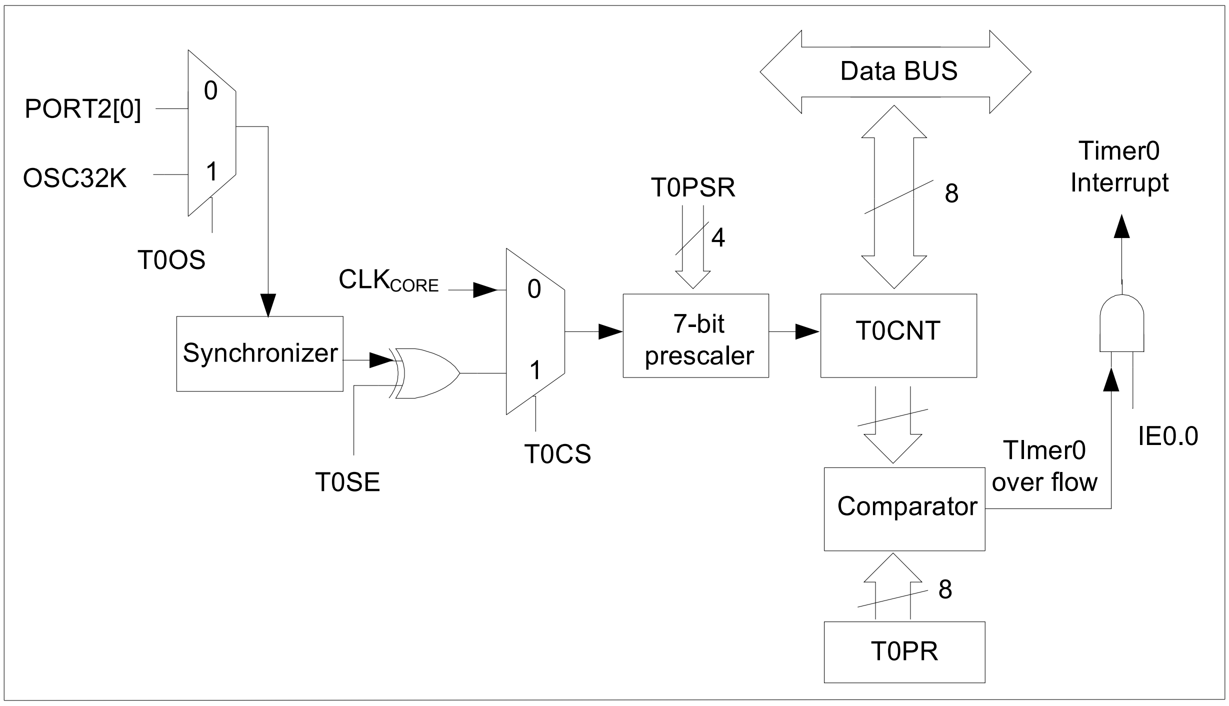
8 Multi-Function Timer

AX206 includes Timer, Timer0, Timer1, Timer2, RTCC and Watchdog Timer. Every Timer has its special function. User can use different timer for different purpose.

8.1 Timer0

Timer0 is an 8-bit timer/counter, with a 8-bit prescaler. Figure 8-1 shows the block diagram of the Timer0 module.

Figure 8-1: Block diagram of Timer0



8.1.1 Timer0 Registers

Register 8-1: TMR0CON – Timer0 Control Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR0CON Timer 0 Control Register	0xB1	T0PND	-	-	-	T0OS	T0CS	T0SE	T0ON	0- - - 0000
		R	-	-	-	R/W	R/W	R/W	R/W	

Timer Pending Flag
0: No pending
1: Pending

Not used

External Clock Select Bit
0: Select P2.1 as external clock
1: Select RTC as external clock

Clock mode select
0: Select system clock to count
1: Select external clock to count

Timer Enable Select Bit
0: Disable
1: Enable

Timer Clock Source Edge Select Bit
0: Increment on rising edge on external input clock pin
1: Increment on falling edge on external input clock pin

Register 8-2: TMR0CNT – Timer0 Counter Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR0CNT Timer 0 Counter Register	0xB3	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

Timer counter

Note:

When Timer0 is on (T0ON = 1), TMR0CNT will increment every time counter increment condition is met. When TMR0CNT = TMR0PR, TMR0CNT will be cleared to 00h automatically and will continue to be incremented, and Timer0 will overflow.

Register 8-3: TMR0PR – Timer0 Period Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR0PR Timer 0 Period Register	0xB4	W	W	W	W	W	W	W	W	1111 1111

Timer period

Register 8-4: TMR0PSR – Timer0 Prescaler Register

TMR0PSR	Address	7	6	5	4	3	2	1	0	Default Value
Timer 0 Prescaler Register	0xB5	-	-	-	-	-	T0PSR			- - - - -xxx
		-	-	-	-	-	W	W	W	

Bits 7–3: **Not used**

Bits 2–0: **Timer0 Prescaler divide rate**

000:	1:1
001:	1:2
010:	1:4
011:	1:8
100:	1:16
101:	1:32
110:	1:64
111:	1:128

8.1.2 Operation Modes

There are two operation modes:

1. Timer Mode

   Timer mode is selected by clearing bit T0CS (TMR0CON.2). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler).

2. Counter Mode

   Counter mode is selected by setting bit T0CS (TMR0CON.2). In counter mode, Timer0 will increment either on every rising or falling edge of pin PORT2[1]/OSC32K. Timer0 Clock Source Edge Select bit T0SE determines the edge. Clearing the bit T0SE selects the rising edge. Setting the bit T0SE selects the falling edge. Clearing the bit T0OS (TMR0CON.3) selects Port2.1 as external clock input. Setting the bit T0OS selects OSC32K input.

8.1.3 Interrupt

Please refer to Chapter 5 Interrupt Processing.

8.1.4 Operation Guide

1. Select Timer0 Prescaler divide rate (TMR0PSR[2:0])
2. Initialize Timer0 Counter Register (TMR0CNT)
3. Set Timer0 Period Register (TMR0PR)
4. Configure TMR0CON register by
   a) Select the Clock mode T0CS (TMR0CON[2])
   b) Select External Clock if external clock has been selected to count T0OS (TMR0CON[3])
   c) Select Timer Clock Source Edge if external clock has been selected to count T0SE (TMR0CON[1])
   d) EnableT0ON (TMR0CON[0])

5. Enable T0IE (IE.0) and EA (IE.7)

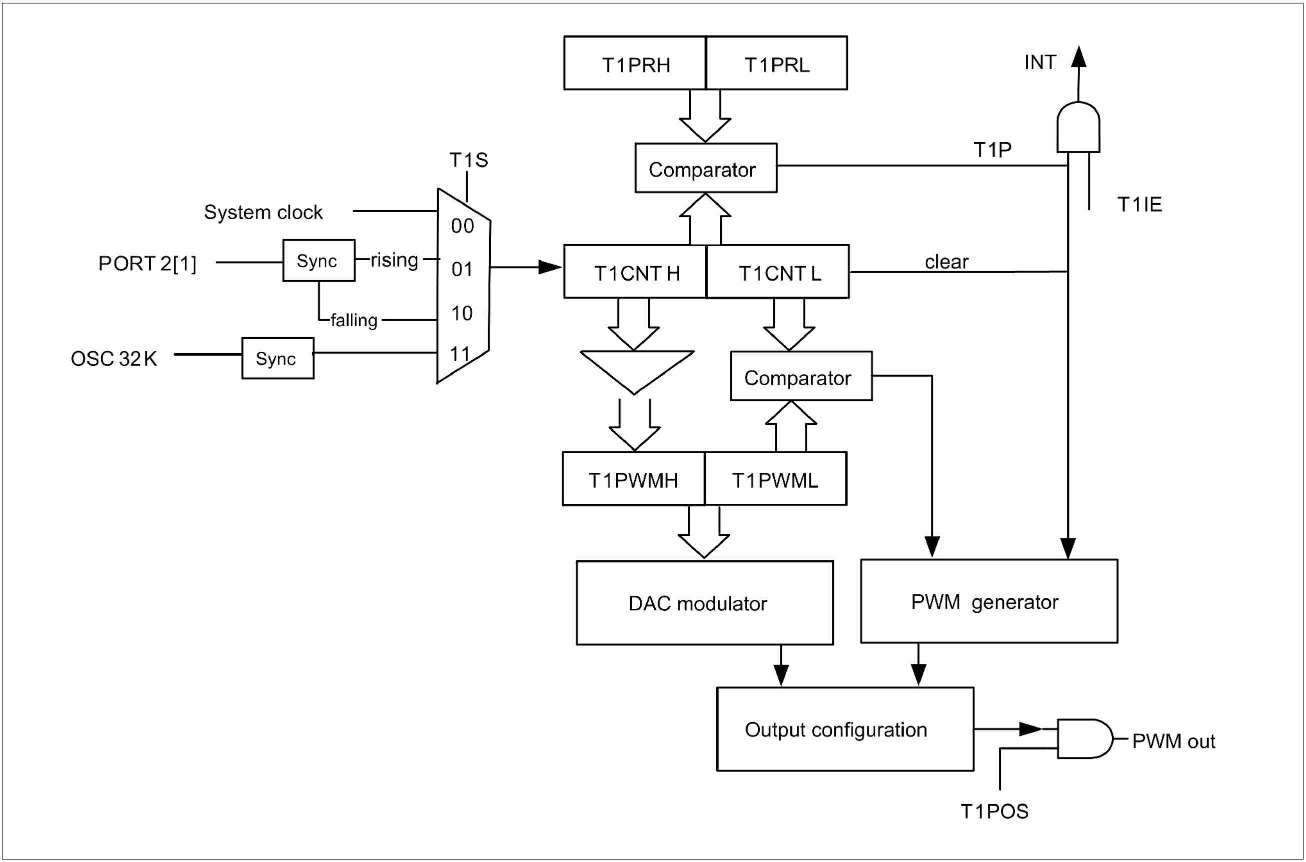
Example: Timer0 ISR code

```
CSEG   AT     0x1003

Timer0_ISR:

......

RETI
```

8.2 Timer1

Timer1 is a 16-bit timer/counter with a 7-bit prescaler. It can be configured as a timer, a counter, a PWM generator or a DAC. Figure 8-2 shows the block diagram of Timer1 module.

Figure 8-2: **Timer1 Block Diagram**



8.2.1 Timer1 Registers

Register 8-5: TMR1CON – Timer1 Control Register

TMR1CON	Address	7	6	5	4	3	2	1	0	Default Value
Timer 1 Control Register	0xE1	T1POS2	T1POS1	T1S		-	T1OD	T1POEN	T1ON	0000 0000
		R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	

Timer PWM output Select P4.0
0: Not select 1: Select P4.0

Timer PWM output Select P2.3
0: Not select 1: Select P2.3

Clock mode select
00: Select system clock to count
01: Select external pin rising to count
10: Select external pin falling to count
11: Select RTC to count

Not used

PWM open-drain output
0: Disable 1: Enable

PWM output enable select bit
0: Disable
1: Enable

Timer Enable Select Bit
0: Disable
1: Enable

Register 8-6: TMR1CNTL – Timer1 Counter Low Byte Register

TMR1CNTL	Address	7	6	5	4	3	2	1	0	Default Value
Timer 1 Counter Low Byte Register	0xE2	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

Timer counter

Register 8-7: TMR1CNTH – Timer1 Counter High Byte Register

TMR1CNTH	Address	7	6	5	4	3	2	1	0	Default Value
Timer 1 Counter High Byte Register	0xE3	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

Timer counter

Note:

Timer1 16-bit counter register is formed by {TMR1CNTH,TMR1CNTL}. Timer1 will increase in proper condition after Timer1 is turned on. TMR1CNT will be cleared to 0x0000 when TMR1CNT = TMR1PR. Also, Timer1 will overflow when TMR1CNT goes from TMR1PR to 0x0000.

Register 8-8: TMR1PERL – Timer 1 Period Low Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR1PERL										
Timer 1 Period Low Byte Register	0xE4	W	W	W	W	W	W	W	W	xxxx xxxx
		Timer period								

Register 8-9: TMR1PERH – Timer 1 Period High Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR1PERH										
Timer 1 Period High Byte Register	0xE5	W	W	W	W	W	W	W	W	xxxx xxxx
		Timer period								

Register 8-10: TMR1PWML – Timer 1 PWM Low Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR1PWML										
Timer 1 PWM Low Byte Register	0xE6	W	W	W	W	W	W	W	W	xxxx xxxx
		PWM								

Register 8-11: TMR1PWMH – Timer 1 PWM High Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
TMR1PWMH										
Timer 1 PWM High Byte Register	0xE7	W	W	W	W	W	W	W	W	xxxx xxxx
		PWM								

Note:
(TMR1PWMH,TMR1PWML) are used as duty cycle setting.

8.2.2 Timer1 Operation Mode

There are 3 operation modes in Timer1:

1. Timer Mode

   Timer mode is selected by clearing bits T1S (TMR1CON[5:4]). In Timer mode, the Timer1 will increment every instruction cycle in Timer Mode.

2. Counter Mode

   Counter Mode is selected by setting T1S (TMR1CON[5:4]). Timer1 will increment either on every rising and falling edge of pin PORT2[0] or rising edge of OSC32K.

3. PWM Mode

   In PWM mode, timer1 is used as a PWM generator. Write data to TMR1PWMH/TMR1PWML as duty cycle, and TMR1PRH/L as Period.

8.2.3 Interrupt

Please refer to Chapter 5 Interrupt Processing.

8.2.4 Operation Guide

1. Initialize Timer1 Counter Registers (TMR1CNTL and TMR1CNTH)
2. Set Timer1 Period Registers (TMR1PERL and TMR1PERH)
3. Configure Timer1 PWM register if PWM mode is used (TMR1PWML and TMR1PWMH)
4. Configure TMR1CON register by
   a) Select Clock mode T1S (TMR1CON[5:4])
   b) Select PWM output if PWM mode is used T1POS2 and T1POS1 (TMR1CON[7:6])
   c) Select PWM open-drain output if PWM mode is used (TMR1CON[2]). The bit is used especially for Digital Photo Frame application.
   d) Enable T1ON (TMR1CON[0])
   e) Enable T1POEN (TMR1CON[1])
5. Enable T1IE (IE.1) and EA (IE.7)

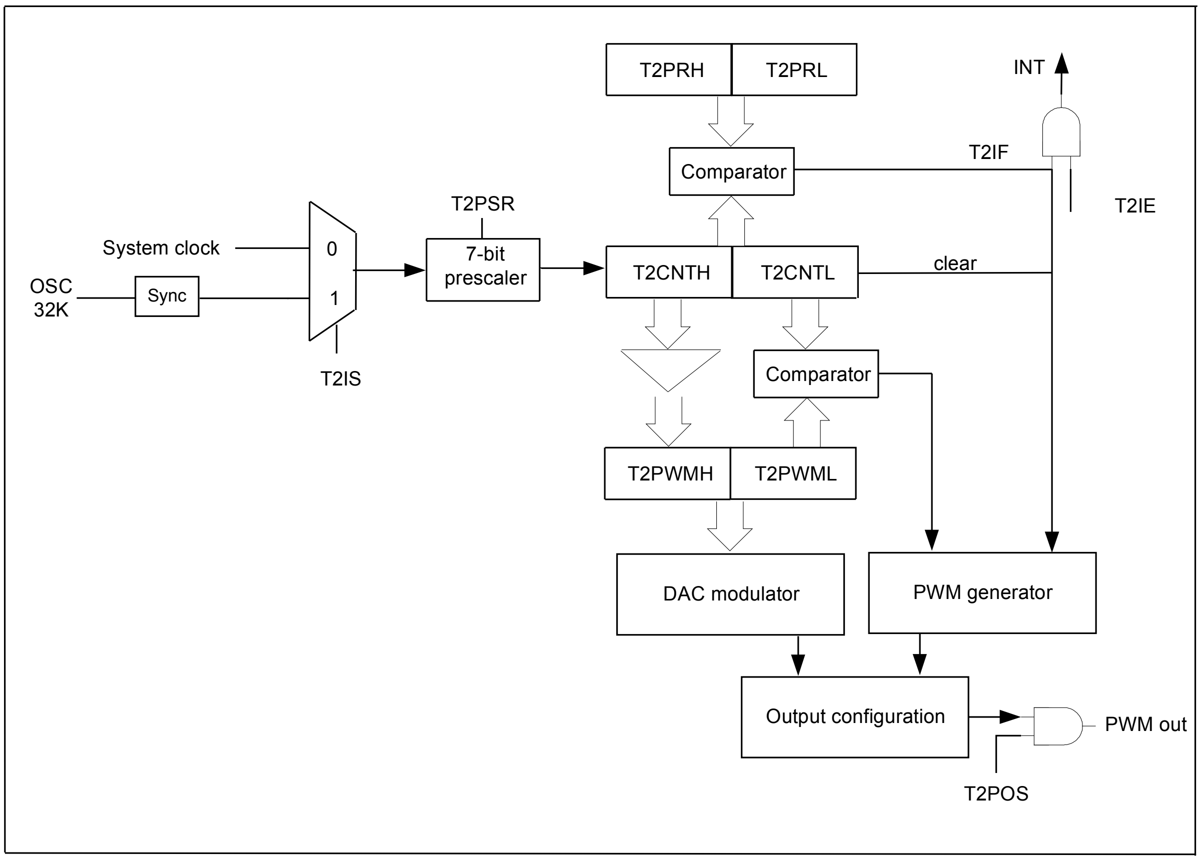
Example: Timer1 ISR code

```
CSEG  AT    0x100B
Timer1_ISR:
......
RETI
```

8.3 Timer2

Timer2 is a 16-bit timer/counter with a 7-bit prescaler. It can be configured as a timer, a counter, a PWM generator or a DAC. Figure 8-3 shows the block diagram of Timer2 module.

