Chip for people

CFP5102
Data Sheet

Document No.： CFP5102 DataSheet v0.1

Issue Date： 2007-12-3

CFP5102 High Performance 8-bit RISC Microcontroller

High Performance 8-bit RISC MCU

- DC-24MHz operation
- Compatible with 8051
- 160ns internal interrupt response at 24 MHz
- Single cycle per byte fetch
- All instructions are single-cycled except branching instructions

Program Memory and Data Memory

- 16K Bytes OTP program memory
  384 Bytes internal SRAM for data/stack memory
- External 64K Bytes external memory interface compatibles to true 8051
- Two data pointers for indirect addressing

Interrupt Features

- 14 vectored interrupts
- 2-level interrupt priority
- External wakeup/interrupt capabilities on 3 GPIO pins and USB PHY

Flexible I/O

- 32 GPIO pins organized in 4 ports.
- Support open-drain or push-pull driving
- CMOS/TTL-level Schmitt triggered inputs
- Optional internal 20Kohm pull-up resistor
- 6mA sink/source current driving capability

Digital Peripheral Features

- Two 16-bit timers compatible with 8051's timer0 timer1
  Enhanced 8052's timer 2. Support Capture/Reload, PWM mode and clock generation
- Real-time wake up for software clock implementation.
- Watchdog Timer with on-chip 16KHz RC oscillator
- High-speed full-duplex 8051-compatible serial port
  Enhanced SPI with double-buffer for read and write. Operates up to 12Mb/s
- GPSI with 16-bit/32-bit packet header, for interfacing RF module, Ethernet PHY and for inter-chip communication
- Full-speed USB 2.0 Device controller module with 3 endpoints (including endpoint 0) and 208 bytes FIFO
  Endpoint 0: 16 bytes
  Endpoint 1: IN 64 bytes / OUT 64 bytes
  Endpoint 2: IN 32 bytes / OUT 32 bytes

Analog Peripheral Features

- 4~24MHz Crystal Oscillator
- 32,768 Hz Real-time Crystal Oscillator
- 16KHz RC oscillator
- Full-speed USB 2.0 Device PHY
- Digital PLL for 48MHz clock generation
- 8 channels 8-bit ADC
- Power-on reset
- On-chip regulator for 5V to 3.3V conversion

Programming and Debugging Support

- In-System Programming (ISP) support
- In-System Debugging (ISD) support

Packages

- 48-pin LQFP
- DIE form

1 Product Overview

1.1 Description

CFP5102 is an 8051 compatible mixed-signal 8-bit microcontroller. It integrates advanced digital and analog peripherals to accommodate different applications. For digital peripherals, it supports digital interfaces including UART, SPI, GPSI, three timers, watchdog timer, and an USB 2.0 full-speed device controller. For analog peripherals, it integrates an 8-bit ADC, a low dropout voltage regulator.

The microcontroller has an advanced RISC-based architecture and compatible with standard 8051 core running at 24 MIPS. All instructions take single clock cycle, except for program branching and accessing external memory.

CFP5102 has 16K byte OTP program memory and 384 bytes data memory. It also supports external memory interface (EMI), which allows program code running from external memory.

CFP5102 I/O pins have flexible programmable capabilities. It supports external port wakeup, open-drain and push-pull modes with internal pull-up resistor option.

CFP5102 has built-in In-System Programming (ISP) and In-System Debugging (ISD) capabilities, which supports third party tools to provide an integrated development environment including editor, macro assembler, debugger, programming and software peripherals.

In order to support portable applications with low power consumption, CFP5102 has three power-saving modes: IDLE, HALT and Power Down. It also has a sophisticated clock control system that can enable clock sources only for the peripherals that are running to further reduce power consumption.

1.2 System Architecture

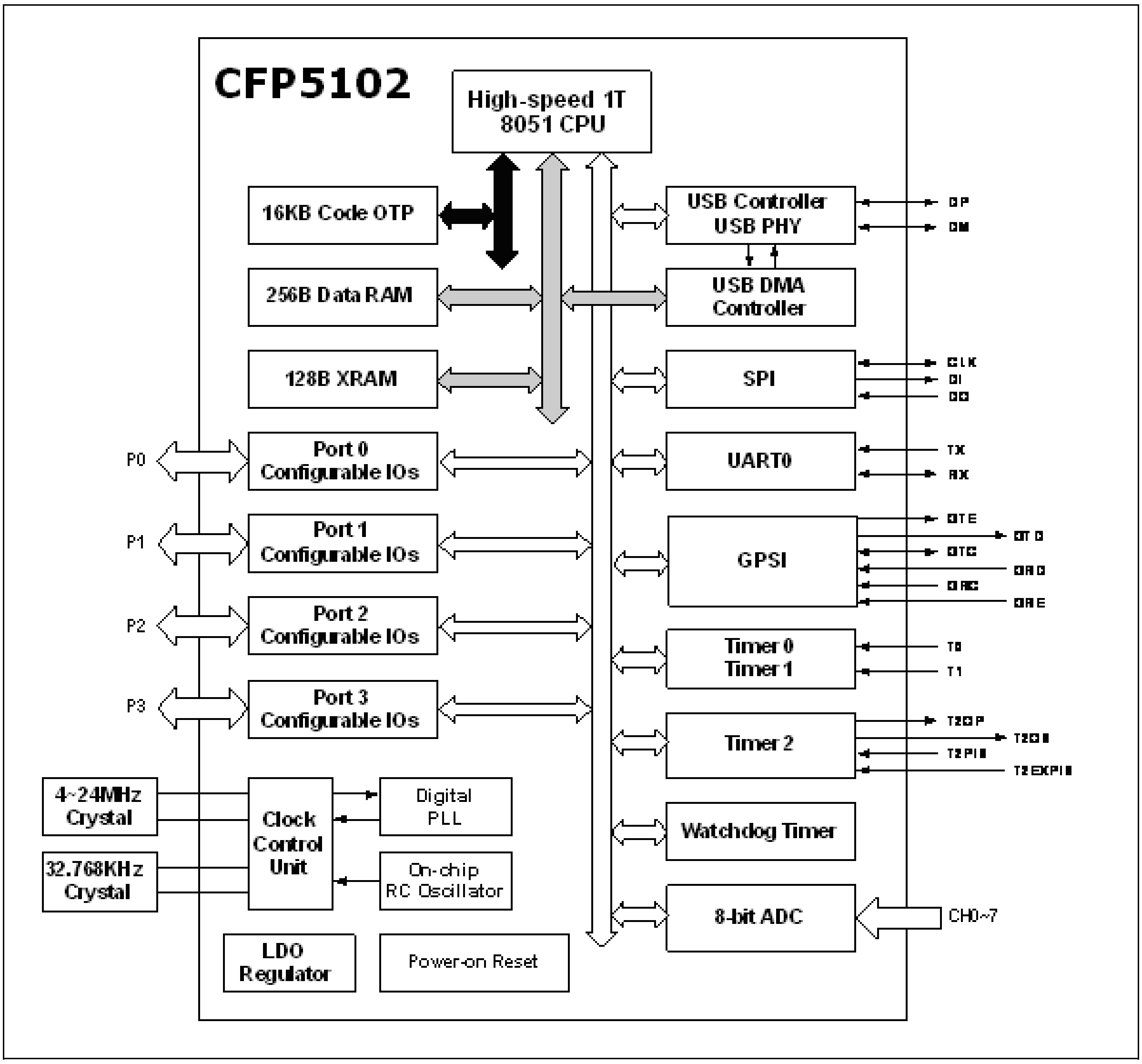
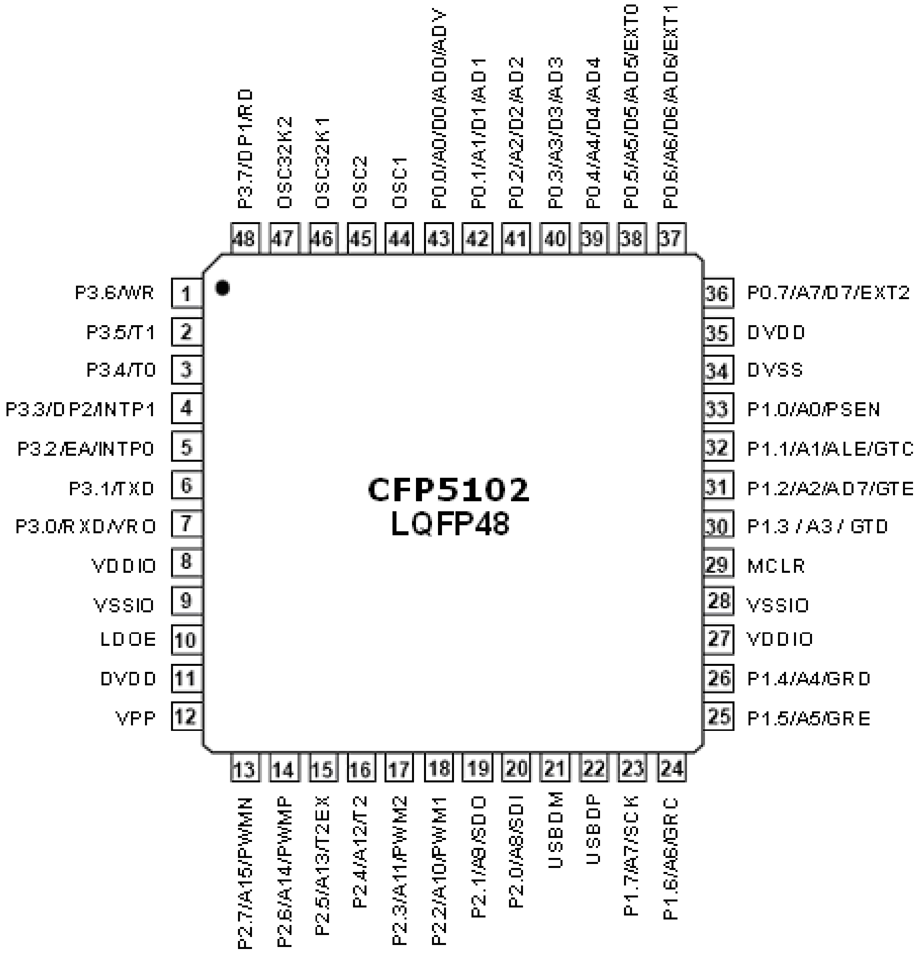