Figure 8-3: Timer2 Block Diagram



8.3.1 Timer2 Register

Register 8-12: TMR2CON – Timer2 Control Register

TMR2CON	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 Control Register	0xC1	T2POS2	T2POS1	T2S	T2POEN	T2PSR			T2ON	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Timer PWM output Select P4.1
0: Not select 1: Select

Timer PWM output Select P2.6
0: Not select 1: Select

Clock source select
0: Select system clock as clock source
1: Select RTC as clock source

PWM output enable select bit
0: Disable 1: Enable

Timer Enable Select Bit
0: Disable 1: Enable

Timer Prescaler select
000: 1
001: 2
010: 4
011: 8
100:16
101: 32
110: 64
111: 128

Register 8-13: TMR2CNTL – Timer2 Counter Low Byte Register

TMR2CNTL	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 Counter Low Byte Register	0xC2	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

Timer counter

Register 8-14: TMR2CNTH – Timer2 Counter High Byte Register

TMR2CNTH	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 Counter High Byte Register	0xC3	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

Timer counter

Register 8-15: TMR2PERL – Timer 2 Period Low Byte Register

TMR2PERL	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 Period Low Byte Register	0xC4	W	W	W	W	W	W	W	W	xxxx xxxx
		Timer period								

Register 8-16: TMR2PERH – Timer2 Period High Byte Register

TMR2PERH	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 Period High Byte Register	0xC5	W	W	W	W	W	W	W	W	xxxx xxxx
		Timer period								

Register 8-17: TMR2PWML – Timer2 PWM Low Byte Register

TMR2PWML	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 PWM Low Byte Register	0xC6	W	W	W	W	W	W	W	W	xxxx xxxx
		PWM								

Register 8-18: TMR2PWMH – Timer2 PWM High Byte Register

TMR2PWMH	Address	7	6	5	4	3	2	1	0	Default Value
Timer 2 PWM High Byte Register	0xC7	W	W	W	W	W	W	W	W	xxxx xxxx
		PWM								

Note:
TMR2PWMH and TMR2PWML are used as duty cycle setting.

8.3.2 Operation Mode

There are 2 Operation Modes in Timer2.

1. Timer Mode

   Timer mode is selected by clearing bits T2S(T2S=0). In Timer mode, the Timer2 will increment every instruction cycle in Timer Mode.

2. PWM Mode

   In PWM mode, timer2 is used as a PWM generator. Write the data to TMR2PWMH/TMR2PWML as duty cycle, and TMR2PERH/TMR2PERL as Period.

8.3.3 Interrupt

Please refer to Chapter 5 Interrupt Processing.

8.3.4 Operation Guide

1. Initialize Timer2 Counter Registers (TMR2CNTL, TMR2CNTH)
2. Set Timer2 Period Register (TMR2PERL, TMR2PERH)
3. Configure Timer2 PWM Registers if PWM mode is used (TMR2PWML, TMR2PWMH)
4. Configure TMR2CON register by
   a) Select Timer2 Prescaler divide rate T2PSR (TMR2CON[3:1])
   b) Select the Clock source (TMR2CON[5])
   c) Select PWM output if PWM mode is used (TMR2CON[7:6])
   d) Enable T2ON (TMR2CON[0])
   e) EnableT2PO (TMR2CON[4])
5. Enable T2IE (IE.2) and EA (IE.7)

Example: Timer2 ISR code

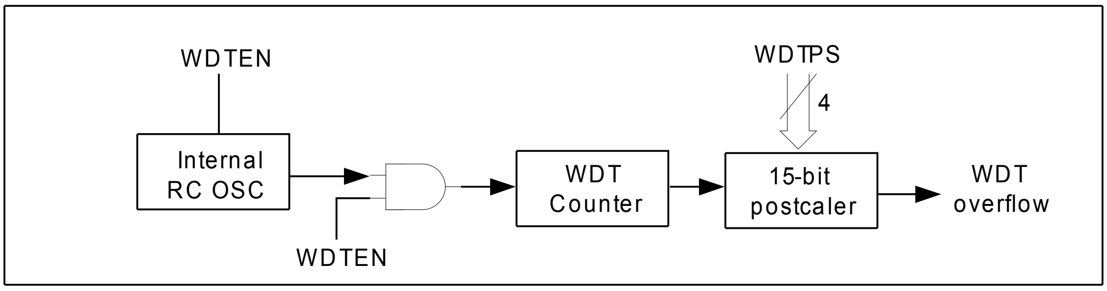
```
CSEG  AT      0x1013
Timer2_ISR:
……
RETI
```

8.4 Watchdog Timer with On-chip 16kHz RC oscillator

The Watchdog Timer (WDT) logic consists of a Watchdog Timer, 8-bit counter and a 15-bit programmable postscaler. The Watchdog Timer is clocked by its own internal RC oscillator running at 16KHz. When device resets, the WDT is disabled and user should enable the WDT if it is needed.

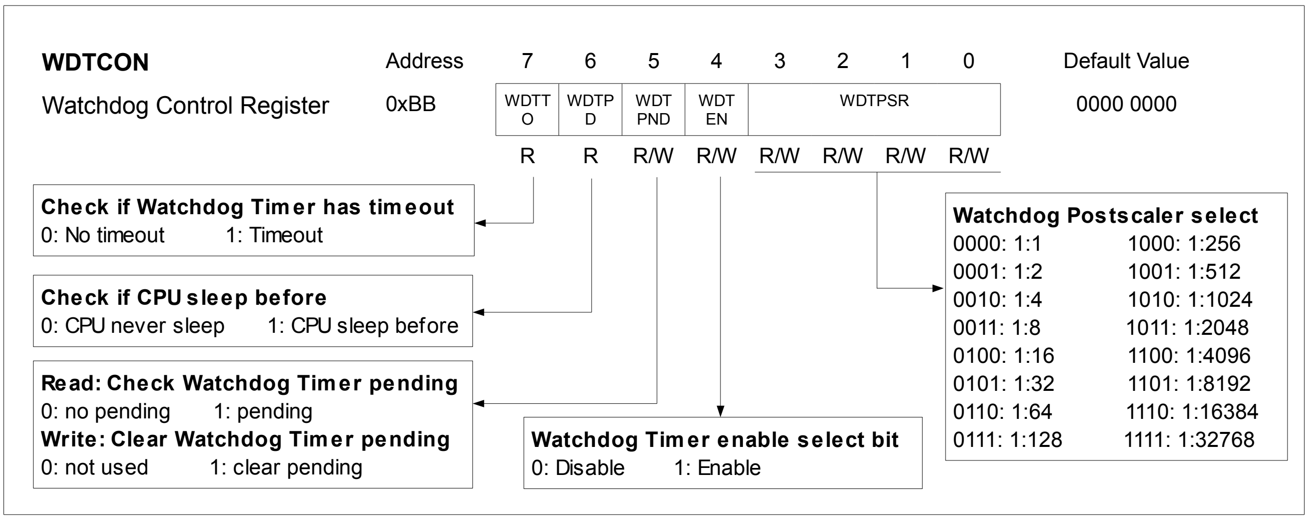
In the default configuration (postscaler divide ratio set to 1:1), the 8-bit counter counts from 00h to FFh in 16 milliseconds. The application program needs to write a “0” into WDTCON[5] at least 16 milliseconds to prevent a watchdog timeout reset. The lower four bits of the WDTCON register control the selection of divide ratio. Figure 8-4 shows the WDT block diagram.

Figure 8-4: **WDT block diagram**



8.4.1 Watchdog Timer Register

Register 8-19: WDTCON – Watchdog Control Register



WDTCON	Address	7	6	5	4	3	2	1	0	Default Value
Watchdog Control Register	0xBB	WDTTO	WDTPD	WDT PND	WDT EN	WDTPSR				0000 0000
		R	R	R/W	R/W	R/W	R/W	R/W	R/W	

Check if Watchdog Timer has timeout
0: No timeout 1: Timeout

Check if CPU sleep before
0: CPU never sleep 1: CPU sleep before

Read: Check Watchdog Timer pending
0: no pending 1: pending
Write: Clear Watchdog Timer pending
0: not used 1: clear pending

Watchdog Timer enable select bit
0: Disable 1: Enable

Watchdog Postscaler select

0000: 1:1	1000: 1:256
0001: 1:2	1001: 1:512
0010: 1:4	1010: 1:1024
0011: 1:8	1011: 1:2048
0100: 1:16	1100: 1:4096
0101: 1:32	1101: 1:8192
0110: 1:64	1110: 1:16384
0111: 1:128	1111: 1:32768

8.4.2 Operation Mode

There are 2 modes for wakeup operation: wakeup without reset and reset wakeup. It is determined by WDTIE bit (IP0.7). When WDTIE sets to 1, the watchdog will generate a non-reset wakeup after counter overflows. Only in STOP CLOCK Mode, non-reset wakeup can wakeup MCU and MCU will continue to execute next instruction. MCU can not be waken up in SLEEP Mode. When WDTIE sets to 0, the watchdog will generate a reset wakeup after counter overflows. Both in HOLD Mode and SLEEP Mode, watchdog can wakeup MCU and MCU will reset and come back to the initial state.

8.5 RTCC Timer

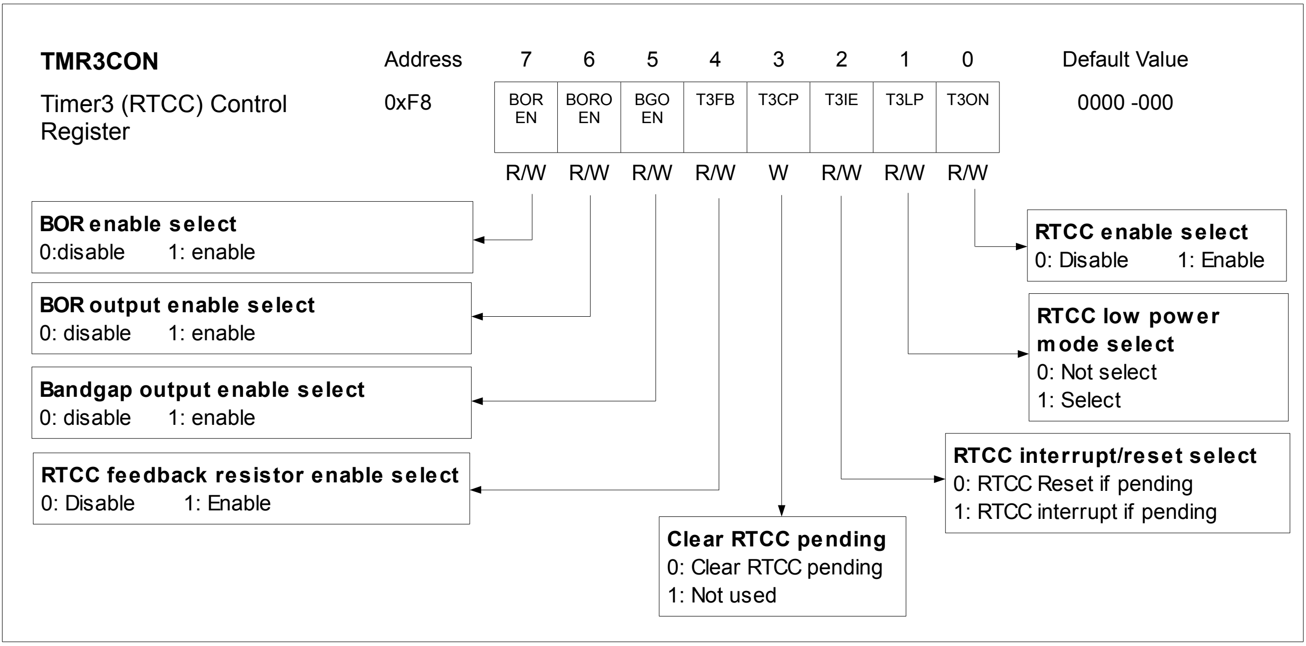
Real time clock module provides regular periodic interrupts based on 32,768 Hz clock. It can be used for generating software real time clock or low-frequency interrupt that cannot be handled by Timer modules.

The clock source for real time clock module comes from 32,768 Hz oscillator. User should refer to section 6.1.3 for more details on enabling or disabling 32,768 Hz clock.

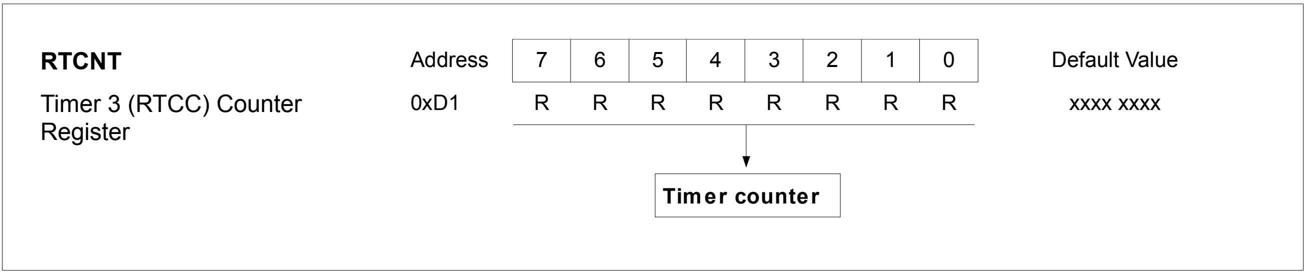
Real time clock interrupt can be enabled by writing 1 to T3IE (TMR3CON.2) bit. The internal 15-bit counter will be incremented due to real time clock source (when T3ON=1). When the counter overflows, interrupt pending flag RTPND will be set to 1. RTPND can be cleared by software by writing 0 to T3CP bit. The internal counter can generate real time clock interrupt every 1 second.

8.5.1 RTCC Timer Register

Register 8-20: TMR3CON – Timer3 (RTCC) Control Register



Register 8-21: RTCNT – Timer3 (RTCC) Counter Register



8.5.2 Operation Mode

RTCC can run in 2 modes:

1. RTCC generates a pending in every second exactly if RTCC's interrupt enable (T3IE) has been set.
2. RTCC will wakeup MCU at the end of one second if MCU runs in Stop Clock Mode no matter what T3IE is.

RTCC also has one special mode, Low Power Mode, which can save power. It is disabled by default. It can be enabled by setting TMR3CON[1].

8.5.3 Operation Guide

1. Configure TMR3CON register by
    a) Enable RTCC feedback resistor enable bit T3FB (TMR3CON[4])
    b) Select RTCC interrupt/reset T3IE (TMR3CON[2])
    c) Enable RTCC low power mode T3LP (TMR3CON[1])
    d) Enable T3ON (TMR3CON[0])
2. Set RTCC interrupt enable bit RAW IE (IE[6]) and EA (IE[7])

Example: RTCC ISR code

```
CSEG  AT      0x1033
RTCC_ISR:
Clr      TMR3CON.3                    ;clr RTCC pending
……
RETI
```

9 Serial Protocol Interface (SPI)

SPI supports two modes: master mode and slave mode.