Figure 1-1: Block Diagram

2 Pin Information

2.1 Pin Assignment



2.2 Pin Description

Symbol	Pin	Type[note1]	Description
P0.0 / A0 / D0 / AD0 / ADV	43	I/O O O I I	**P0.0** – port 0 bit 0 **A0** – external memory address bus bit 0 [note2] **D0** – external memory data bus bit 0 **AD0** – analog-to-digital converter channel 0 **ADV** – analog-to-digital converter voltage reference
P0.1 / A1 / D1 / AD1	42	I/O O O I	**P0.1** – port 0 bit 1 **A1** – external memory address bus bit 1 [note2] **D1** – external memory data bus bit 1 **AD1** – analog-to-digital converter channel 1
P0.2 / A2 / D2 / AD2	41	I/O O O I	**P0.2** – port 0 bit 2 **A2** – external memory address bus bit 2 [note2] **D2** – external memory data bus bit 2 **AD2** – analog-to-digital converter channel 2
P0.3 / A3 / D3 / AD3	40	I/O O O I	**P0.3** – port 0 bit 3 **A3** – external memory address bus bit 3 [note2] **D3** – external memory data bus bit 3 **AD3** – analog-to-digital converter channel 3
P0.4 / A4 / D4 / AD4	39	I/O O O I	**P0.4** – port 0 bit 4 **A4** – external memory address bus bit 4 [note2] **D4** – external memory data bus bit 4 **AD4** – analog-to-digital converter channel 4
P0.5 / A5 / D5 / AD5 / EXT0	38	I/O O O I I	**P0.5** – port 0 bit 5 **A5** – external memory address bus bit 5 [note2] **D5** – external memory data bus bit 5 **AD5** – analog-to-digital converter channel 5 **EXT0** – external interrupt 0
P0.6 / A6 / D6 / AD6 / EXT1	37	I/O O O I I	**P0.6** – port 0 bit 6 **A6** – external memory address bus bit 6 [note2] **D6** – external memory data bus bit 6 **AD6** – analog-to-digital converter channel 6 **EXT1** – external interrupt 1
P0.7 / A7 / D7 / EXT2	36	I/O O O I	**P0.7** – port 0 bit 7 **A7** – external memory address bus bit 7 [note2] **D7** – external memory data bus bit 7 **EXT2** – external interrupt 2
P1.0 / A0 / PSEN	33	I/O O O	**P1.0** – port 1 bit 0 **A0** – external memory address bus bit 0 [note3] **PSEN** – external memory program select enable
P1.1 / A1 / ALE / GTC	32	I/O O O O	**P1.1** – port 1 bit 1 **A1** – external memory address bus bit 1 [note3] **ALE** – external memory address latch enable **GTC** – GPSI transmit clock
P1.2 / A2 / AD7 / GTE	31	I/O O I O	**P1.2** – port 1 bit 2 **A2** – external memory address bus bit 2 [note3] **AD7** – analog-to-digital converter channel 7 **GTE** – GPSI transmit enable
P1.3 / A3 / GTD	30	I/O O O	**P1.3** – port 1 bit 3 **A3** – external memory address bus bit 3 [note3] **GTD** – GPSI transmit data
P1.4 / A4 / GRD	26	I/O O I	**P1.4** – port 1 bit 4 **A4** – external memory address bus bit 4 [note3] **GRD** – GPSI receive data
P1.5 / A5 / GRE	25	I/O O	**P1.5** – port 1 bit 5 **A5** – external memory address bus bit 5 [note3]

		I	**GRE** – GPSI receive enable
P1.6 / A6 / GRC	24	I/O O I	**P1.6** – port 1 bit 6 **A6** – external memory address bus bit 6 [note3] **GRC** – GPSI receive clock
P1.7 / A7 / SCK	23	I/O O I/O	**P1.7** – port 1 bit 7 **A7** – external memory address bus bit 7[note3] **SCK** – SPI clock
P2.0 / A8 / SDI	20	I/O O I/O	**P2.0** – port 2 bit 0 **A8** – external memory address bus bit 8 **SDI** – SPI data input [note4]
P2.1 / A9 / SDO	19	I/O O O	**P2.1** – port 2 bit 1 **A9** – external memory address bus bit 9 **SDO** – SPI data output [note4]
P2.2 / A10 / PWM1	18	I/O O O	**P2.2** – port 2 bit 2 **A10** – external memory address bus bit 10 **PWM1** – 8-bit PWM channel 1
P2.3 / A11 / PWM2	17	I/O O O	**P2.3** – port 2 bit 3 **A11** – external memory address bus bit 11 **PWM2** – 8-bit PWM channel 2
P2.4 / A12 / T2	16	I/O O I	**P2.4** – port 2 bit 4 **A12** – external memory address bus bit 12 **T2** –timer/counter 2 external count input
P2.5 / A13 / T2EX	15	I/O O I	**P2.5** – port 2 bit 5 **A13** – external memory address bus bit 13 **T2EX** –timer/counter 2 reload/capture/direction control
P2.6 / A14 / PWMP	14	I/O O O	**P2.6** – port 2 bit 6 **A14** – external memory address bus bit 14 **PWMP** – 16-bit PWM positive channel
P2.7 / A15 / PWMN	13	I/O O O	**P2.7** – port 2 bit 7 **A15** – external memory address bus bit 15 **PWMN** – 16-bit PWM negative channel
P3.0 / RXD / VRO	7	I/O I O	**P3.0** – port 3 bit 0 **RXD** – serial port input **VRO** – voltage reference out (1.25V)
P3.1 / TXD	6	I/O O	**P3.1** – port 3 bit 1 **TXD** – serial port output
P3.2 / EA / INTP0	5	I/O I I	**P3.2** – port 3 bit 2 (pull-up enable by default) **INTP0** – port interrupt 0 **EA** – external access enable (active LOW) EA must be externally held LOW to enable the device to fetch code from external program memory locations. If EA is held HIGH, the device executes from internal program memory. The value on the EA pin is latched when MCLR is released and any subsequent changes have no effect.
P3.3 / DP2 / INTP1	4	I/O I I	**P3.3** – port 3 bit 3 (pull-up enable by default) **INTP1** – port interrupt 1 **DP2** – in-system programming pin 2 (active LOW) DP2 must be externally held LOW to enable in-system programming. The value on the DP2 pin is latched when MCLR is released and any subsequent changes have no effect.
P3.4 / T0	3	I/O I	**P3.4** – port 3 bit 4 **T0** – timer/counter 0 capture/compare/count input
P3.5 / T1	2	I/O I	**P3.5** – port 3 bit 5 **T1** – timer/counter 1 capture/compare/count input
P3.6 / WR	1	I/O I	**P3.6** – port 3 bit 6 **WR** – external data memory write strobe (active LOW)
P3.7 / DP1 / RD	48	I/O	**P3.7** – port 3 bit 7