SPI Module uses 2 pins for two-pin mode: P1.5, P1.7

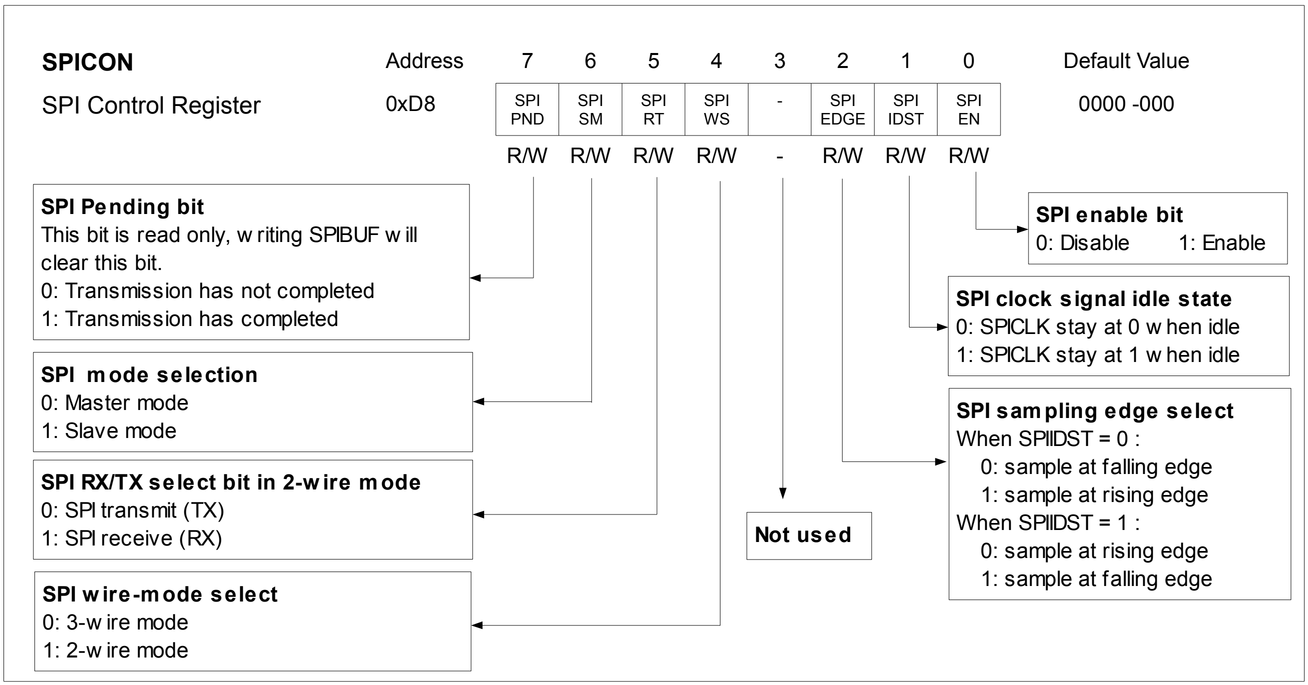
- Serial Data (SPIDIDO) - P1.5
- Serial Clock (SPICLK) - P1.7

SPI Module uses 3 pins for three-pin mode: P1.5, P1.6, P1.7

- Serial Data Out (SPIDO) - P1.6
- Serial Data In (SPIDI) - P1.5
- Serial Clock (SPICLK) - P1.7

9.4 SPI Registers

Register 9-1: SPICON – SPI Control Register



	Address	7	6	5	4	3	2	1	0	Default Value
SPICON SPI Control Register	0xD8	SPI PND	SPI SM	SPI RT	SPI WS	-	SPI EDGE	SPI IDST	SPI EN	0000 -000
		R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	

SPI Pending bit
This bit is read only, writing SPIBUF will clear this bit.
0: Transmission has not completed
1: Transmission has completed

SPI mode selection
0: Master mode
1: Slave mode

SPI RX/TX select bit in 2-wire mode
0: SPI transmit (TX)
1: SPI receive (RX)

SPI wire-mode select
0: 3-wire mode
1: 2-wire mode

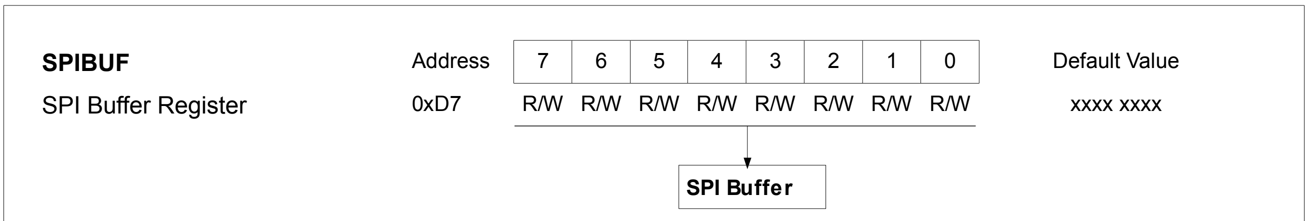
Not used

SPI sampling edge select
When SPIIDST = 0 :
0: sample at falling edge
1: sample at rising edge
When SPIIDST = 1 :
0: sample at rising edge
1: sample at falling edge

SPI clock signal idle state
0: SPICLK stay at 0 when idle
1: SPICLK stay at 1 when idle

SPI enable bit
0: Disable 1: Enable

Register 9-2: SPIBUF – SPI Buffer Register



	Address	7	6	5	4	3	2	1	0	Default Value
SPIBUF SPI Buffer Register	0xD7	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

SPI Buffer

Note: Write to SPIBUF will clear SPI Pending.

9.5 Baudrate Setting

SPI has a dedicated baudrate generator. It is controlled by register SPIBAUD, and it determines the output clock in master mode as:

$$Baudrate = \frac{F_{\text{system\_clock}}}{2\,x\,(SPIBAUD + 1)}$$

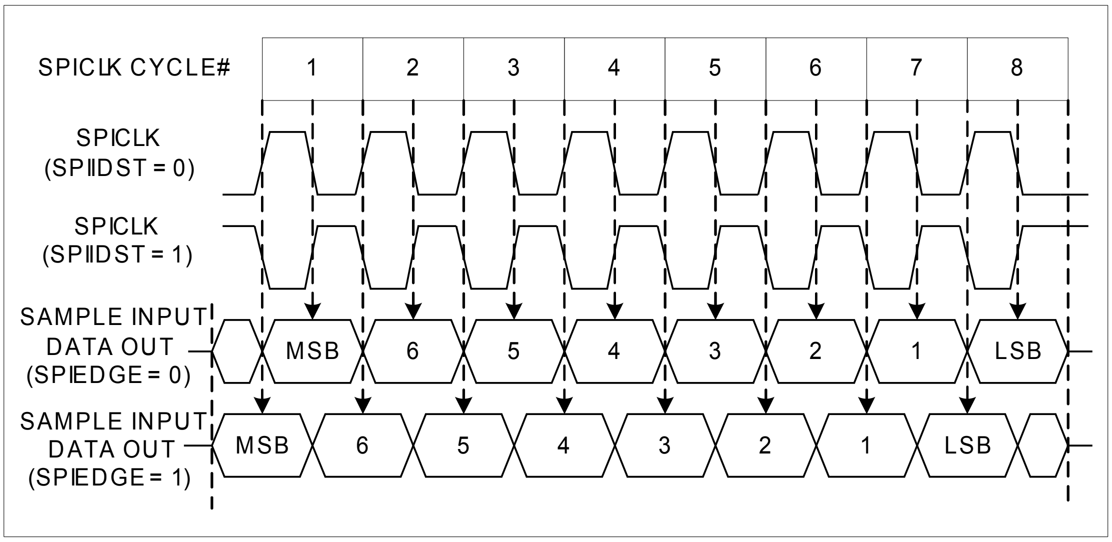
Register 9-3: SPIBAUD – SPI Baudrate Register

SPIBAUD	Address	7	6	5	4	3	2	1	0	Default Value
SPI Baud Rate Register	0xD6	W	W	W	W	W	W	W	W	xxxx xxxx
		SPI Baudrate								

9.6 Operation Mode

Figure 9-4 shows SPI signal timing waveform.

Figure 9-4: SPI Signal Timing Wave Form



9.7 Operation Guide

1. Set P1.5 as output in transmission and as input in reception, set P1.7 as output in master mode and as input in slave mode, P1.6 is not used in 2-wire mode
2. Select SPIRT in 2-wire mode if 2-wire mode is selected
3. Select master mode or slave mode in SPISM
4. Configure clock frequency if master mode is selected in step 3
5. Select one of the four timing mode (refer to Figure 9-4)
6. Enable SPI module by setting SPIEN to ‘1’
7. Set SPIIE ‘1’ if needed (IE.4)
8. Write data to SPIBUF to kick-start a process
9. Wait for SPIPND change to ‘1’, or wait for interrupt
10. Read received data from SPIBUF if needed
11. Go to Step 8 to start another process if needed or turn off SPI by clearing SPIIE (IE.4) and SPIEN (SPICON.0)

10 USB Device Controller and PHY

AX206 has a built-in USB Device Controller (UDC), which can interpret the USB Standard Command. The Serial Interface Engine (SIE) inside the USB controller handles NRZI encoding/decoding, bit stuffing/unstuffing, and CRC generation/checking, so that it will not trade-off the speed performance of the device. The program developer, therefore, can spend less programming effort in dealing with USB transaction.

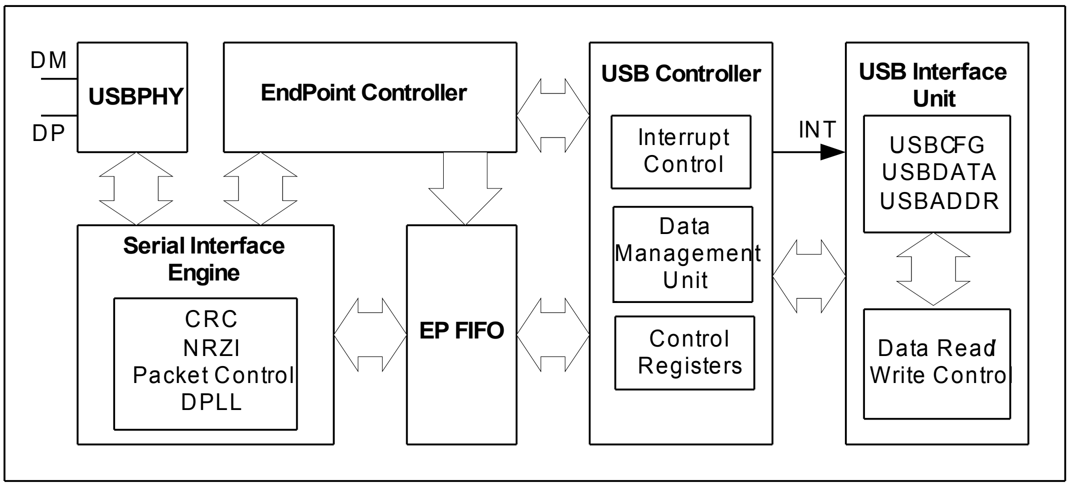
The USB module complies with the USB 2.0 specifications for full-speed (12 Mbps) functions. The device comes with a built-in USB PHY (no need external power supply) and no additional off-chip components are required for USB interface. The dedicated package pins, USBDP and USBDM, can be directly connected to the USB signal wires, D+ and D-, respectively.

AX206 supports 2 endpoints, EP0 for control, EP1 endpoint (IN/OUT) for data transmission. Every endpoint supports three modes: Interrupt mode, Bulk mode and Isochronous mode. All endpoints total 144 bytes FIFO buffer sharing with XDATA (see Figure 3-1). Table 10-1 shows the endpoints buffer size.

Table 10-1: Endpoints Buffer Size

Endpoint	Input FIFO	Output FIFO
EP0	16 bytes (shared)	
EP1	64 bytes	64 bytes

Figure 10-1: Built-in USB controller and interface



10.1 Registers

Register 10-1: USBCON0 – USB Control 0 Register

	Address	7	6	5	4	3	2	1	0	Default Value
USBCON0 USB Control 0 Register	0xC8	SOF PND	USB DONE	USB RST	PHY EN	DPPU	USB RNW	-	USB KS	0000 0000
		R/W	R	R/W	R/W	R/W	R/W	-	W	

Read: USB SOF Pending bit
0: Not pending
1: Pending
Write: Clear USB SOF Pending bit
0: Clear SOF pending
1: not used

USB read/write done bit
0: RD/WR not finished
1: RD/WR finished

Reset USB
0: Not reset 1: Reset

PHY enable select
0: Disable 1: Enable

Kick start USB read/Write
0: Not used
1: Kickstart

Not used

USB Read/Write operation select
0: Write operation
1: Read operation

DP PU enable select
0: Disable 1: Enable

Note: USBRST and DPPU can only be reset by POR or MCLR.

Register 10-2: USBBUF – USB Buffer Register

	Address	7	6	5	4	3	2	1	0	Default Value
USBBUF USB Buffer Register	0xC9	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx

USB Buffer

Register 10-3: USBADR – USB Address Register

	Address	7	6	5	4	3	2	1	0	Default Value
USBADR USB Address Register	0xCA	W	W	W	W	W	W	W	W	xxxx xxxx

USB Address

10.2 Operation Mode

10.2.1 Registers Read From/Writer to USB Controller

1. Read from USB Controller

   Reading data from USB controller is similar to writing data to it. Store the register address to USBADR from which data is going to be read. For a read operation, set USBRNW to '1'. Then, set USBKS to '1' to start the read operation. The data of the register, pointed by USBADR, will be stored in USBBUF when the read operation is complete. When read operation is kick-started, USBDONE will be cleared. User can poll the USBDONE bit and it will change to '1' when read operation is complete. Reading from most registers takes 2 CLK cycles and reading from FIFO registers take 5 CLK cycles.

2. Write to USB Controller

   USBADR and USBBUF registers must be used when reading data from or writing data to USB controller register. USBADR register stores the 6-bit address when reading from or writing to USB controller. USBBUF register stores the actual data to which USBADR is pointing.

   To perform a write operation, the address and data are stored to USBADR and USBBUF registers, respectively. For a write operation, clear USBRNW bit to '0'. Write operation to most of the registers takes two CLK cycles (USB clock). Writing to CSR0 takes 7 CLK cycles while writing to other CSR registers (InCSR1, OutCSR1, InCSR2, OutCSR2) takes 6 CLK cycles. The write operation will begin in background once it is kick-started by setting USBKS to '1'. USBKS is write-only bit. Reading it will get a '0'. When write operation is kick-started, USBDONE will be cleared. User can poll the USBDONE bit and it will change to '1' to indicate that write operation is complete.

10.2.2 Control The USB Module

The built-in USB module is controlled through the USB function registers, which is divided into four groups:

1. **Common USB control registers (addresses 00h to 0Fh):** These registers provide control and status information for the entire controller (see Table 10-2).
2. **Indexed registers (addresses 10h to 1Fh):** These registers provide control and status information for different endpoints (see Table 10-3). Endpoint can be selected by writing the endpoint number to the Index register (0Eh). So, to access the registers for IN Endpoint 1 and OUT Endpoint 1, 1 must first be written to the Index register: the corresponding control and status registers for the specified endpoint will then appear in the memory map.
3. **FIFO registers (addresses 20h to 21h):** The FIFO registers for the IN endpoints appear as single bytes, consecutively in the memory map starting at address 20h (see Table 10-4). The FIFO registers for the OUT endpoints also appear consecutively at the same set of addresses. Writing a byte to address 21h results in a byte being loaded into the FIFO buffer for IN Endpoint 1. Reading a byte from address 21h results in a byte being unloaded from the FIFO buffer for OUT Endpoint 1.

Table 10-2: Common USB control Registers

USBADDR	Name	Description
00h	FAddr	Function address register.
01h	Power	Power management register.
02h	IntrIn1	Interrupt register for Endpoint 0 plus IN Endpoints 1.