		I I	**DP1** – in-system programming pin 1 Cooperate with DP2 to operate in-system programming **RD** – external data memory read strobe (active LOW)
USBDP	22	I/O	**USBDP** – USB data (positive)
USBDM	21	I/O	**USBDP** – USB data (negative)
LDOE	10	O	**LDOE** – on-chip regulator (LDO) enable (active LOW) When LDOE is tied to HIGH, the LDO will be disabled. It is recommended to supply power to chip through external power source into DVDD pad. When LDOE is tied to LOW, the LDO will be enabled and will regulate the voltage to 3.3V after a short start-up time.
OSC1	44	I	**OSC1** – high speed crystal pin 1 Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
OSC2	45	O	**OSC2** – high speed crystal pin 2 Output from the inverting oscillator amplifier.
OSC32K1	46	I	**OSC32K1** – 32,768 Hz crystal pin 1 Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
OSC32K2	47	O	**OSC32K2** –32,768 Hz crystal pin 1 Output from the inverting oscillator amplifier.
MCLR	29	I	**MCLR** – master reset (active LOW) A LOW on this pin for 8 ms resets the device.
VPP	12	P	**VPP** – OTP programming voltage supply
DVDD	11,35	P	**DVDD** – voltage supply for core When LDO is enabled, this pad is served for compensation for LDO output. Please refer to Section 6.3.1 for external circuitry.
DVSS	34	P	**DVSS** – supply ground for core
VDDIO	8,27	P	**VDDIO** – voltage supply for IO and LDO
VSSIO	9,28	P	**VSSIO** – supply ground for IO and LDO

Note1: O means output pin
I means input pin
I/O means input and output pin
P means power pin
Note2: For external memory interface in multiplexed address mode
Note3: For external memory interface in non-multiplexed address mode

3 Programming Model

3.1 Memory Organization

CFP5102, like 8052 contains several separated memory spaces, namely CODE, DATA, XDATA, STACK and REGISTER. These memory spaces are accessed by different addressing and by different instructions. OTP, SRAM and peripheral-control registers are mapped to these spaces. To suit for the memory requirement of different programs, CFP5102 provides enhanced configurable memory architecture. Users can select suitable memory mapping by programming the control register. The memory models are discussed in the following sections.

3.2 Program Memory (CODE)

The memory space CODE is a 64K-byte space dedicated for programs and constants. The whole space is addressed by program counter (PC) and also by two 16-bit data pointers (DPTR0 and DPTR1) using MOVC instruction. In CFP5102, the program can be implemented on-chip (in 16K-byte OTP memory), off-chip or as a combination. For applications with program excesses 16K bytes, CFP5102 provides an external memory interface (EMI) to expand the program memory through external storage, such as NOR Flash, ROM, EEPROM and etc.

The memory mapping in CODE is illustrated in Figure 3-1. In the default setting, the on-chip 16K bytes OTP memory is mapped to 0x0000 to 0x3FFF of CODE. The rest of the addressable area is not defined. Users should restrict the program within the mapped area to get rid of unpredictable operations.

CFP5102 supports two ways to enable external memory interface: (1) setting register by the program in OTP; (2) pull down the pin EA externally. In the former case, the program in OTP is working like a boot-loader, the setup program, reset program, interrupt subroutines, libraries and frequently called functions are stored in the internal OTP. As the CPU is able to work at full speed when accessing OTP, this arrangement is tuned for these frequently used parts of the program. The EMI is enabled by writing a ‘1’ to bit 2 of DPCON (EMIEN). The details of the interconnection, timing and operation of EMI are discussed in Section 4

When users intend to execute program in external storage without programming the on-chip OTP memory, the only way is to configure CFP5102 by pulling down the pin EA during system reset until the execution of the first instruction. To stop hacking of OTP through this configuration, OTP is disabled and unmapped to any memory space. No any instruction allows accessing OTP in this mode. Further security measurements are discussed in Section 8.

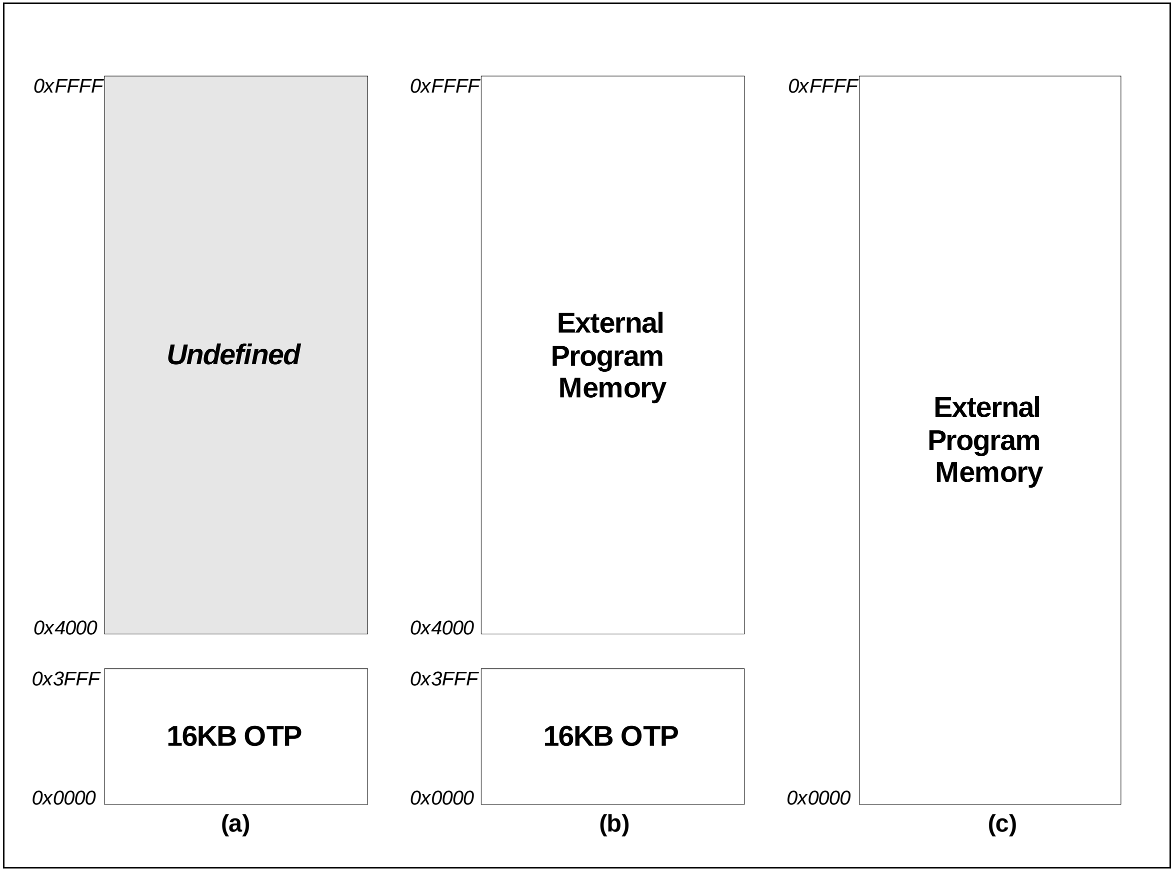


Figure 3-1: (a) Default mapping of CODE. Access undefined area causes unpredictable result. (b) Mapping of CODE when external memory interface is enabled. (c) Special mapping of CODE when enabling external memory interface by forcing EA to low during system reset.

Table 3-1: Summary of CODE memory settings

EA	EMIEN	Memory Mapping
1	0	*0x0000 – 0x3FFF: OTP* *0x4000 – 0xFFFF: Undefined*
1	1	*0x0000 – 0x3FFF: OTP* *0x4000 – 0xFFFF: EMI*
0	X	*0x0000 – 0xFFFF: EMI*

In standard 8051/8052 architecture, the reset program is located at 0x0000 of CODE. The interrupt subroutines starts from 0x0003, allocated 8 bytes for each interrupt subroutine. CFP5102 provide a remapping option to relocate the reset and interrupt entries to upper 32K-byte area of CODE. Programmers are freed from the limitation of 8051/8052 to do flexible memory planning, in addition to do fast switching between two different reset and interrupt handling schemes. To remap the reset and interrupt entries, programmers have to write '1' to bit 4 of CCON (ISRM). These setting cannot be cancelled by watchdog reset and wakeup reset. Only power-on reset clears this bit to default value, logic '0'.