03h	RESERVED	
04h	IntrOut1	Interrupt register for OUT Endpoints 1 .
05h	RESERVED	
06h	IntrUSB	Interrupt register for common USB interrupts.
07h	IntrIn1E	Interrupt enable register for IntrIn1.
08h	RESERVED	
09h	IntrOut1E	Interrupt enable register for IntrOut1.
0Ah	RESERVED	
0Bh	IntrUSBE	Interrupt enable register for IntrUSB.
0Ch	Frame1	Frame number bits 0 to 7.
0Dh	Frame2	Frame number bits 8 to 10.
0Eh	Index	Index register for selecting the endpoint status and control registers.
0Fh	RESERVED	

Table 10-3: Indexed Endpoint control Registers

USBADDR	Name	Description
10h	InMaxP	Maximum packet size for IN endpoint. (Endpoints)
11h	CSR0	Control Status register for Endpoint 0. (Control Endpoint)
	InCSR1	Control Status register 1 for IN endpoint. (Endpoints 1)
12h	InCSR2	Control Status register 2 for IN endpoint. (Endpoints 1)
13h	OutMaxP	Maximum packet size for OUT endpoint. (Endpoints 1)
14h	OutCSR1	Control Status register 1 for OUT endpoint. (Endpoints 1)
15h	OutCSR2	Control Status register 2 for OUT endpoint. (Endpoints 1)
16h	Count0	Number of received bytes in Endpoint 0 FIFO. (Control Endpoint)
	OutCount1	Number of bytes in OUT endpoint FIFO (lower byte). (Endpoints 1)
17h	OutCount2	Number of bytes in OUT endpoint FIFO (upper byte). (Endpoints 1)

Table 10-4: Endpoint FIFO registers

USBADDR	Name	Description
20h	FIFO0	FIFO register of EP0
21h	FIFO1	FIFO register of EP1

Note:
For more details of USB function registers, please refer to Appendix II USB Function Register.

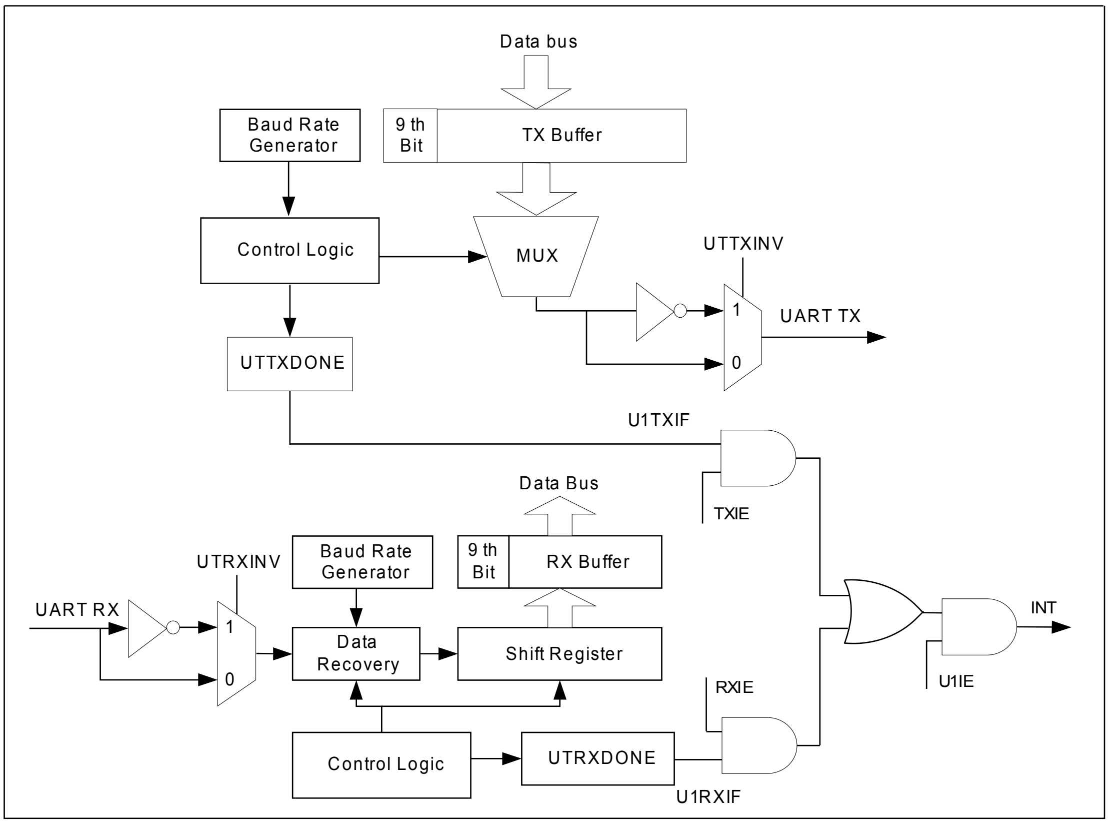
11 UART

UART is a serial port capable of asynchronous transmission. The UART can function in full duplex mode. Receive data is buffered in a holding register. This allows the UART to start reception of a second incoming data byte before software has finished reading the previous data byte.

- **Receive pin (U1RX) – P2.5/P4.0**
- **Transmit pin (U1TX) – P2.7/P4.1**

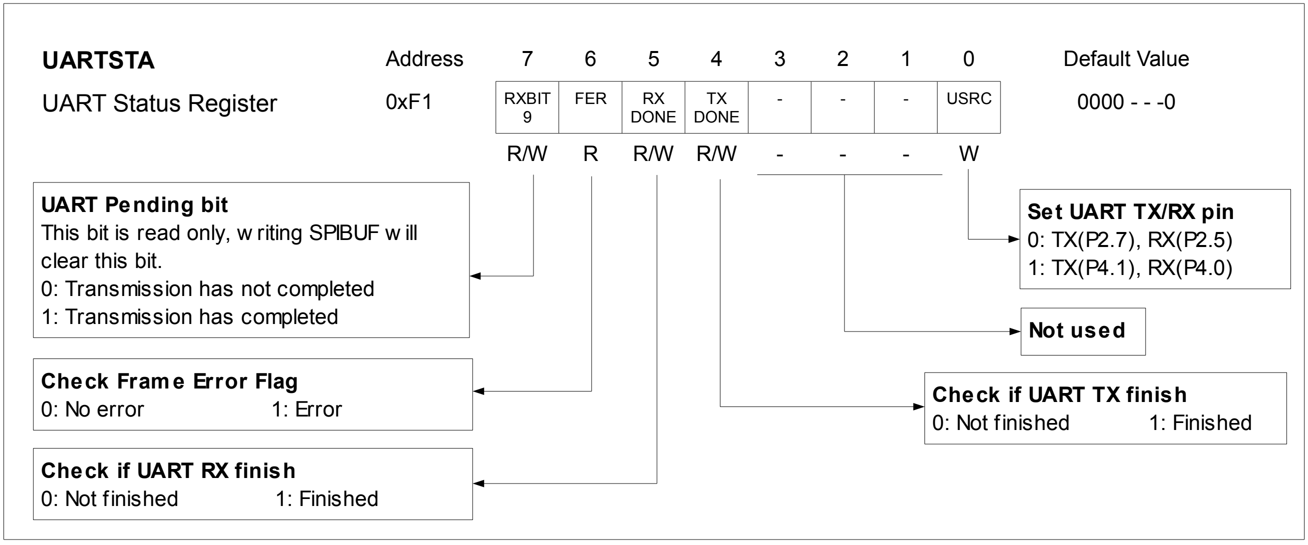
Figure 11-1 show UART block diagram.

Figure 11-1: UART block diagram



11.1 Registers

Register 11-1: UARTSTA – UART Status Register



	Address	7	6	5	4	3	2	1	0	Default Value
UARTSTA UART Status Register	0xF1	RXBIT9	FER	RX DONE	TX DONE	-	-	-	USRC	0000 - - -0
		R/W	R	R/W	R/W	-	-	-	W	

UART Pending bit
This bit is read only, writing SPIBUF will clear this bit.
0: Transmission has not completed
1: Transmission has completed

Check Frame Error Flag
0: No error 1: Error

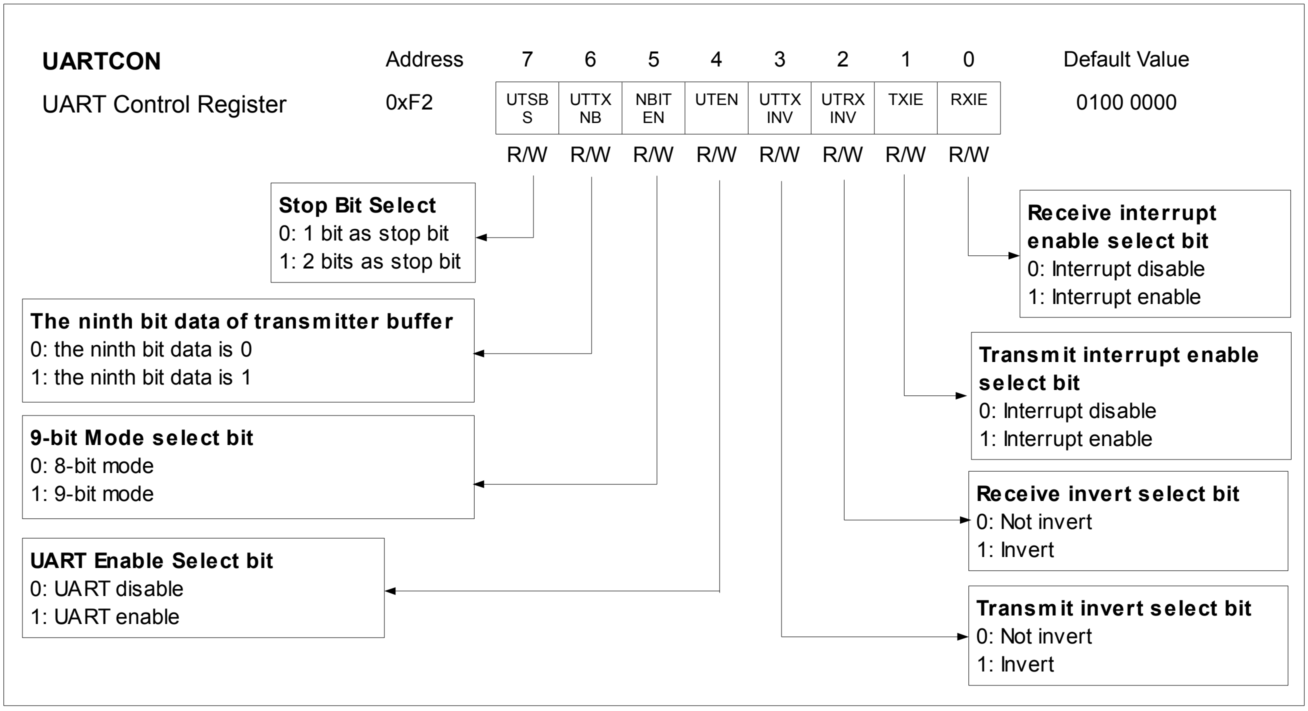
Check if UART RX finish
0: Not finished 1: Finished

Check if UART TX finish
0: Not finished 1: Finished

Not used

Set UART TX/RX pin
0: TX(P2.7), RX(P2.5)
1: TX(P4.1), RX(P4.0)

Register 11-2: UARTCON – UART Control Register



	Address	7	6	5	4	3	2	1	0	Default Value
UARTCON UART Control Register	0xF2	UTSBS	UTTXNB	NBITEN	UTEN	UTTXINV	UTRXINV	TXIE	RXIE	0100 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Stop Bit Select
0: 1 bit as stop bit
1: 2 bits as stop bit

The ninth bit data of transmitter buffer
0: the ninth bit data is 0
1: the ninth bit data is 1

9-bit Mode select bit
0: 8-bit mode
1: 9-bit mode

UART Enable Select bit
0: UART disable
1: UART enable

Transmit invert select bit
0: Not invert
1: Invert

Receive invert select bit
0: Not invert
1: Invert

Transmit interrupt enable select bit
0: Interrupt disable
1: Interrupt enable

Receive interrupt enable select bit
0: Interrupt disable
1: Interrupt enable

Register 11-3: UARTBAUD – UART Baudrate Register

	Address	7	6	5	4	3	2	1	0	Default Value
UARTBAUD UART Baudrate Register	0xF3	W	W	W	W	W	W	W	W	xxxx xxxx
		UART baudrate								

Register 11-4: UARTBUF – UART Buffer Register

	Address	7	6	5	4	3	2	1	0	Default Value
UARTBUF UART Buffer Register	0xF4	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	xxxx xxxx
		UART buffer								

11.2 Operation Guide

11.2.1 UART TX Operation Flow

1. Configure UARTBAUD. The allowed values are from 1 to 255.
2. Select 8-bit or 9-bit mode by NBITEN (UARTCON.5)
3. Select invert output or not by UTTXINV (UARTCON.3)
4. Set P2.7/P4.1 as output by USRC (UARTSTA.0)
5. Set TXIE to ‘1’ if needed
6. Set UTEN to ‘1’ to enable UART.
7. Write data to UARTBUF, it will clear TXDONE.
8. Wait for TXDONE change to ‘1’, or wait for interrupt.
9. Go to step 7 to start another TX if needed, or turn off UART by clearing TXIE and UTEN.

11.2.2 UART RX Operation Flow

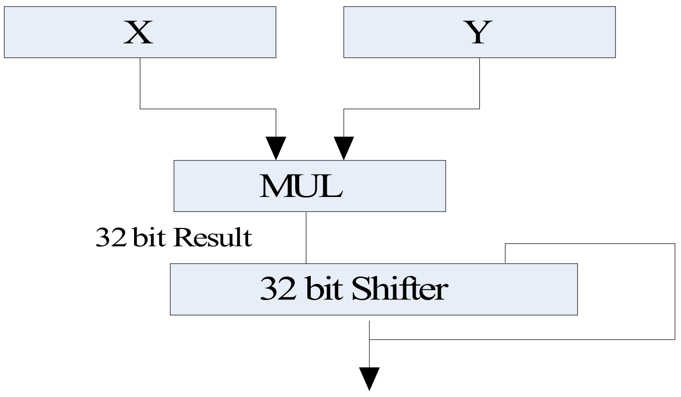
1. Configure UARTBAUD. The allowed values are from 1 to 255.
2. Select 8-bit or 9-bit mode by NBITEN (UARTCON.5).
3. Select invert input or not by UTRXINV (UARTCON.2).
4. Set P2.5/P4.0 as input by USRC (UARTSTA.0)
5. Set RXIE to ‘1’ if needed.
6. Set UTEN to ‘1’ to enable UART.

7. Wait for RXDONE change to ‘1’, or wait for interrupt.
8. Read received data from UARTBUF, clear RXDONE.
9. Go to step 7 to start another RX if needed, or turn off UART by clearing RXIE and UTEN.

12 16-bit x 16-bit Multiplier (MUL)

MUL is a 16-bit x 16-bit Sign Multiplier. It outputs a 32-bit result after "Arithmetic Right Shift" if needed. It is used in IDCT decoding. Figure 12-1 shows the MUL block diagram.