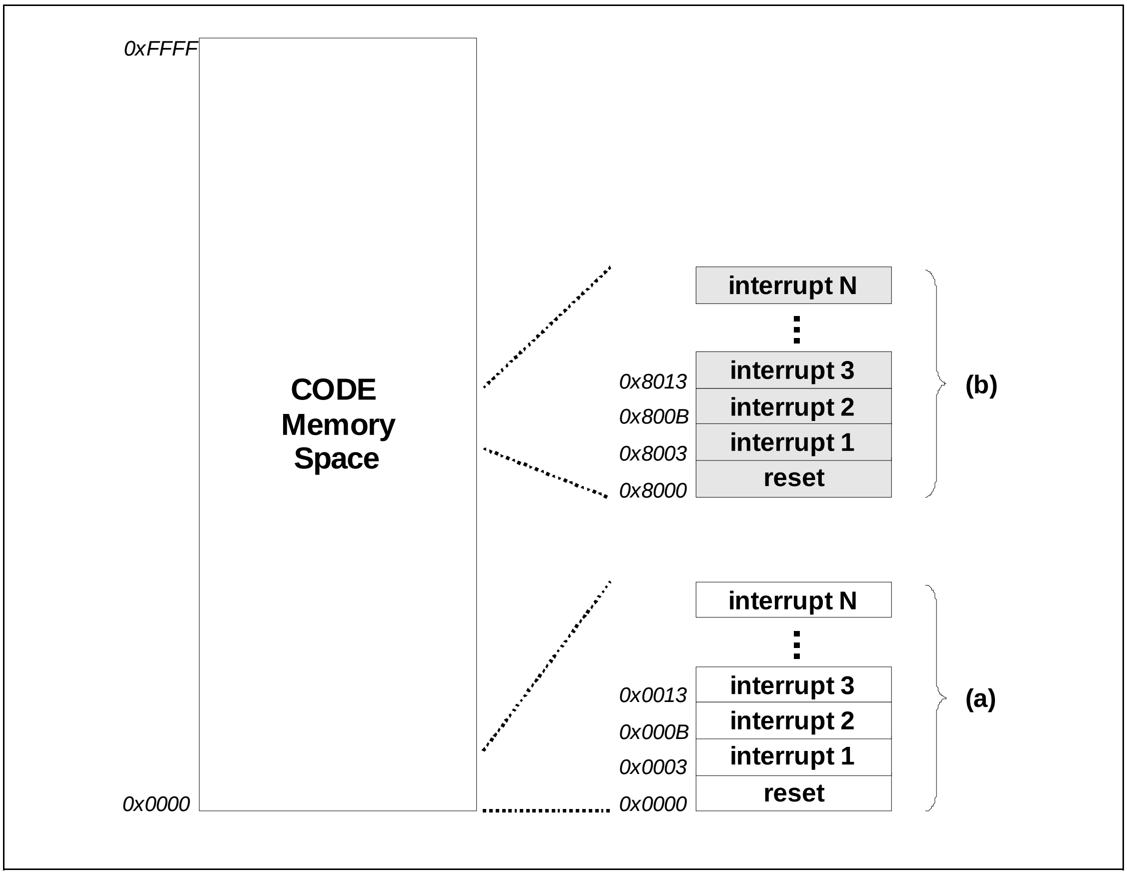


Figure 3-2: (a) The location of reset and interrupt entries by default. (b) The location of reset and interrupt entries when remapping is enabled

3.3 Data Memory (DATA)

Data memory space (DATA) contains 256K 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.

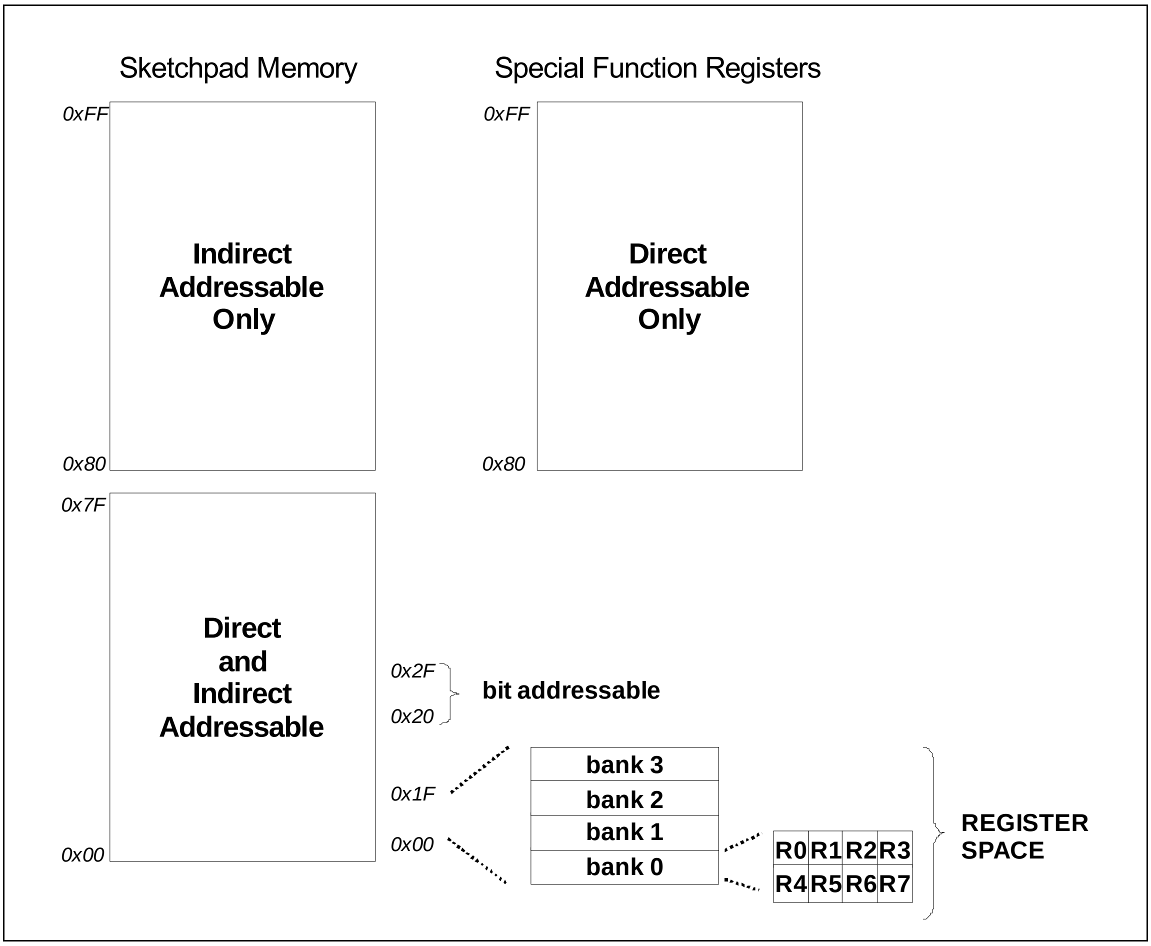


Figure 3-3: The mapping of DATA and REGISTER

3.4 General Purpose Registers (REGISTER)

General purpose registers R0 through R7 link to 32 bytes of internal data memory in the way that allows quick, 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 3-2 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 3-2: Selection of Register Bank

RS0	RS1	Bank	Mapping Address to DATA
0	0	0	0x00 – 0x07
0	1	1	0x08 – 0x0F
1	0	2	0x10 – 0x17
1	1	3	0x18 – 0x1F

3.5 Extended Data Memory (XDATA)

To provide a unified linear memory model, CFP5102 organizes internal data memory and external data memory that out of the scope of DATA into extended data memory space (XDATA). Total 64K bytes of XDATA are addressed through instruction *MOVX* using two 16-bit data pointers, DPTR0 and DPTR1. XDATA contains 128 bytes on-chip SRAM, 64K bytes off-chip SRAM and also 16K bytes OTP. Constant data structures stored in OTP can be manipulated uniformly as data structures in data memory through instruction MOVX.

Table 3-3: Summary of XDATA memory settings

EA	EMIEN	XMAP	Memory Mapping
1	0	0	*0x0000 – 0x007F: On-chip SRAM (XRAM)* *0x0080 – 0xFFFF: Undefined*
1	0	1	*0x0000 – 0x007F: XRAM* *0x0080 – 0x7FFF: Undefined* *0x8000 – 0x9FFF: OTP* *0xA000 – 0xFFFF: Undefined*
1	1	0	*0x0000 – 0x007F: XRAM* *0x0080 – 0xFFFF: EMI*
1	1	1	*0x0000 – 0x007F: XRAM* *0x0080 – 0x7FFF: EMI* *0x8000 – 0x9FFF: OTP* *0xA000 – 0xFFFF: EMI*
0	0	X	*0x0000 – 0x007F: XRAM* *0x0080 – 0xFFFF: Undefined*
0	1	X	*0x0000 – 0x007F: XRAM* *0x0080 – 0xFFFF: EMI*

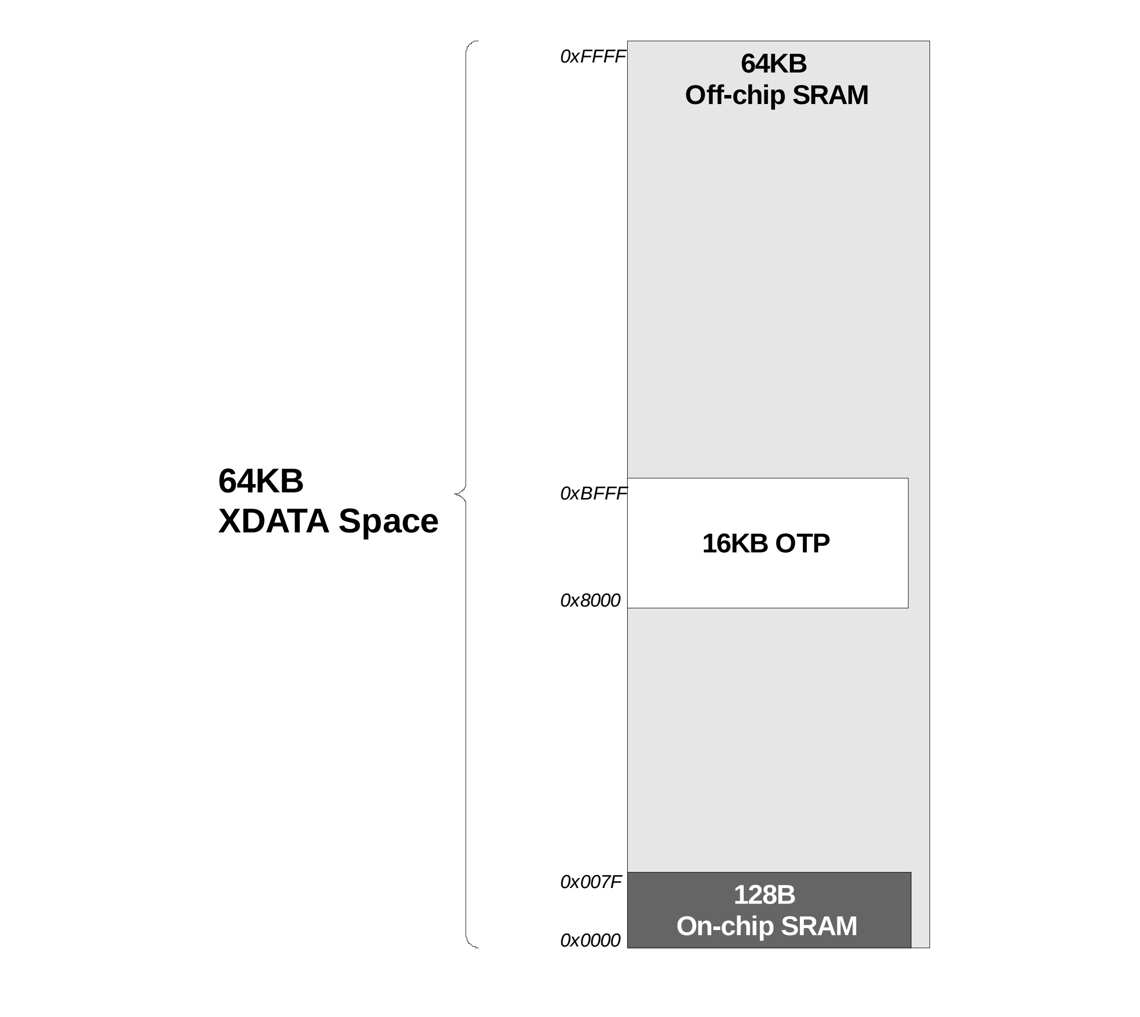


Figure 3-4: The mapping of XDATA

3.6 Program Stack Memory (STACK)

The classic 8051/8052 provides a stack for program counter storage during interrupts, subroutine calls and also for temporary data. The stack is implemented in DATA memory, therefore limited to 128 bytes in 8051 or 256 bytes in 8052. The limitation thus hampers developing large scale systems on 8051.

To breakthrough the burden to support nowadays complex systems, CFP5102 extends the program stack memory to XDATA, which means the size of the STACK can be up to 64K bytes when using external memory. The 16-bit STACK pointer is formed by combining register SPH and SP. (refer to Section 3.10)

Table 3-4: Summary of STACK memory settings

SPXSEL	EMIEN	Memory Mapping
0	0	*0x0000 – 0x00FF: Sketchpad memory (DRAM)*
1	0	*0x0000 – 0x007F: On-chip SRAM (XRAM)*
1	1	*0x0000 – 0x007F: XRAM* *0x0080 – 0xFFFF: EMI*

3.7 CPU Timing

CFP5102 is a high performance MCU. It executes most instruction in 1 cycle, in contrast to 12 cycles in classic 8051/8052. To fetch instructions, CFP5102 reads 1 byte at each cycle. To take all into account, the cycle required to complete an instruction can be calculated as follows:

Cycle required = number of bytes + execution cycle – 1

For example, the instruction ADDC A, Rn takes 1 cycle; the instruction MOVC A, @A+PC takes 3 cycles

3.8 Instruction Set Summary

#	Mnemonics		Description	Number of Bytes	Execution Cycle
			Arithmetic Instructions		
1	ADD	A, Rn	Add register to A	1	1
2	ADD	A, direct	Add direct byte to A	2	1
3	ADD	A, @Ri	Add indirect RAM to A	1	1
4	ADD	A, #data	Add immediate to A	2	1
5	ADDC	A, Rn	Add register to A with carry	1	1
6	ADDC	A, direct	Add direct byte to A with carry	2	1
7	ADDC	A, @Ri	Add indirect RAM to A with carry	1	1
8	ADDC	A, #data	Add immediate to A with carry	2	1
9	SUBB	A, Rn	Subtract register from A with borrow	1	1
10	SUBB	A, direct	Subtract direct byte from A with borrow	2	1
11	SUBB	A, @Ri	Subtract indirect RAM from A with borrow	1	1
12	SUBB	A, #data	Subtract immediate from A with borrow	2	1
13	INC	A	Increment A	1	1
14	INC	Rn	Increment register	1	1
15	INC	direct	Increment direct byte	2	1
16	INC	@Ri	Increment indirect RAM	1	1
17	DEC	A	Decrement A	1	1
18	DEC	Rn	Decrement register	1	1
19	DEC	direct	Decrement direct byte	2	1
20	DEC	@Ri	Decrement indirect RAM	1	1
21	INC	DPTR	Increment Data Pointer	1	1
22	MUL	AB	Multiply A and B	1	1
23	DIV	AB	Divide A by B	1	1
24	DA	A	Decimal adjust A	1	1
			Logical Instructions		
25	ANL	A, Rn	AND register to A	1	1
26	ANL	A, direct	AND direct byte to A	2	1
27	ANL	A, @Ri	AND indirect RAM to A	1	1

28	ANL A, #data	AND immediate to A	2	1
29	ANL direct, A	AND A to direct byte	2	1
30	ANL direct, #data	AND immediate to direct byte	3	1
31	ORL A, Rn	OR register to A	1	1
32	ORL A, direct	OR direct byte to A	2	1
33	ORL A, @Ri	OR indirect RAM to A	1	1
34	ORL A, #data	OR immediate to A	2	1
35	ORL direct, A	OR A to direct byte	2	1
36	ORL direct, #data	OR immediate to direct byte	3	1
37	XRL A, Rn	XOR register to A	1	1
38	XRL A, direct	XOR direct byte to A	2	1
39	XRL A, @Ri	XOR indirect RAM to A	1	1
40	XRL A, #data	XOR immediate to A	2	1
41	XRL direct, A	XOR A to direct byte	2	1
42	XRL direct, #data	XOR immediate to direct byte	3	1
43	CLR A	Clear A	1	1
44	CPL A	Complement A	1	1
45	RL A	Rotate A left	1	1
46	RLC A	Rotate A left through carry	1	1
47	RR A	Rotate A right	1	1
48	RRC A	Rotate A right through carry	1	1
49	SWAP A	Swap nibbles of A	1	1
		Data Transfer Instructions		
50	MOV A, Rn	Move register to A	1	1
51	MOV A, direct	Move direct byte to A	2	1
52	MOV A, @Ri	Move indirect RAM to A	1	1
53	MOV A, #data	Move immediate to A	2	1
54	MOV Rn, A	Move A to register	1	1
55	MOV Rn, direct	Move direct byte to register	2	1
56	MOV Rn, #data	Move immediate to register	2	1
57	MOV direct, A	Move A to direct byte	2	1
58	MOV direct, Rn	Move register to direct byte	2	1
59	MOV direct, direct	Move direct byte to direct byte	3	1
60	MOV direct, @Ri	Move indirect RAM to direct byte	2	1
61	MOV direct, #data	Move immediate to direct byte	3	1
62	MOV @Ri, A	Move A to indirect RAM	1	1
63	MOV @Ri, direct	Move direct byte to indirect RAM	2	1
64	MOV @Ri, #data	Move immediate to indirect RAM	2	1
65	MOV DPTR, #data	Load DPTR with 16-bit constant	3	1
66	MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
67	MOVC A, @A+PC	Move code byte relative PC to A	1	3
68	MOVX A, @Ri	Move external data (8-bit address) to A	1	1
69	MOVX @Ri, A	Move A to external data (8-bit address)	1	1
70	MOVX A, @DPTR	Move external data (16-bit address) to A	1	1
71	MOVX @DPTR, A	Move A to external data (16-bit address)	1	1
72	PUSH direct	Push direct byte onto stack	2	1
73	POP direct	Pop direct byte from stack	2	1