Figure 12-1: MUL block diagram



12.1 Registers

Register 12-1: MULCON – Multiplier Control Register

	Address	7	6	5	4	3	2	1	0	Default Value
MULCON Multiplier Control Register	0x98	-	-	MULR 2	MULR 1	MULR 0	MULSH			- - 00 0000
		-	-	R/W	R/W	R/W	R/W	R/W	R/W	

Not used

MUL Round Bit 2 After Shift
0: round bit = 0
1: round bit = 1

MUL Round Bit 1 After Shift
0: round bit = 0
1: round bit = 1

MUL Round Bit 0 After Shift
0: round bit = 0
1: round bit = 1

Arithmetic right shift result after multiple operation
000: shift 0 bit
001: shift 1 bit
010: shift 2 bits
011: shift 3 bits
100: shift 4 bits
101: shift 5 bits
110: shift 6 bits
111: shift 7 bits

Register 12-2: MULXL – Multiplier Operand X Low Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
MULXL Multiplier Operand X Low Byte Register	0xCB	W	W	W	W	W	W	W	W	xxxx xxxx

Multiplier Operand X [7:0]

Register 12-3: MULXH – Multiplier Operand X High Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
MULXH Multiplier Operand X's High Byte Register	0xCC	W	W	W	W	W	W	W	W	xxxx xxxx

Multiplier Operand X [15:8]

Register 12-4: MULYL – Multiplier Operand Y Low Byte Register

MULYL	Address	7	6	5	4	3	2	1	0	Default Value
Multiplier Operand Y Low Byte Register	0xCD	W	W	W	W	W	W	W	W	xxxx xxxx
		Multiplier Operand Y [7:0]								

Note: Write MULYL to start multiple operation

Register 12-5: MULYH – Multiplier Operand Y High Byte Register

MULYH	Address	7	6	5	4	3	2	1	0	Default Value
Multiplier Operand Y High Byte Register	0xCE	W	W	W	W	W	W	W	W	xxxx xxxx
		Multiplier Operand Y [15:8]								

Register 12-6: MULRES0 – Multiplier Result 0 Register

MULRES0	Address	7	6	5	4	3	2	1	0	Default Value
Multiplier Result 0 Register	0x91	R	R	R	R	R	R	R	R	xxxx xxxx
		Multiplier Result 0 [7:0]								

Register 12-7: MULRES1 – Multiplier Result 1 Register

MULRES1	Address	7	6	5	4	3	2	1	0	Default Value
Multiplier Result 1 Register	0x92	R	R	R	R	R	R	R	R	xxxx xxxx
		Multiplier Result 1 [7:0]								

Register 12-8: MULRES2 – Multiplier Result 2 Register

MULRES2	Address	7	6	5	4	3	2	1	0	Default Value
Multiplier Result 2 Register	0x93	R	R	R	R	R	R	R	R	xxxx xxxx
		Multiplier Result 2 [7:0]								

Register 12-9: MULRES3 – Multiplier Result 3 Register

MULRES3	Address	7	6	5	4	3	2	1	0	Default Value
Multiplier Result 3 Register	0x94	R	R	R	R	R	R	R	R	xxxx xxxx
		Multiplier Result 3 [7:0]								

12.2 Operation Guide

1. Write MULCON [2:0] to set Arithmetic Right Shift Count.
2. Write MULXH;
3. Write MULXL;
4. Write MULYH;
5. Write MULYL;
6. Wait 3 Cycle
7. Read result from MULRES3 to MULRES0
8. Read Round bit from MULCON [5:3] if needed.

13 Bit-Fetcher

Bit-Fetcher is used in Huffman Decode.

13.1 Registers

Register 13-1: BFCNT – Bit-fetcher Counter Register

	Address	7	6	5	4	3	2	1	0	Default Value
BFCNT Bit-fetcher Counter Register	0xBC	-	-	-	-	BFCNT				- - - - 0000
		-	-	-	-	W	W	W	W	

Bits 7–4: **Not used**

Bits 3–0: **Left shift bit count. It can shift 1-16 bits**

{BFDATA2, BFDATA1, BFDATA0}<<BFCNT

0000: shift 1 bit	1000: shift 9 bits
0001: shift 2 bit	1001: shift 10 bits
0010: shift 3 bits	1010: shift 11 bits
0011: shift 4 bits	1011: shift 12 bits
0100: shift 5 bits	1100: shift 13 bits
0101: shift 6 bits	1101: shift 14 bits
0110: shift 7 bits	1110: shift 15 bits
0111: shift 8 bits	1111: shift 16 bits

Register 13-2: BFBUF0 – Bit-fetcher Buffer Low Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
BFBUF0 Bit-fetcher Buffer Low Byte Register	0xBD	W	W	W	W	W	W	W	W	xxxx xxxx

Bits 7–0: **BFDATA0 [7:0]**

Register 13-3: BFBUF1 – Bit-fetcher Buffer Medium Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
BFBUF1 Bit-fetcher Buffer Medium Byte Register	0xBE	R	R	R	R	R	R	R	R	xxxx xxxx

Bits 7–0: **BFDATA1 [7:0]**

Register 13-4: BFBUF2 – Bit-fetcher Buffer High Byte Register

BFBUF2	Address	7	6	5	4	3	2	1	0	Default Value
Bit-fetcher Buffer High Byte Register	0xBF	R	R	R	R	R	R	R	R	xxxx xxxx
		BFDATA2 [7:0]								

13.2 Operation Guide

1. Write data in BFBUF0.
2. Write BFCNT for shifting.
3. Read Result from BFBUF1 and BFBUF2. {BFBUF2, BFBUF1} combine to form 16 bits result.

14 ADC

AX206 provides an eight-channel moderate conversion speed and a moderate resolution 10-bit successive approximated register Analog to Digital Converter (SARADC) for users to develop applications in the following areas:

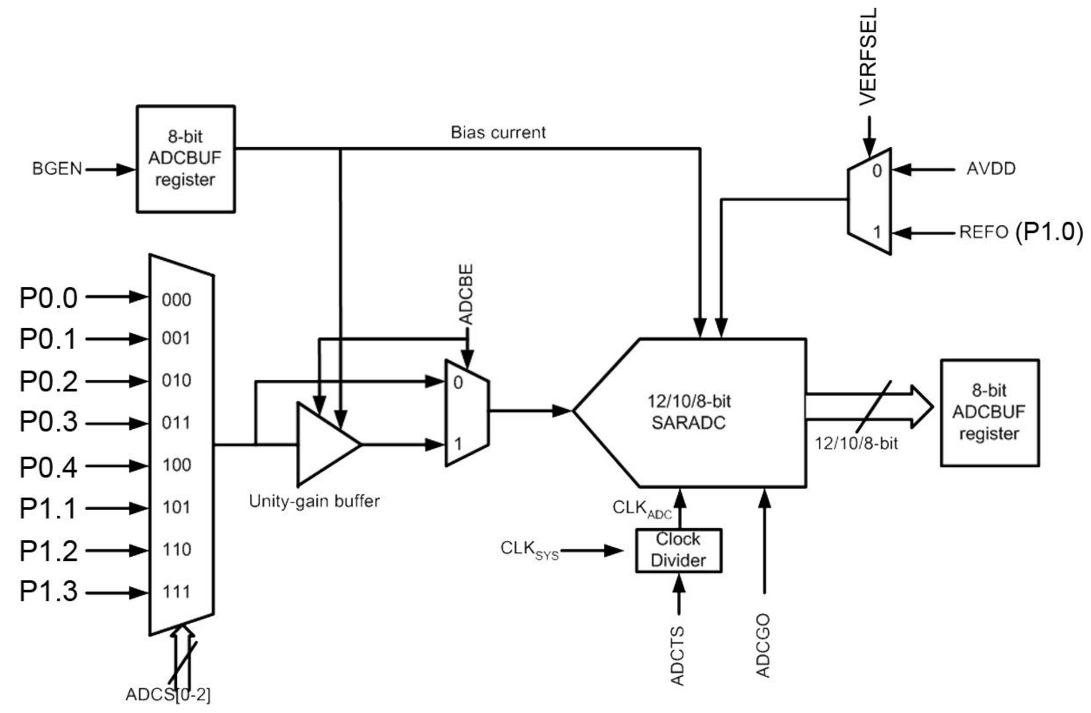
- Voice grade applications
- Audio applications requiring moderate performance
- Measurement requiring moderate performance and speed

To help users to fully understand AX206 ADC architecture, this chapter is divided into five sections each section is self-explained and self-contained.

- ADC Architecture and Registers
- ADC Conversion and Interrupt
- ADC Resolution and Speed
- ADC Clock Division
- ADC Digital Result Mapping

14.1 ADC Architecture

Figure 14-1: AX206 ADC architecture



AX206 has employed a successive approximation register type Analog to Digital Converter for the well-known moderate speed and simplicity. It also featured low power consumption compared to those high-end delta sigma type Analog to Digital Converter or pipelined Analog to Digital Converter.

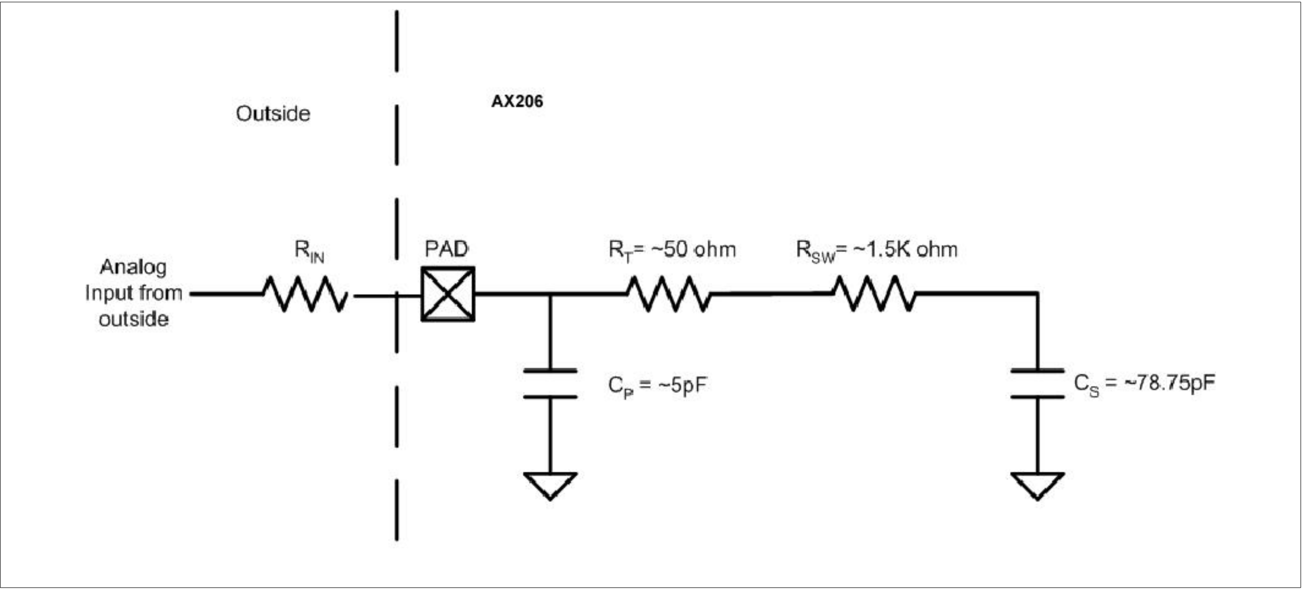
Figure 14-1 illustrates the internal ADC logic, with an 8-to1 analog MUX channel for selecting which channel is to input to the ADC circuitry. Channels 0 to 7 are multiplexed with GPIO pin: **P0.0, P0.1, P0.2, P0.3, P0.4, P1.1, P1.2 and P1.3.**

To use either one of the channels from 0 to 7, the GPIO has to be configured as input and the conversion results are stored in ADCBUFH/L registers. ADCCON register controls the operation of the ADC and the ADCBAUD register controls the ADC clock frequency which finally controls the ADC conversion speed.

To provide more flexibility, the ADC reference voltage can be selected to use AVDD or REFO during conversion. The reference voltage defines the highest possible conversion voltage the ADC can obtain during conversion. REFO has been multiplexed to GPIO P1.0; and AVDD is the internal AVDD power supply, which may have a slight difference to the external system power during use. There is an internal unity-gain buffer to buffer external analog signal to the input of ADC to meet the minimum sample-and-hold requirement. If user does not know the driving strength of the incoming analog signal, it is recommended to turn on the internal buffer to provide enough driving for the signal to the ADC circuitry, Figure 14-2 illustrates an equivalent circuit model for the sample-and-hold circuit to demonstrate the maximum allowed input impedance for the ADC channel input without enabling the internal input buffer.

To operate the ADC effectively and correctly, the AX206 provides register ADCCON to control the operation of the ADC conversion. The ADC clock is separately controlled by ADCBAUD register to select the ADC clock frequency. The ADC results are stored in an 10-bit data register (ADCBUFH, ADCBUFH) and user is required to get the value from the register twice in order to get a complete 10-bit digital output.

Figure 14-2: AX206 ADC equivalent circuit model (Input Buffer is disabled)



14.2 Registers

Register 14-1: ADCCON – ADC Control Register

	Address	7	6	5	4	3	2	1	0	Default Value
ADCCON ADC Control Register	0xD2	ADC GO	EOC	ADC CHOP	VERF SEL	ADC EN	ADCSEL			0000 0000
		R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	

Read: Check if conversion is finished
0: Finished
1: Not finished
Write: Start conversion
0: Not used
1: Start conversion

Check if end of conversion
0: End of conversion
1: Conversion has not ended

ADC offset chopper signal
0: Disable
1: Enable

ADC input signal select
000: P0.0
001: P0.1
010: P0.2
011: P0.3
100: P0.4
101: P1.1
110: P1.2
111: P1.3

ADC enable
0: disable 1: enable

VERF select
0: AVDD as ADC reference
1: P1.0 as ADC reference

Register 14-2: ADCBAUD – ADC Baudrate Register

	Address	7	6	5	4	3	2	1	0	Default Value
ADCBAUD ADC Baudrate Register	0xD3	TMR EN	ADC TS	ADCBAUD						00xx xxxx
		W	W	W	W	W	W	W	W	

Timer input enable
0: Disable
1: Enable

ADC conversion baud rate
0-63 cycles

Timer source select
0: Timer 0
1: Timer 1

Register 14-3: ADCBUFL – ADC Buffer Low Byte Register

	Address	7	6	5	4	3	2	1	0	Default Value
ADCBUFL ADC Buffer Low Byte Register	0xDC	R	R	R	R	R	R	R	R	xxxx xxxx

ADC Buffer [7:0]

Register 14-4: ADCBUFH – ADC Buffer High Byte Register

ADCBUFH	Address	7	6	5	4	3	2	1	0	Default Value
ADC Buffer High Byte Register	0xD4	R	R	R	R	R	R	R	R	xxxx xxxx
		ADC Buffer [15:8]								

14.3 Operation Guide

For ADC conversion, a general procedure is described below:

1. Configure ADCCON register by
   I. Select the channel to use (ADCCON[2:0])
   II. Select the reference voltage (ADCCON.4)
   III. Enable ADCEN (ADCCON.3=1)
   IV. Enable ADCGO (ADCCON.7=1)
2. Configure ADCBAUD register by
   I. Select the ADC trigger mode either manual or timer-trigger (ADCBAUD.7)
   II. Select the Timer to use (ADCBAUD.6)
   III. Set ADC conversion Clock Frequency (ADCBAUD[5:0])
3. Kick-start the ADC conversion by selecting which timer0/1 under timer-trigger mode or selecting ADCGO under manual mode
4. During conversion, ADCGO bit will keep as 1 and after each conversion, ADCGO bit will change to 0. The ADC done pending bit will change to 1 after each ADC conversion finish and interrupt will be flagged if ADC interrupt enable bit is set.