74	XCH A, Rn	Exchange register with A	1	1
75	XCH A, direct	Exchange direct byte with A	2	1
76	XCH A, @Ri	Exchange indirect RAM with A	1	1
77	XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	1
		Bit Manipulation Instructions		
78	CLR C	Clear carry	1	1
79	CLR bit	Clear direct bit	2	1
80	SETB C	Set carry	1	1
81	SETB bit	Set direct bit	2	1
82	CPL C	Complement carry	1	1
83	CPL bit	Complement direct bit	2	1
84	ANL C, bit	AND direct bit to carry	2	1
85	ANL C, /bit	AND complement of direct bit to carry	2	1
86	ORL C, bit	OR direct bit to carry	2	1
87	ORL C, /bit	OR complement of direct bit to carry	2	1
88	MOV C, bit	Move direct bit to Carry	2	1
89	MOV bit, C	Move Carry to direct bit	2	1
90	JC rel code	Jump if carry is set	2	1 | 3*
91	JNC rel code	Jump if carry is not set	2	1 | 3*
92	JB bit, rel code	Jump if direct bit is set	3	1 | 3*
93	JNB bit, rel code	Jump if direct bit is not set	3	1 | 3*
94	JBC bit, rel code	Jump if direct bit is set and clear bit	3	1 | 3*
		Program Branching Instructions		
95	ACALL page code	Absolute subroutine call	2	3
96	LCALL long code	Long subroutine call	3	3
97	RET	Return from subroutine	1	3
98	RETI	Return from interrupt	1	3
99	AJMP page code	Absolute jump	2	3
100	LJMP long code	Long jump	3	3
101	SJMP rel addr	Short jump (relative address)	2	3
102	JMP @A+DPTR	Jump indirect relative to DPTR	1	3
103	JZ rel code	Jump if A equals zero	2	1 | 3*
104	JNZ rel code	Jump if A does not equal zero	2	1 | 3*
105	CJNE A, direct, rel code	Compare direct byte to A and jump if not equal	3	1 | 3*
106	CJNE A, #data, rel code	Compare immediate to A and jump if not equal	3	1 | 3*
107	CJNE Rn, #data, rel code	Compare immediate to Register and jump if not equal	3	1 | 3*
108	CJNE @Ri, #data, rel code	Compare immediate to indirect and jump if not equal	3	1 | 3*
109	DJNZ Rn, rel code	Decrement Register and jump if not zero	2	1 | 3*
110	DJNZ direct, rel code	Decrement direct byte and jump if not zero	3	1 | 3*
111	NOP	No operation	1	1
Note: * indicates a branch occurs				

Notes on Registers, Operands and Addressing Modes:

Rn - Register R0-R7 of the currently selected register bank.
@Ri - Data RAM location addressed indirectly through R0 or R1.
rel - 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's address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).
#data - 8-bit constant
#data16 - 16-bit constant
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 64K-byte program memory space.

3.9 Dual Data Pointers

To facilitate data movement in XDATA, CFP5102 provides dual data pointers. Two independent data pointers are especially useful when moving, copying and manipulating large data types, such arrays, structures and unions. Based on this feature, CFP5102 saves many operations in manipulating pointer, comparing to single data pointer classic 8051/8052. The additional data pointer can be selected through register DPCON. No additional instruction is introduced to distinguish these 2 pointers.

DPCON: Data pointer Control Register
TYPE: R/W
ADDRESS: 0x86

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	DPID1	DPID0	DPTSE	DPAID	-	EMIEN	SPXSEL	DPS
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
DPID1	RW	Replace instruction INC DPTR1 by DEC DPTR1	0: INC 1: DEC
DPID0	RW	Replace instruction INC DPTR0 by DEC DPTR0	0: INC 1: DEC
DPTSE	RW	Automatic DPTR select toggle enable. Flip to other DPTR automatically after any MOVX or MOVC instructions	0: disable 1: enable
DPTAID	RW	Automatic DPTR increment/decrement enable bit. Increase/decrease DPTR automatically after any MOVX or MOVC instructions	0: disable 1: enable
EMIEN	RW	External memory interface enable bit.	0: disable 1: enable
SPXSEL	RW	Stack point XRAM mapping select bit. Map the stack point to XRAM space	0: map to DRAM 1: map to XRAM
DPS	RW	Data pointer select	0: DPTR0 1: DPTR1

DPH: Data Pointer 0 High Byte Register
TYPE: R/W
ADDRESS: 0x83

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	DPTR[15:8]							
Default	00000000							

DPL: Data Pointer 0 High Byte Register
TYPE: R/W
ADDRESS: 0x82

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	DPTR[7:0]							
Default	00000000							

DPH1: **Data Pointer 1 High Byte Register**
TYPE: **R / W**
ADDRESS: **0x85**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	DPTR1[15:8]							
Default	00000000							

DPL1: **Data Pointer 1 High Byte Register**
TYPE: **R / W**
ADDRESS: **0x84**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	DPTR1[7:0]							
Default	00000000							

3.10 CPU Registers

CCON: **CPU Control Register**
TYPE: **R / W**
ADDRESS: **0xAB**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	PROT	-	LPM	ISRM	XMAP	-	-	-
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
PROT	RW	OTP code protection enable bit. When set to '1', MOVC to OTP memory is disabled. This bit can clear by reset sources only	0: disable 1: enable
LPM	RW	Low power mode enable bit. Reduce the power consumption by OTP memory. Caution: this mode can only be enabled when system clock is under 10MHz	0: disable 1: enable
ISRM	RW	Interrupt subroutine mapping bit. 0x8000 offset is added to the address of the interrupt subroutine when this bit is set	0: no offset 1: add 0x8000 offset
XMAP	RW	OTP mapping to XRAM enable bit. Map OTP memory to XRAM space when set to '1'. User can use MOVX instruction to access	0: disable 1: enable

PSW: **Processor Status Word**
TYPE: **R / W**
ADDRESS: **0xD0**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	CY	AC	F0	RS1	RS0	OV	F1	P
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
P	R	Odd parity check of ACC	
OV	RW	Overflow flag	
F1, F0	RW	User-defined bits	
RS1, RS0	RW	Register bank select	
AC	RW	Auxiliary carry flag	
CY	RW	Carry flag	

SP: **Stack Pointer Register**
TYPE: **R / W**
ADDRESS: **0x81**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SP							
Default	00000111							

SPH: **Stack Pointer High Byte Register**
TYPE: **R / W**
ADDRESS: **0x9B**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SPH							
Default	00000000							

Description	Setting
SPH and **SP** registers are combined to form a 16-bit stack pointer when SPXSEL in DPCON is '1'	

ACC: **Accumulator Register**
TYPE: **R / W**
ADDRESS: **0xE0**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	ACC							
Default	00000000							

B: **B Register**
TYPE: **R / W**
ADDRESS: **0xF0**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	B							
Default	00000000							

3.11 Special Function Registers (SFR)

3.11.1 SFR table listed in alphabetical order

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’.

Register Name	Description	Address	Bit Functions								Reset
			MSB							LSB	
ACC	Accumulator	0xE0									0000 0000
ADCB	ADC buffer	0xEE									XXXX XXXX
ADCCON0	ADC control 0	0xF4	ADCPND	INBUF	REFS	TM	TC[1]	TC[0]	GO	ADCEN	0000 0000
ADCCON1	ADC control 1	0xF5	CM	GT	-	REFBE	-	AS[2]	AS[1]	AS[0]	0000 0000
ADCSC	ADC sampling clock control	0xF3									0000 0000
ADCT	ADC threshold	0xF1									XXXX XXXX
B	B register	0xF0									0000 0000
CCON	CPU control	0xAB	PROT	-	LPM	ISRM	XMAP	-	-	-	0000 0000
CKCON0	Clock control 0	0x91	-	-	LPOSC	PLLEN	-	SCPLL	SCS	SCKD	0000 0000
CKCON1	Clock control 1	0x92	FFCKE	32CKE	RTCKE	MCKE	PCKE	-	TCKE	UCKE	0001 1110
CKCON2	Clock control2	0x93	-	-	-	-	-	CKS	32CE	HSCE	0000 1101
CLKDIV	Clock divider	0x94									0000 0000
DP1H	Data pointer 1 high byte	0x85									0000 0000
DP1L	Data pointer 1 low byte	0x84									0000 0000
DPCON	Data pointer control	0x86	DPID1	DPID0	DPTSE	DPAID	-	EMIEN	SPXSEL	DPS	0000 0000
DPH	Data pointer high byte	0x83									0000 0000
DPL	Data pointer low byte	0x82									0000 0000
EMICON1	EMI control 1	0x8E	RW[3]	RW[2]	RW[1]	RW[0]	HD[1]	HD[0]	SET[1]	SET[0]	1111 1111
EMICON2	EMI control 2	0x8F	-	SZ[1]	SZ[0]	-	PSON	MUX	WRON	AL	0100 1111
FIFOCON	FIFO interface control	0x9C	FEN	ONLY	-	-	-	-	WR	RD	0000 0000
FIFODATA	FIFO data	0x9F									0000 0000
FIFOPTR	FIFO pointer	0x9D									0000 0000
FSET	Special function setting	0xBF	-*	T0DD	OCSEL[1]	OCSEL[0]	-*	T1DD	EWDT	WDTRST	0100 1110
GPSIBAUD	GPSI baud rate divider	0xC5									0000 0000
GPSIBUF	GPSI data buffer	0xC4									0000 0000