15 Electrical Characteristics

15.1 Absolute Maximum Rating

Table 15-1: Absolute maximum ratings

Ambient temperature under bias	0°C to +125°C
Storage temperature	-65°C - 150°C
Voltage on VDD with respect to VSS	0V to +3.6V

15.2 DC Characteristics

Table 15-2: POR DC Voltage Parameters

Symbol	Characteristics	Min	Typ	Max	Units	Conditions
$\mathbf{V_{POR}}$	Power-on Reset Voltage trip point	-	2.3	-	V	V_{DD} = 3.3V

Table 15-3: DC Voltage Parameters

Symbol	Characteristics	Min	Typ	Max	Units	Conditions
$\mathbf{V_{DD}}$	Supply Voltage	3.0	3.3	3.6	V	Without LDO Input from pin VDD
		3.0	3.3	3.6	V	Using LDO Output from LDO
$\mathbf{V_{IL}}$	Input Low Voltage I/O ports	VSS	-	0.8	V	VDD = 3.3V
$\mathbf{V_{IH}}$	Input High Voltage I/O ports	2.0	-	VDD	V	VDD = 3.3V
$\mathbf{V_{OL}}$	Output Low Voltage I/O ports	-	-	0.65	V	VDD = 3.3V, I_{OL}= 8mA
$\mathbf{V_{OH}}$	Output High Voltage I/O ports	2.80	-	-	V	VDD = 3.3V, I_{OH}= 8mA
$\mathbf{V_{SYS}}$	Hysteresis of I/O Schmitt Trigger	-	200	-	mV	VDD = 3.3V

Table 15-4: DC Current Parameters

Symbol	Characteristics	Min	Typ	Max	Units	Conditions
I_{IDLE} (without LDO)	Idle mode current	-	3.01	-	mA	24M OSC as system clock, stop USB clock VDD = 3.0V
	Idle mode current (32,768 Hz oscillator in normal mode)	-	11.18	-	uA	Fsystem_clock* = 32,768 Hz 24MHz oscillator off, Disable USB VDD = 3.0V
	Idle mode current (32,768 Hz oscillator in low power mode)	-	7.00	-	uA	
I_{HALT} (without LDO)	Halt mode current	-	617	-	uA	all ports pull-up Fsystem_clock* = 24MHZ 32,768 Hz oscillator off USB Disable VDD = 3.0V
	Halt mode current (32,768 Hz oscillator in normal mode)	-	6.20	-	uA	all ports pull-up Fsystem_clock* = 32,768 Hz 24MHz oscillator off USB Disable VDD = 3.0V
	Halt mode current (32,768 Hz oscillator in low power mode)	-	2.50	-	uA	
I_{PD}	Power Down mode current (without LDO)	-	0.83	-	uA	all ports pull-up High-speed oscillator off 32,768 Hz oscillator off VDD = 3.0V
I_{LDO}	LDO quiescent current	-	45.89	-	uA	VDD output around 3.3V
I_{LDO_LOAD}	LDO maximum loading current	-	-	50	mA	The sum of internal current consumption at VDD and the current output through VDD
I_{DD} (without LDO)	Normal mode current	-	18.06	-	mA	All ports pull-up Fsystem_clock* = 24MHZ 32,768 Hz oscillator off VDD = 3.0V
		-	23.91	-	mA	All ports pull-up Fsystem_clock* = 24MHZ 32,768 Hz oscillator off VDD = 3.0V

*Note: * - clock frequency after clock divider, please refer to Section 6.1.*

15.3 AC Parameters

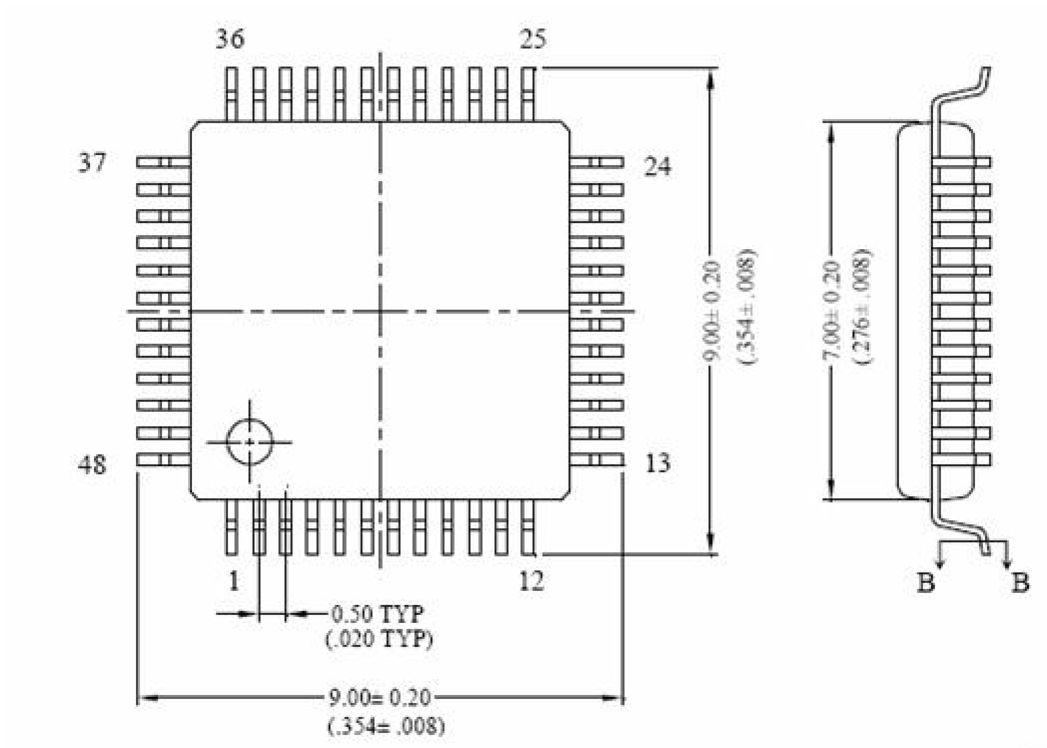
Table 15-5: AC Parameters

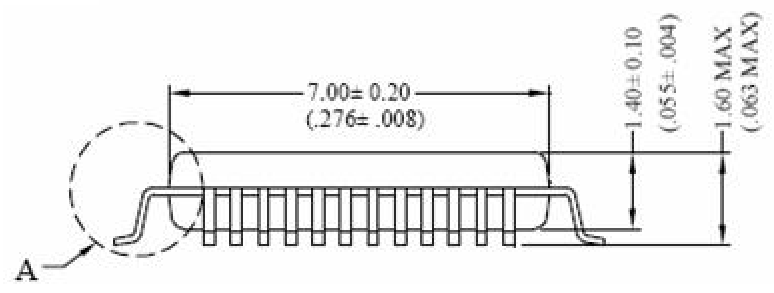
Symbol	Characteristics	Min	Typ	Max	Units	Conditions
F_{OSC}	External OSC1 Frequency	DC	-	24	MHz	
	Oscillator Frequency	1	-	24	MHz	
F_{CORE}	Digital Core Operating Frequency	-	-	24	MHZ	VDD = 3.3V
T_{CY}	Instruction Cycle Time	41.7	-	DC	ns	VDD = 3.3V
T_{MCLR}	/MCLR Pulse Width	2	-	-	us	VDD = 3.3V
T_{WDT}	Watchdog Timer Time-out Period	-	16	-	ms	VDD = 3.3V, postscaler divide ratios = 1:1
		-	70	-	s	VDD = 3.3V, postscaler divide ratios= 1:4096
T_{OR}	Output Rise Time I/O ports	-	8.0	-	ns	VDD = 3.3V RL = 510 ohm; CL = 47 pF
		-	6.4	-	ns	VDD = 5.0V RL = 510 ohm; CL = 47 pF
T_{OF}	Output Fall Time I/O ports	-	8.4	-	ns	VDD = 3.3V RL = 510 ohm; CL = 47 pF
		-	7.4	-	ns	VDD = 3.3V RL = 510 ohm; CL = 47 pF

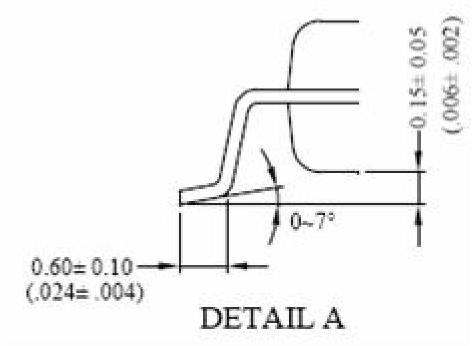
Table 15-6: ADC Parameters

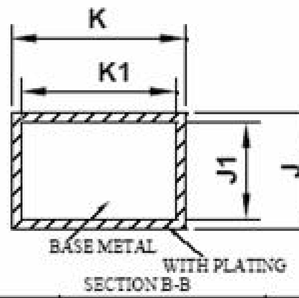
Parameter	Condition	Min	Typ	Max	Unit
DC Accuracy					
Resolution		-		8	bits
INL		-	+/-1	-	LSB
DNL		-	+/-1	-	LSB
Offset Error		-	+/-1	-	LSB
Dynamic Performance					
Conversion speed	Cload = 100pF Rload = infinite INBUF = 0	-	-	5	us
Analog Output					
Analog output range	INBUF = 0	0	-	VDDA	V
Power Specifications					
Standby Current		-	-	10	nA
Analog supply (VDDA)		2.7	3.3	3.6	V
Digital supply (VDD)		2.7	3.3	3.6	V
Analog supply current		-	0.5	-	mA
Digital supply current (avg)	DIN change @4MHz	-	19.2	-	uA

16 Package dimensions









DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
J	0.107	0.197	.0042	.0076
J1	0.107	0.147	.0042	.0058
K	0.200	0.250	.0079	.0098
K1	0.200	0.300	.0079	.0118

引线间距 Lead Pitch	0.50mm(19.7mil)
载体尺寸 Pad Size	205mil×205mil
载体打凹深度 Depressed Die Pad	0.120±0.025 (0.007±0.001)
单位 Unit	mm(inch)

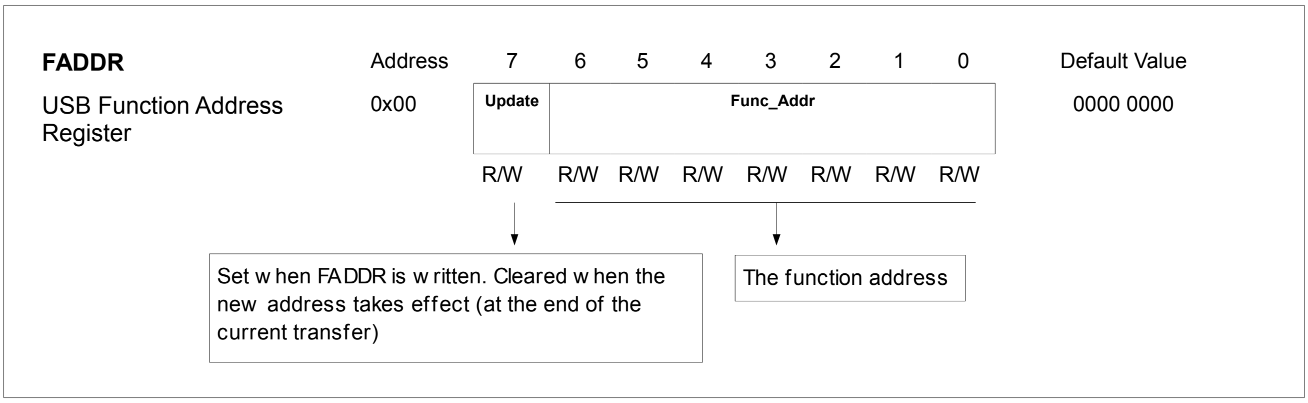
Appendix I Revision History

Date	Version	Revised items	Revised by
2008	0.0.3	1. First draft	
2009.2.27	1.0.0	1. Revise layout 2. Section 6.3: Revise Figure 6-5 and Figure 6-6	Erica Cheong erica.cheong@appotech.com
2009.6.29	1.1.0	1. Section 2.1.1: update Figure 2-1 pin assignment diagram 2. Section 6.1: update Figure 6-2 clock system block diagram 3. Section 6.1.5: update frequency of RC oscillator 4. Section 6.3: update recommended circuit for operating in 5V system 5. Section 8.1.4: add Timer0 operation guide 6. Section 8.2.4: add Timer1 operation guide 7. Section 8.3.4: add Timer2 operation guide 8. Section 8.4: update Watchdog description 9. Section 8.5.2: add description for low power mode 10. Section 8.5.3: add RTCC Timer operation guide 11. Section 9.5: add baudrate equation 12. Section 10: correct Table 10-1, EP0 has shared input and output buffer 13. Section 10.1: correct Figure 14-1 14. Update special function register names 15. Update factsheet	Erica Cheong erica.cheong@appotech.com

Appendix II USB Function Register

FADDR

FADDR is an 8-bit register that should be written with the function's 7-bit address (received through a SET_ADDRESS descriptor). The new address will not take effect until the status stage of the device request is completed. It is then used for decoding the function address in subsequent token packets.



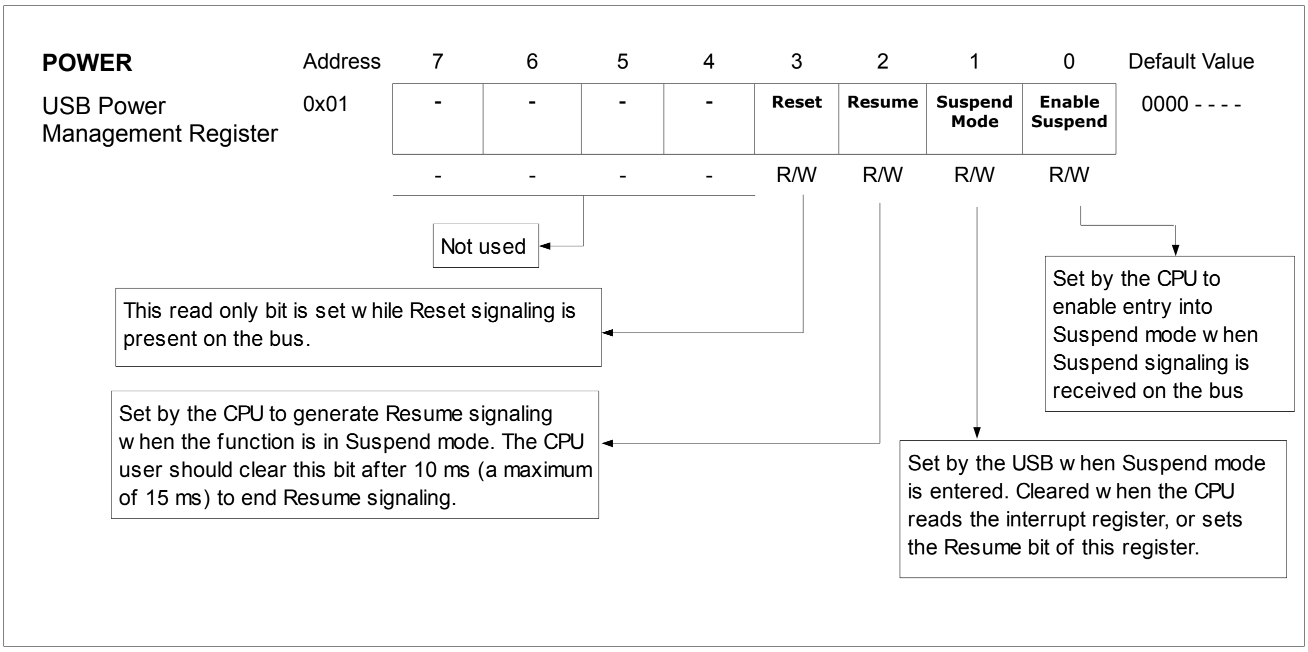
FADDR	Address	7	6	5	4	3	2	1	0	Default Value
USB Function Address Register	0x00	Update	Func_Addr							0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Update: Set w hen FADDR is w ritten. Cleared w hen the new address takes effect (at the end of the current transfer)

Func_Addr: The function address

POWER

POWER is an 4-bit register that is used for controlling Suspend and Resume signaling.



POWER	Address	7	6	5	4	3	2	1	0	Default Value
USB Power Management Register	0x01	-	-	-	-	Reset	Resume	Suspend Mode	Enable Suspend	0000 - - - -
		-	-	-	-	R/W	R/W	R/W	R/W	

Bits 7–4: Not used

Reset: This read only bit is set w hile Reset signaling is present on the bus.

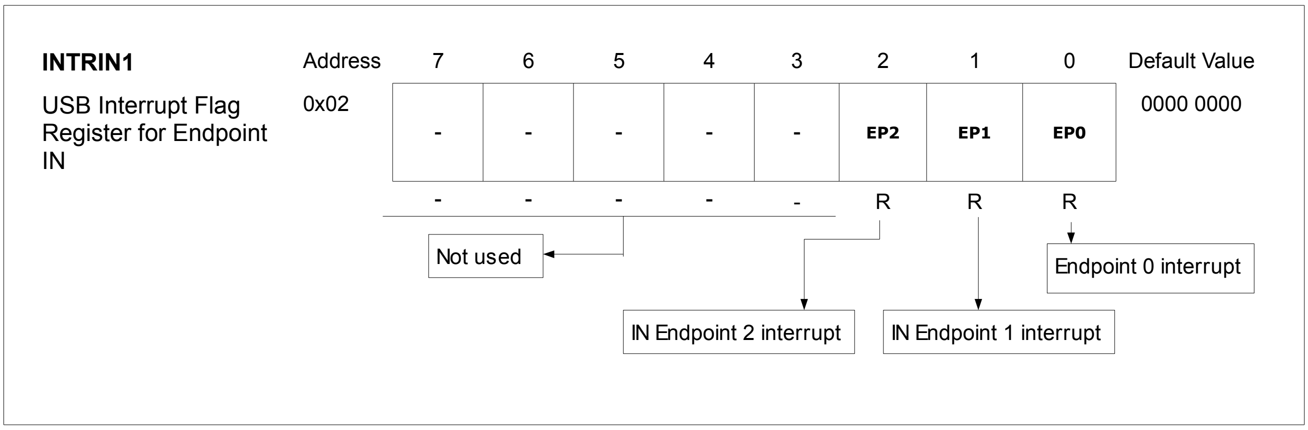
Resume: Set by the CPU to generate Resume signaling w hen the function is in Suspend mode. The CPU user should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling.