GPSICON0	GPSI control 0	0xC0	-	-	GPSIRXIE	GPSITXIE	TXCLKSEL	RXEDSEL	TXEDSEL	GPSIEN	0000 0000
GPSICON1	GPSI control 1	0xC1	RXBUF FULL	TXBUF EPTY	RXHDREN	TXHDREN	EOP	RXENOFF	TXMODE[1]	TXMODE[0]	0000 0000
GPSICON2	GPSI control 2	0xC2	HDRRXCLR	HDRTXCLR	GPSIEOPIE	GPSIRXHIE	GPSITXHIE	HEAD32	HDRRX MATCH	HDRTX LOAD	0000 0000
GPSIHDR	GPSI header	0xC3									0000 0000
IEN0	Interrupt enable 0	0xA8	EA	ERT	ET2	ES	ET1	EX1	ET0	EX0	0000 0000
IEN1	Interrupt enable 1	0xA9	-	-	EGPSI	EADC	EUD	EUI	ESI	EPW	0000 0000
IP0	Interrupt priority 0	0xB8	-	PRT	PT2	PS	PT1	PX1	PT0	PX0	0000 0000
IP1	Interrupt priority 1	0xB9	-	-	PGPSI	PADC	PUD	PUI	PSI	PPW	0000 0000
P0	Port 0	0x80									XXXX XXXX
P0AIE	Port 0 analog input enable	0xE1									0000 0000
P0DIR	Port 0 direction control	0xAC									1111 1111
P0OD	Port 0 open-drain mode	0xA4									0000 0000
P0PUP	Port 0 pull-up control	0xB1									0000 0000
P1	Port 1	0x90									XXXX XXXX
P1AIE	Port 1 analog input enable	0xE2									0000 0000
P1DIR	Port 1 direction control	0xAD									1111 1111
P1OD	Port 1 open-drain mode	0xA5									0000 0000
P1PUP	Port 1 pull-up control	0xB2									0000 0000
P2	Port 2	0xA0									XXXX XXXX
P2DIR	Port 2 direction control	0xAE									1111 1111
P2OD	Port 2 open-drain mode	0xA6									0000 0000
P2PUP	Port 2 pull-up control	0xB3									0000 0000
P3	Port 3	0xB0									XXXX XXXX
P3AIE	Port 3 analog input enable	0xE4									0000 0000
P3DIR	Port 3 direction control	0xAF									1111 1111
P3OD	Port 3 open-drain mode	0xA7									0000 0000
P3PUP	Port 3 pull-up control	0xB4									0000 1100
PCON	Power control	0x87	SMOD	EA	HLT	POF	-	-	PD	IDL	0X01 0000
PSW	Program status word	0xD0	CY	AX	F0	RS1	RS0	OV	F1	P	0000 0000
PSYN	Port synchronizer enable	0xB6	-	-	-	-	P3SYN	P2SYN	P1SYN	P0SYN	0000 0000
RCAP2H	Timer 2 Capture/Reload/PWM duty cycle high byte	0xCB									0000 0000
RCAP2L	Timer 2 Capture/Reload/PWM duty cycle low byte	0xCA									0000 0000
RTCON	Real time control	0xD8	-	RTWK	-	-	RTWE	-	-	-	0000 0000

RTDIV	Real time clock divider control	0xD4									0000 0000
SBUF	Serial port buffer	0x99									0000 0000
SCON	Serial port control	0x98	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000
SP	Stack pointer	0x81									0000 0111
SPH	Stack pointer high byte	0x9B									0000 0000
SPIBAUD	SPI baud control	0x96									0000 0000
SPIBUF	SPI data buffer	0x97									0000 0000
SPICON	SPI control	0x95	SPIPND	SM	RTS	WS	RXO	SMP	IDS	SPIEN	0000 0000
T2CON	Timer 2 control	0xC8	TF2	EXF2	-	-	EXEN2	TR2	C/T2	CP/RL2	0000 0000
T2MOD	Timer 2 mode	0xC9	-	PWM8	T2CLK[1]	T2CLK[0]	PWM16N	PWM16P	T2OE	DCEN	0000 0000
TCON	Timer 0/1 control	0x88	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000
TH0	Timer 0 high byte	0x8C									0000 0000
TH1	Timer 1 high byte	0x8D									0000 0000
TH2	Timer 2 counter high byte	0xCD									0000 0000
TL0	Timer 0 low byte	0x8A									0000 0000
TL1	Timer 1 low byte	0x8B									0000 0000
TL2	Timer 2 counter low byte	0xCC									0000 0000
TMOD	Timer 0/1 mode	0x89	GATE1	C/T1	T1M[1]	T1M[0]	GATE0	C/T0	T0M[1]	T0M[0]	0000 0000
TPR2H	Timer 2 PWM period high byte	0xCF									0000 0000
TPR2L	Timer 2 PWM period low byte	0xCE									0000 0000
USBADR	USB access address	0xFB	-	-	USBADR[5:0]						00XX XXXX
USBCON0	USB control 0	0xF8	-	UCD	URST	-	-	URW	UDS	UCS	0100 0X00
USBCON1	USB control 1	0xF9	-	-	-	-	UDMU	UDPU	UPE	USPD	0000 0001
USBDADR	USB DMA base address	0xFC									XXXX XXXX
USBDATA	USB access data	0xFA									XXXX XXXX
USBDD	USB DMA done	0xFF	-	-	-	-	USBDD[3:0]				0000 0000
USBDM	USB DMA interrupt mask	0xFD	-	-	-	-	USBDM[3:0]				0000 0000
USBDR	USB DMA request	0xFE	-	-	-	-	USBDR[3:0]				0000 0000
WDTCON	Watchdog control	0xA1	WDTTO	WDTPD	WDTPND	WDTEN	WDTPS[3]	WDTPS[2]	WDTPS[1]	WDTPS[0]	0000 0000
WKEDG	Port wakeup edge select	0xBD	-	-	-	-	WKEDG[3]	WKEDG[2]	WKEDG[1]	WKEDG[0]	0000 XXXX
WKEN	Port wakeup control	0xBC	-	-	-	-	WKEN[3]	WKEN[2]	WKEN[1]	WKEN[0]	0000 0000
WKPND	Port wakeup pending	0xBB	-	-	-	-	WKPND[3]	WKPND[2]	WKPND[1]	WKPND[0]	0000 XXXX

3.11.2 SFR table listed according to address

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'.

Address	Register Name	Description	Bit Functions								Reset
			MSB							LSB	
0x80	P0	Port 0									XXXX XXXX
0x81	SP	Stack pointer									0000 0111
0x82	DPL	Data pointer low byte									0000 0000
0x83	DPH	Data pointer high byte									0000 0000
0x84	DP1L	Data pointer 1 low byte									0000 0000
0x85	DP1H	Data pointer 1 high byte									0000 0000
0x86	DPCON	Data pointer control	DPID1	DPID0	DPTSE	DPAID	-	EMIEN	SPXSEL	DPS	0000 0000
0x87	PCON	Power control	SMOD	EA	HLT	POF	-	-	PD	IDL	0X01 0000
0x88	TCON	Timer 0/1 control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000
0x89	TMOD	Timer 0/1 mode	GATE1	C/T1	T1M[1]	T1M[0]	GATE0	C/T0	T0M[1]	T0M[0]	0000 0000
0x8A	TL0	Timer 0 low byte									0000 0000
0x8B	TL1	Timer 1 low byte									0000 0000
0x8C	TH0	Timer 0 high byte									0000 0000
0x8D	TH1	Timer 1 high byte									0000 0000
0x8E	EMICON1	EMI control 1	RW[3]	RW[2]	RW[1]	RW[0]	HD[1]	HD[0]	SET[1]	SET[0]	1111 1111
0x8F	EMICON2	EMI control 2	-	SZ[1]	SZ[0]	-	PSON	MUX	WRON	AL	0100 1111
0x90	P1	Port 1									XXXX XXXX
0x91	CKCON0	Clock control 0	-	-	LPOSC	PLLEN	-	SCPLL	SCS	SCKD	0000 0000
0x92	CKCON1	Clock control 1	FFCKE	32CKE	RTCKE	MCKE	PCKE	-	TCKE	UCKE	0001 1110
0x93	CKCON2	Clock control2	-	-	-	-	-	CKS	32CE	HSCE	0000 1101
0x94	CLKDIV	Clock divider									0000 0000
0x95	SPICON	SPI control	SPIPND	SM	RTS	WS	RXO	SMP	IDS	SPIEN	0000 0000
0x96	SPIBAUD	SPI baud control									0000 0000
0x97	SPIBUF	SPI data buffer									0000 0000
0x98	SCON	Serial port control	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000
0x99	SBUF	Serial port buffer									0000 0000
0x9B	SPH	Stack pointer high byte									0000 0000