Suspend Mode: Set by the USB w hen Suspend mode is entered. Cleared w hen the CPU reads the interrupt register, or sets the Resume bit of this register.

Enable Suspend: Set by the CPU to enable entry into Suspend mode w hen Suspend signaling is received on the bus

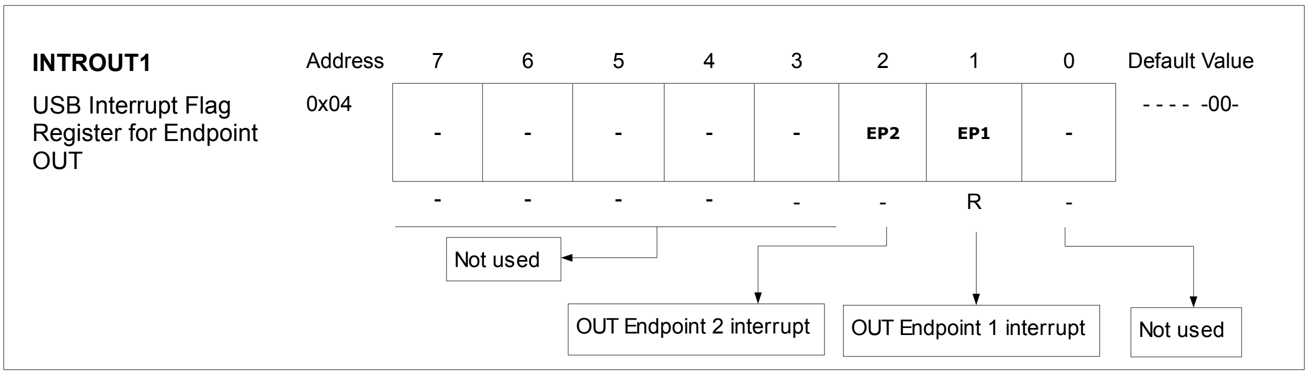
INTRIN1

IntrIn1 is a 3-bit read-only register that indicates which of the interrupts for IN Endpoints 1 and 2 is currently active. It also indicates whether the Endpoint 0 interrupt is currently active. All active interrupts will be cleared when this register is read.



INTROUT1

IntrOut1 is a 2-bit read-only register that indicates which of the interrupts for OUT Endpoints 1 are currently active. All active interrupts will be cleared when this register is read.



INTRUSB

INTRUSB is a 4-bit read-only register that indicates which USB interrupts are currently active. All active interrupts will be cleared when this register is read.

INTRUSB	Address	7	6	5	4	3	2	1	0	Default Value
USB Interrupt Flag Register	0x06	-	-	-	-	**SOF**	**Reset**	**Resume**	**Suspend**	- - - - 0000
		-	-	-	-	R	R	R	R	
		Not used				Set at the start of each frame	Set when Reset signaling is detected on the bus	Set when Resume signaling is detected on the bus while the device is in Suspend mode	Set when Suspend signaling is detected on the bus	

INTRIN1E

IntrIn1E is a 3-bit register that provide interrupt enable bits for the interrupts in IntrIn1 and IntrIn2 respectively. On reset, the bits corresponding to Endpoint 0 and the IN endpoints included in the design are set to 1, while the remaining bits are set to 0.

INTRIN1E	Address	7	6	5	4	3	2	1	0	Default Value
USB Interrupt Enable Register for Endpoint IN	0x07	-	-	-	-	-	**IntrInEn 2**	**IntrInEn 1**	**IntrInEn 0**	- - - - -111
		-	-	-	-	-	-	RW	RW	
		Not used					IN Endpoint 1 interrupt enable	IN Endpoint 1 interrupt enable	Endpoint 0 interrupt enable	

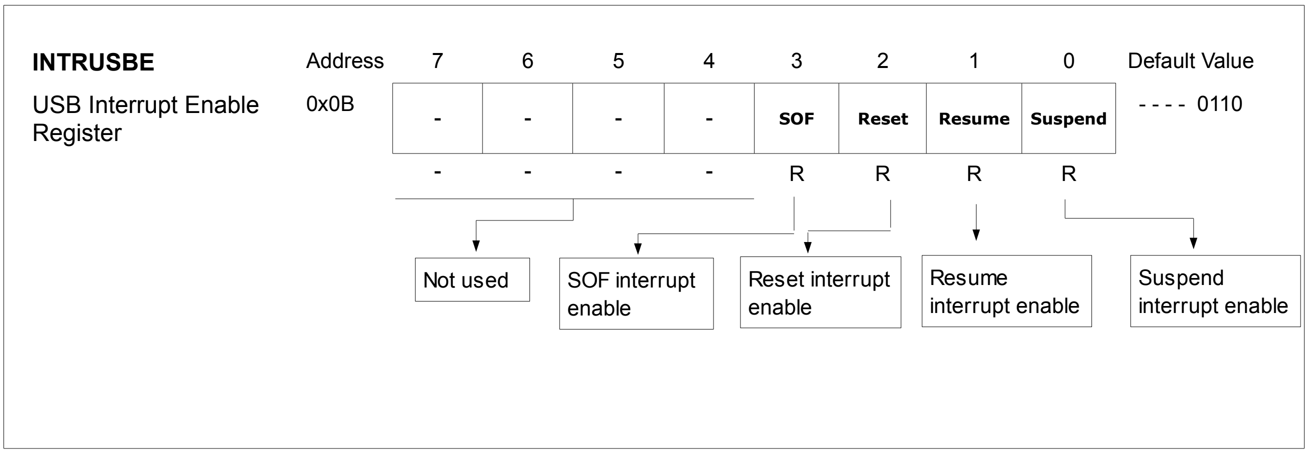
INTROUT1E

IntrOut1E is a 2-bit register that provide interrupt enable bits for the interrupts in IntrOut1 and IntrOut2. On reset, the bits corresponding to the OUT end points included in the design are set to 1, while the remaining bits are set to 0.

INTROUT1E	Address	7	6	5	4	3	2	1	0	Default Value
USB Interrupt Enable Register for Endpoint OUT	0x09	-	-	-	-	-	**IntrOut En2**	**IntrOutE n1**	-	0000 0110
		-	-	-	-	-	R/W	RW	-	
		Not used					OUT Endpoint 2 interrupt enable	OUT Endpoint 1 interrupt enable	Not used	

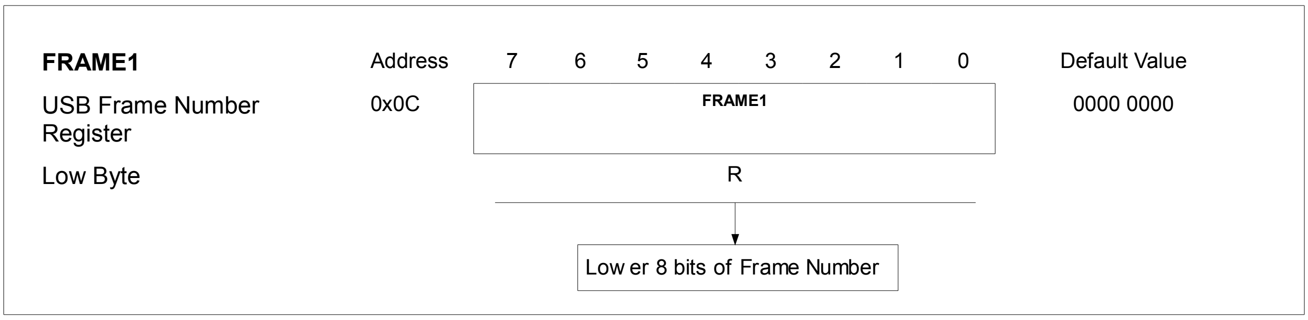
INTRUSBE

INTRUSBE is a 4-bit register provides interrupt enable bits for each of the interrupts in INTRUSB.



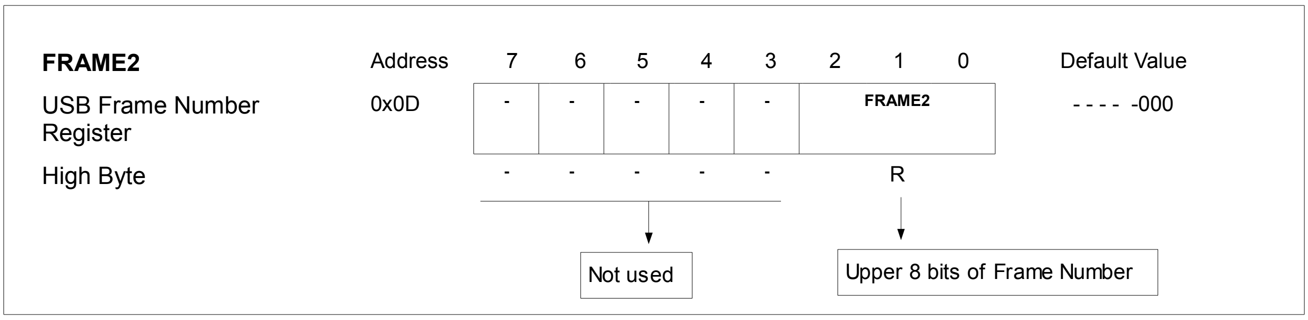
FRAME1

FRAME1 is an 8-bit read-only register that holds the lower 8 bits of the last received frame number.



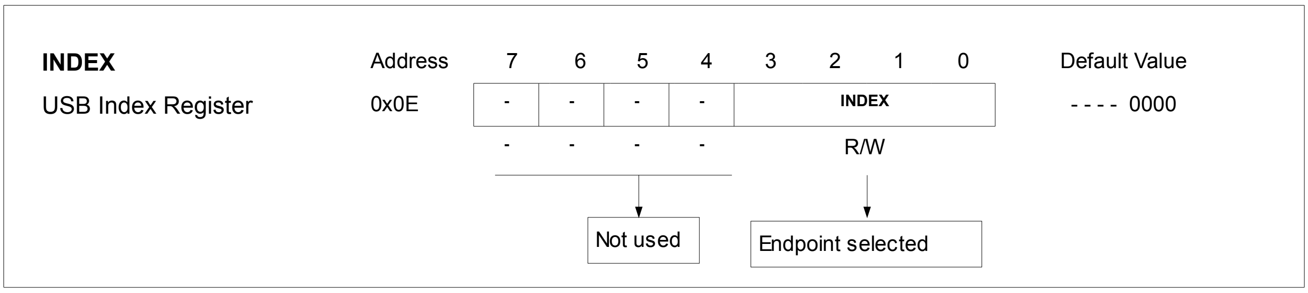
FRAME2

Frame2 is a 3-bit read-only register that holds the upper 3 bits of the last received frame number.



INDEX

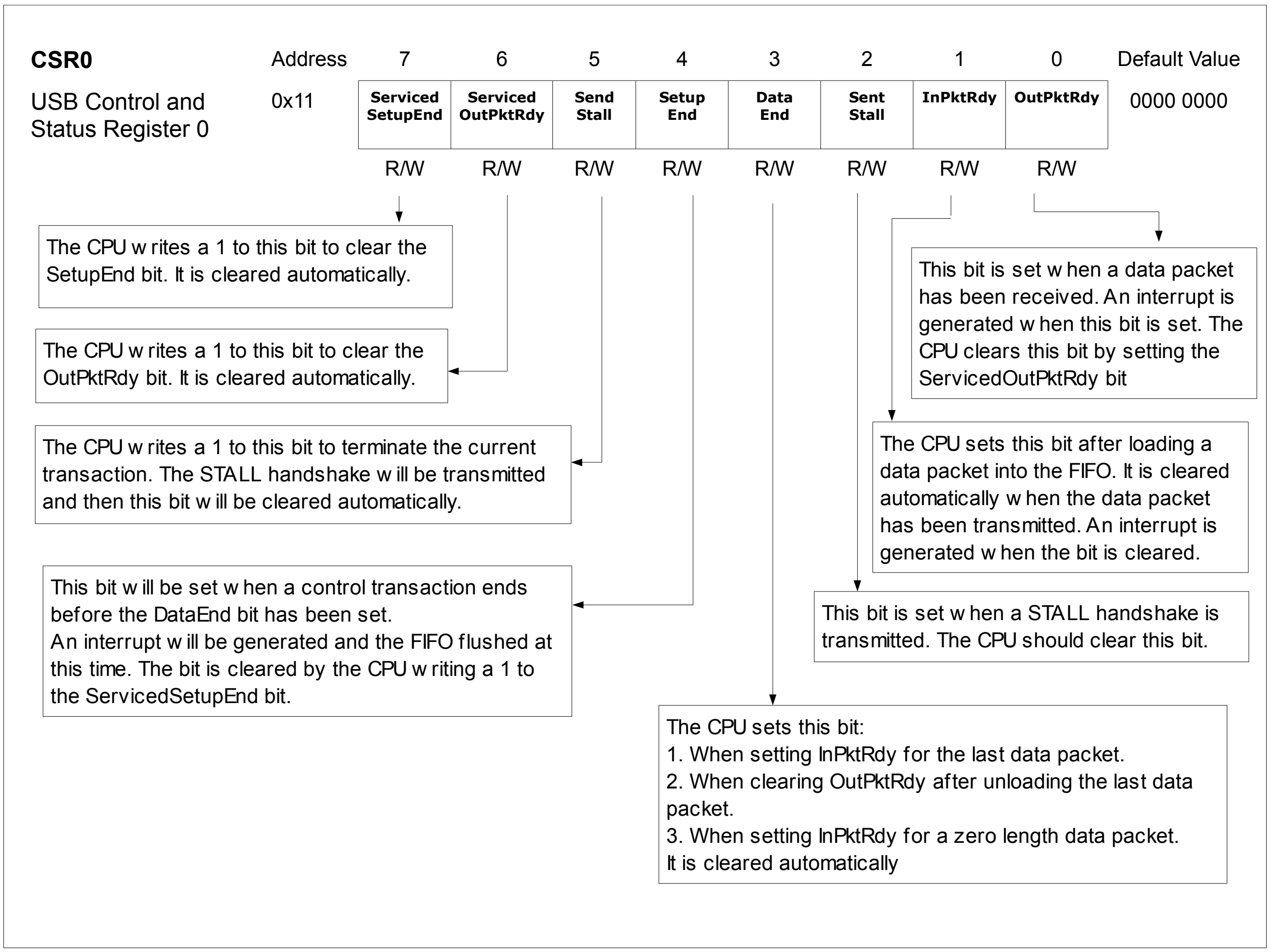
INDEX is a 4-bit register that determines which endpoint control/status registers are accessed at addresses 10h to 17h. Each IN endpoint and each OUT endpoint have their own set of control/status registers. Only one set of IN control/status and one set of OUT control/status registers appear in the memory map at any one time. Before accessing an endpoint's control/status registers, the endpoint number should be written to the index register to ensure that the correct control/status registers appear in the memory map.



	Address	7	6	5	4	3	2	1	0	Default Value
INDEX USB Index Register	0x0E	-	-	-	-	INDEX				- - - - 0000
		-	-	-	-	R/W				

CSR0

CSR0 is an 8-bit register that provides control and status bits for Endpoint 0.



	Address	7	6	5	4	3	2	1	0	Default Value
CSR0 USB Control and Status Register 0	0x11	Serviced SetupEnd	Serviced OutPktRdy	Send Stall	Setup End	Data End	Sent Stall	InPktRdy	OutPktRdy	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The CPU w rites a 1 to this bit to clear the SetupEnd bit. It is cleared automatically.

The CPU w rites a 1 to this bit to clear the OutPktRdy bit. It is cleared automatically.

The CPU w rites a 1 to this bit to terminate the current transaction. The STALL handshake w ill be transmitted and then this bit w ill be cleared automatically.

This bit w ill be set w hen a control transaction ends before the DataEnd bit has been set.
An interrupt w ill be generated and the FIFO flushed at this time. The bit is cleared by the CPU w riting a 1 to the ServicedSetupEnd bit.

The CPU sets this bit:
1. When setting InPktRdy for the last data packet.
2. When clearing OutPktRdy after unloading the last data packet.
3. When setting InPktRdy for a zero length data packet.
It is cleared automatically

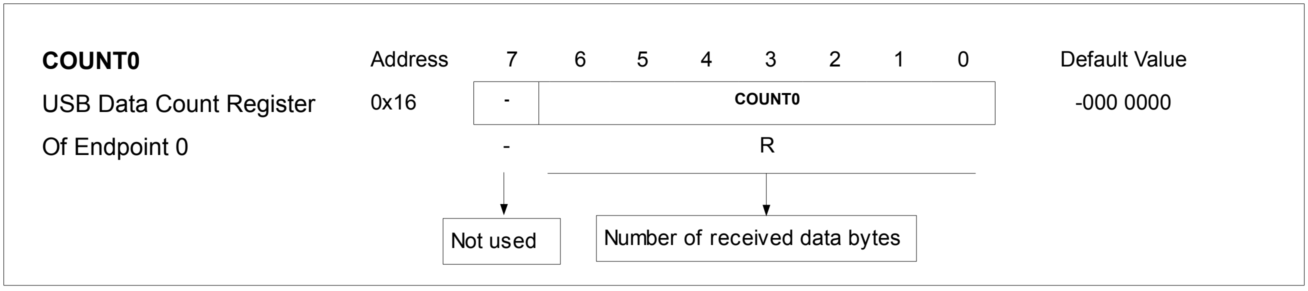
This bit is set w hen a STALL handshake is transmitted. The CPU should clear this bit.

The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically w hen the data packet has been transmitted. An interrupt is generated w hen the bit is cleared.

This bit is set w hen a data packet has been received. An interrupt is generated w hen this bit is set. The CPU clears this bit by setting the ServicedOutPktRdy bit

COUNT0

COUNT0 is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned is valid while OutPktRdy (CSR0.D0) is set.



INMAXP

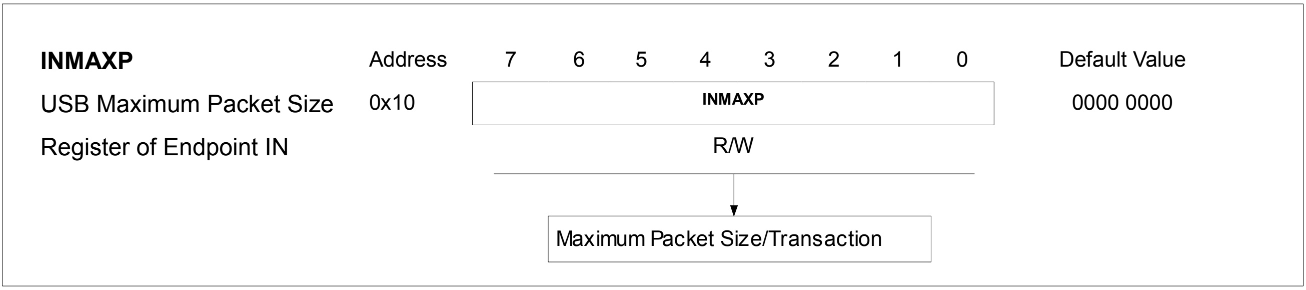
INMAXP is an 8-bit register that holds the maximum packet size for transactions through the currently selected IN endpoint – in units of 8 bytes. In setting this value, user should note the constraints placed by the USB Specification on packet sizes for Bulk and Interrupt transactions in Full-speed operations.

There is an INMAXP register for each IN endpoint (except Endpoint 0).

The value written to this register should match the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint (see Universal Serial Bus Specification Revision 1.1, Chapter 9). A mismatch could cause unexpected results.

If a value greater than the configured IN FIFO size for the endpoint is written to this register, the value will be automatically changed to the IN FIFO size. If the value written to this register is less than, or equal to, half the IN FIFO size, two IN packets can be buffered.

The register is reset to 0. If this register is changed after packets have been sent from the endpoint, then the endpoint IN FIFO should be completely flushed (using the FlushFIFO bit (D3) in InCSR1) after writing the new value to the INMAXP register.



INCSR1

InCSR1 is an 7-bit register that provides control and status bits for transfers through the currently selected IN endpoint. There is an InCSR1 register for each IN endpoint (not including Endpoint 0).

INCSR1	Address	7	6	5	4	3	2	1	0	Default Value
USB Control and Status Register 1 for Endpoint IN	0x11	-	**Clr DataTog**	**Sent Stall**	**Send Stall**	**Flush FIFO**	**Under Run**	**FIFO NotEmpty**	**InPktRdy**	-000 0000
		-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Not used

The CPU w rites a 1 to this bit to reset the endpoint IN data toggle to 0.

This bit is set w hen a STALL handshake is transmitted. The FIFO is flushed and the InPktRdy bit is cleared (see below). The CPU should clear this bit.

The CPU w rites a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. This bit has no effect if the IN endpoint is in ISO mode.

The CPU w rites a 1 to this bit to flush the next packet to be transmitted from the endpoint IN FIFO. The FIFO pointer is reset and the InPktRdy bit (below) is cleared. Note: If the FIFO contains tw o packets, FlushFIFO w ill need to be set tw ice to completely clear the FIFO.

In ISO mode, this bit is set w hen a zero length data packet is sent after receiving IN token w ith the InPktRdy bit not set. In Bulk/Interrupt mode, this bit is set w hen a NAK is returned in response to an IN token. The CPU should clear this bit. It is cleared automatically

This bit is set w hen there is at least 1 packet in the IN FIFO.

The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically w hen a data packet has been transmitted. An interrupt is generated (if enabled) w hen the bit is cleared

INCSR2

INSR2 is an 8-bit register that provides further control bits for transfers through the currently selected IN endpoint. There is an InCSR2 register for each IN endpoint (not including Endpoint 0).

INCSR2	Address	7	6	5	4	3	2	1	0	Default Value
USB Control and Status Register 2 for Endpoint IN	0x12	**AutoSet**	**ISO**	**Mode**	**DMAEnab**	**FrcDataTog**	**-**	**-**	**-**	0000 0- - -
		R/W	R/W	R/W	R/W	R/W	-	-	-	

Not used

If the CPU sets this bit then InPktRdy will be automatically set when data of the maximum packet size (value in InMaxP) is loaded into the IN FIFO. If a packet of less than the maximum packet size is loaded, then InPktRdy will have to be set manually. When 2 packets are in the IN FIFO then InPktRdy will also be automatically set when the first packet has been sent, if the second packet is the maximum packet size.

The CPU sets this bit to force the endpoint IN data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by interrupt IN endpoints that are used to communicate rate feedback for isochronous endpoints.

The CPU sets this bit to enable the IN endpoint for isochronous transfers, and clears it to enable the IN endpoint for bulk or interrupt transfers.

The CPU sets this bit to enable the endpoint direction as IN, and clears it to enable the endpoint direction as OUT. Valid only where the same endpoint FIFO is used for both IN and OUT transactions.

The CPU sets this bit to enable the DMA request for the IN endpoint.

OUTMAXP

OUTMAXP is an 8-bit register that holds the maximum packet size for transactions through the currently selected OUT endpoint –in units of 8 bytes, except that a value of 128 sets the maximum packet size to 1023 (the maximum size for an isochronous packet) rather than 1024. In setting this value, you should note the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transactions in Full-speed operations. There is an OutMaxP register for each OUT endpoint (except Endpoint 0).

The value written to this register should match the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint. A mismatch could cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the OUT endpoint, and should not exceed half the FIFO size if double buffering is required. If a value greater than the configured OUT FIFO size for the endpoint is written to this register, the value will be automatically changed to the OUT FIFO size. If the value written to this register is less than, or equal to, half the OUT FIFO size, two OUT packets can be buffered.

OUTMAXP	Address	7	6	5	4	3	2	1	0	Default Value
USB Maximum Packet Size	0x13	**OUTMAXP**								0000 0000
Register of Endpoint OUT		R/W								
		Maximum Packet Size/Transaction								

OUTCSR1

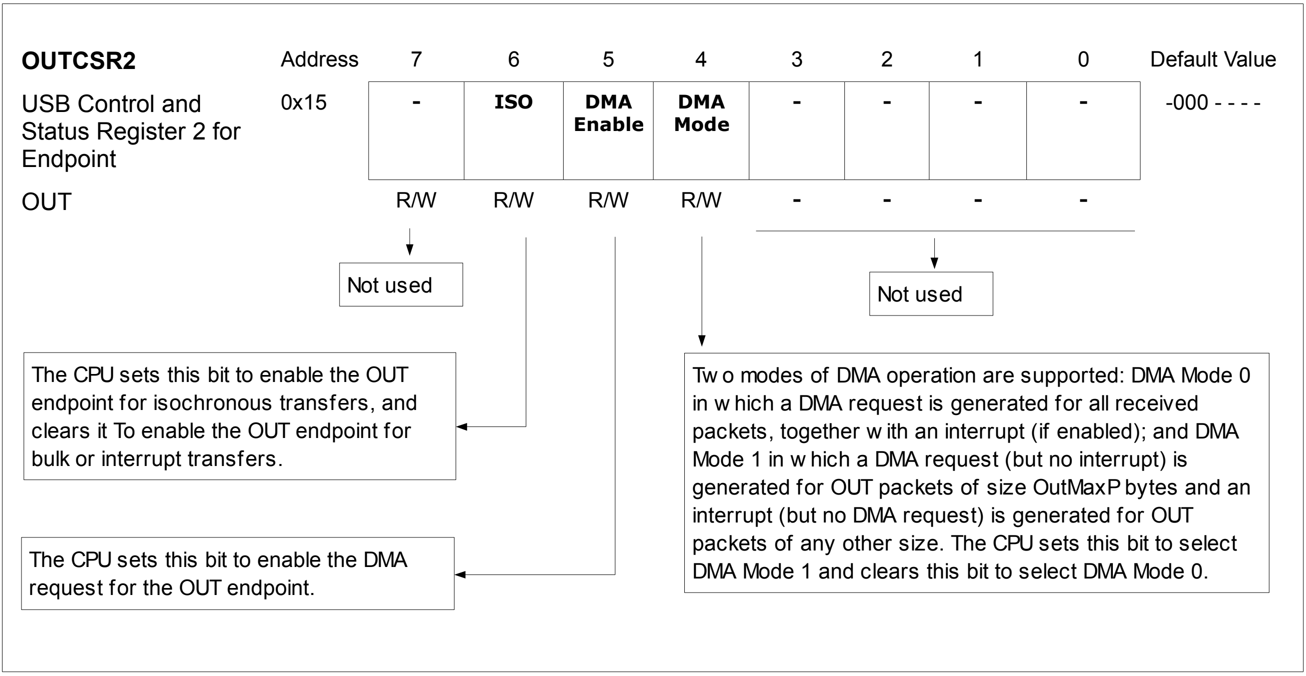
OutCSR1 is an 8-bit register that provides control and status bits for transfers through the currently selected OUT endpoint.

OUTCSR1	Address	7	6	5	4	3	2	1	0	Default Value
USB Control and Status Register 1 for Endpoint	0x14	**Clr DataTog**	**Sent Stall**	**Send Stall**	**Flush FIFO**	**Data Error**	**Over Run**	**FIFO Full**	**OutPktRdy**	0000 0000
OUT		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- Bit 7 (Clr DataTog): The CPU w rites a 1 to this bit to reset the endpoint data toggle to 0.
- Bit 6 (Sent Stall): This bit is set w hen a STALL handshake is transmitted. The CPU should clear this bit.
- Bit 5 (Send Stall): The CPU w rites a 1 to this bit to issue a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect if the OUT endpoint is in ISO mode.
- Bit 4 (Flush FIFO): The CPU w rites a 1 to this bit to flush the next packet to be read from the endpoint OUT FIFO. Note: If the FIFO contains tw o packets, FlushFIFO w ill need to be set tw ice to completely clear the FIFO.
- Bit 3 (Data Error): This bit is set w hen OutPktRdy is set if the data packet has a CRC or bit-stuff error. It is cleared w hen OutPktRdy is cleared. The bit is only valid in ISO mode.
- Bit 2 (Over Run): This bit is set if an OUT packet cannot be loaded into the OUT FIFO. The CPU should clear this bit. The bit is only valid in ISO mode.
- Bit 1 (FIFO Full): This bit is set w hen no more packets can be loaded into the OUT FIFO.
- Bit 0 (OutPktRdy): This bit is set w hen a data packet has been received. The CPU should clear this bit w hen The packet has been unloaded from the OUT FIFO. An interrupt is generated w hen the bit is set.

OUTCSR2

OutCSR2 is an 8-bit register that provides further control bits for transfers through the currently selected OUT endpoint.



OUTCSR2	Address	7	6	5	4	3	2	1	0	Default Value
USB Control and Status Register 2 for Endpoint	0x15	-	**ISO**	**DMA Enable**	**DMA Mode**	-	-	-	-	-000 - - - -
OUT		R/W	R/W	R/W	R/W	-	-	-	-	

Bit 7: Not used

Bits 3–0: Not used

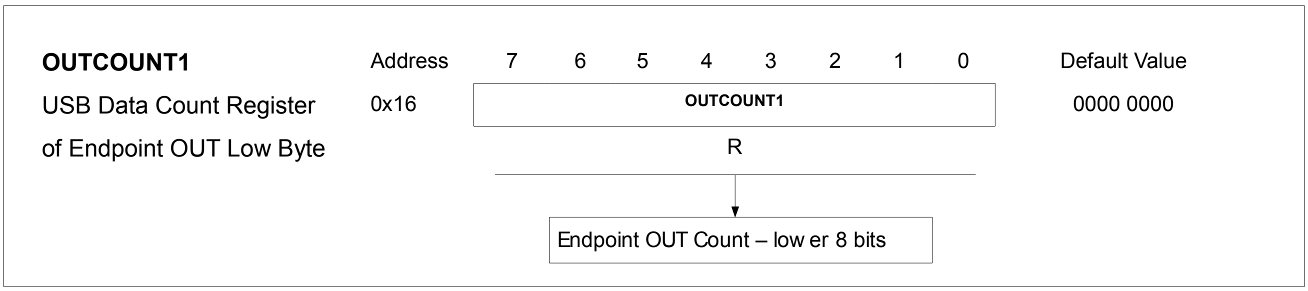
ISO: The CPU sets this bit to enable the OUT endpoint for isochronous transfers, and clears it To enable the OUT endpoint for bulk or interrupt transfers.

DMA Enable: The CPU sets this bit to enable the DMA request for the OUT endpoint.

DMA Mode: Tw o modes of DMA operation are supported: DMA Mode 0 in w hich a DMA request is generated for all received packets, together w ith an interrupt (if enabled); and DMA Mode 1 in w hich a DMA request (but no interrupt) is generated for OUT packets of size OutMaxP bytes and an interrupt (but no DMA request) is generated for OUT packets of any other size. The CPU sets this bit to select DMA Mode 1 and clears this bit to select DMA Mode 0.

OUTCOUNT1

OUTCOUNT1 is an 8-bit read-only register that holds the lower 8 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OutPktRdy (OutCSR1.D0) is set.



OUTCOUNT1	Address	7–0	Default Value
USB Data Count Register	0x16	OUTCOUNT1	0000 0000
of Endpoint OUT Low Byte		R	

Bits 7–0: Endpoint OUT Count – low er 8 bits

OUTCOUNT2

OUTCOUNT2 is a 3-bit read-only register that holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OutPktRdy (OutCSR1.D0) is set.

OUTCOUNT2	Address	7	6	5	4	3	2	1	0	Default Value
USB Data Count Register of	0x17	-	-	-	-	-		OUTCOUNT2		- - - - -000
Endpoint OUT High Byte		-	-	-	-	-		R		
				Not used				Endpoint OUT count – upper 3 bits		

FIFO 0, 1 and 2 (Address 0x20 – 0x22)

These 3 addresses provide CPU access to the FIFOs for each endpoint. Writing to these addresses loads data into the IN FIFO for the corresponding endpoint. Reading from these addresses unloads data from the OUT FIFO for the corresponding endpoint.

The FIFOs are located on byte boundaries (Endpoint 0 at 0x20, Endpoint 1 at 0x21, Endpoint 2 at 0x22).