0x9C	FIFOCON	FIFO interface control	FEN	ONLY	-	-	-	-	WR	RD	0000 0000
0x9D 0x9F	FIFOPTR FIFODATA	FIFO pointer FIFO data									0000 0000 0000 0000
0xA0	P2	Port 2									XXXX XXXX
0xA1	WDTCON	Watchdog control	WDTTO	WDTPD	WDTPND	WDTEN	WDTPS[3]	WDTPS[2]	WDTPS[1]	WDTPS[0]	0000 0000
0xA4	P0OD	Port 0 open-drain mode									0000 0000
0xA5	P1OD	Port 1 open-drain mode									0000 0000
0xA6	P2OD	Port 2 open-drain mode									0000 0000
0xA7	P3OD	Port 3 open-drain mode									0000 0000
0xA8	IEN0	Interrupt enable 0	EA	ERT	ET2	ES	ET1	EX1	ET0	EX0	0000 0000
0xA9	IEN1	Interrupt enable 1	-	-	EGPSI	EADC	EUD	EUI	ESI	EPW	0000 0000
0xAB	CCON	CPU control	PROT	-	LPM	ISRM	XMAP	-	-	-	0000 0000
0xAC	P0DIR	Port 0 direction control									1111 1111
0xAD	P1DIR	Port 1 direction control									1111 1111
0xAE	P2DIR	Port 2 direction control									1111 1111
0xAF	P3DIR	Port 3 direction control									1111 1111
0xB0	P3	Port 3									XXXX XXXX
0xB1	P0PUP	Port 0 pull-up control									0000 0000
0xB2	P1PUP	Port 1 pull-up control									0000 0000
0xB3	P2PUP	Port 2 pull-up control									0000 0000
0xB4	P3PUP	Port 3 pull-up control									0000 1100
0xB6	PSYN	Port synchronizer enable	-	-	-	-	P3SYN	P2SYN	P1SYN	P0SYN	0000 0000
0xB8	IP0	Interrupt priority 0	-	PRT	PT2	PS	PT1	PX1	PT0	PX0	0000 0000
0xB9	IP1	Interrupt priority 1	-	-	PGPSI	PADC	PUD	PUI	PSI	PPW	0000 0000
0xBB	WKPND	Port wakeup pending	-	-	-	-	WKPND[3]	WKPND[2]	WKPND[1]	WKPND[0]	0000 XXXX
0xBC	WKEN	Port wakeup control	-	-	-	-	WKEN[3]	WKEN[2]	WKEN[1]	WKEN[0]	0000 0000
0xBD	WKEDG	Port wakeup edge select	-	-	-	-	WKEDG[3]	WKEDG[2]	WKEDG[1]	WKEDG[0]	0000 XXXX
0xBF	FSET	Special function setting	-	T0DD	OCSEL[1]	OCSEL[0]	-	T1DD	EWDT	WDTRST	0100 1110
0xC0	GPSICON0	GPSI control 0	-	-	GPSIRXIE	GPSITXIE	TXCLKSEL	RXEDSEL	TXEDSEL	GPSIEN	0000 0000
0xC1	GPSICON1	GPSI control 1	RXBUF FULL	TXBUF EPTY	RXHDREN	TXHDREN	EOP	RXENOFF	TXMODE[1]	TXMODE[0]	0000 0000
0xC2	GPSICON2	GPSI control 2	HDRRXCLR	HDRTXCLR	GPSIEOPIE	GPSIRXHIE	GPSITXHIE	HEAD32	HDRRX MATCH	HDRTX LOAD	0000 0000
0xC3	GPSIHDR	GPSI header									0000 0000
0xC4	GPSIBUF	GPSI data buffer									0000 0000
0xC5	GPSIBAUD	GPSI baud rate divider									0000 0000
0xC8	T2CON	Timer 2 control	TF2	EXF2	-	-	EXEN2	TR2	C/T2	CP/RL2	0000 0000

0xC9	T2MOD	Timer 2 mode	-	PWM8	T2CLK[1]	T2CLK[0]	PWM16N	PWM16P	T2OE	DCEN	0000 0000
0xCA	RCAP2L	Timer 2 Capture/Reload/PWM duty cycle low byte									0000 0000
0xCB	RCAP2H	Timer 2 Capture/Reload/PWM duty cycle high byte									0000 0000
0xCC	TL2	Timer 2 counter low byte									0000 0000
0xCD	TH2	Timer 2 counter high byte									0000 0000
0xCE	TPR2L	Timer 2 PWM period low byte									0000 0000
0xCF	TPR2H	Timer 2 PWM period high byte									0000 0000
0xD0	PSW	Program status word	CY	AX	F0	RS1	RS0	OV	F1	P	0000 0000
0xD4	RTDIV	Real time clock divider control									0000 0000
0xD8	RTCON	Real time control	-	RTWK	-	-	RTWE	-	-	-	0000 0000
0xE0	ACC	Accumulator									0000 0000
0xE1	P0AIE	Port 0 analog input enable									0000 0000
0xE2	P1AIE	Port 1 analog input enable									0000 0000
0xE4	P3AIE	Port 3 analog input enable									0000 0000
0xEE	ADCB	ADC buffer									XXXX XXXX
0xF0	B	B register									0000 0000
0xF1	ADCT	ADC threshold									XXXX XXXX
0xF3	ADCSC	ADC sampling clock control									0000 0000
0xF4	ADCCON0	ADC control 0	ADCPND	INBUF	REFS	TM	TC[1]	TC[0]	GO	ADCEN	0000 0000
0xF5	ADCCON1	ADC control 1	CM	GT	-	REFBE	-	AS[2]	AS[1]	AS[0]	0000 0000
0xF8	USBCON0	USB control 0	-	UCD	URST	-	-	URW	UDS	UCS	0100 0X00
0xF9	USBCON1	USB control 1	-	-	-	-	UDMU	UDPU	UPE	USPD	0000 0001
0xFA	USBDATA	USB access data									XXXX XXXX
0xFB	USBADR	USB access address	-	-	USBADR[5:0]						00XX XXXX
0xFC	USBDADR	USB DMA base address									XXXX XXXX
0xFD	USBDM	USB DMA interrupt mask	-	-	-	-	USBDM[3:0]				0000 0000
0xFE	USBDR	USB DMA request	-	-	-	-	USBDR[3:0]				0000 0000
0xFF	USBDD	USB DMA done	-	-	-	-	USBDD[3:0]				0000 0000

4 External Memory Interface (EMI)

External memory interface of CFP5102 is compatible to the external bus of classic 8051/8052, supporting 16-bit addressing space, multiplexed address control, separated program and data memory selection. CFP5102 makes improvement to further support different memories and different connection topologies.

CFP5102 supports the following mode to fit in different IO budgets and to connect to different memories.

1. multiplexed address mode (original in 8051/8052)
2. non-multiplexed address mode (high performance)
3. 8-bit addressing mode (for small sketchpad memory and lookup tables)
4. FIFO mode (push to and pop from external FIFO memory)

In addition, the setup time, the hold time and the duration of the read and write operation can be fine tuned to suit for memories with different speed grading.

Register EMICON1 and EMICON2 are used to configure the EMI. Their contents are listed below and self-explained. In the following sections, the topology of the operation mode, corresponding IO used and the timing of read and write operations are shown.

EMICON1: **EMI Control Register 1**
TYPE: **R/W**
ADDRESS: **0x8E**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**RW[3]**	**RW[2]**	**RW[1]**	**RW[0]**	**HD[1]**	**HD[0]**	**SET[1]**	**SET[0]**
Default	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Name	Mode	Description	Setting
RW	RW	Read or write signal width configuration bit (T_{rdwr})	0000: 1 cycle 0001: 2 cycles 0010: 3 cycles 0011: 4 cycles 0100: 5 cycles 0101: 6 cycles 0110: 7 cycles 0111: 8 cycles 1000: 9 cycles 1001: 10 cycles 1010: 11 cycles 1011: 12 cycles 1100: 13 cycles 1101: 14 cycles 1110: 15 cycles 1111: 16 cycles
HD	RW	Hold time configuration bit (T_{hold})	00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
SET	RW	Setup time configuration bit (T_{setup})	00: 0 cycle 01: 1 cycle 10: 2 cycles

			11: 3 cycles

EMICON2: **EMI Control Register 2**
TYPE: **R/W**
ADDRESS: **0x8F**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**-**	**SZ[1]**	**SZ[0]**	**-**	**PSON**	**MUX**	**WRON**	**AL**
Default	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**1**

Name	Mode	Description	Setting
SZ	RW	EMI address space select	00: No address (FIFO mode) 01: 8-bit address other: 16-bit address
PSON	RW	Program space select enable bit This option is available only in multiplexed address bus mode.	0: disable 1: enable
MUX	RW	Address data multiplex enable bit	0: disable 1: enable
WRON	RW	Write signal enable bit	0: disable 1: enable
AL	RW	Address latch enable signal width configuration bit (T_{ALE})	0: 1 cycle 1: 2 cycles

4.1 Multiplexed Address Mode

Table 4-1: Required IO resources

EMI	Port
A[15:8]	P2
A[7:0]/D[7:0]	P0
ALE	P1.1
PSEN	P1.0
/WR	P3.6
/RD	P3.7

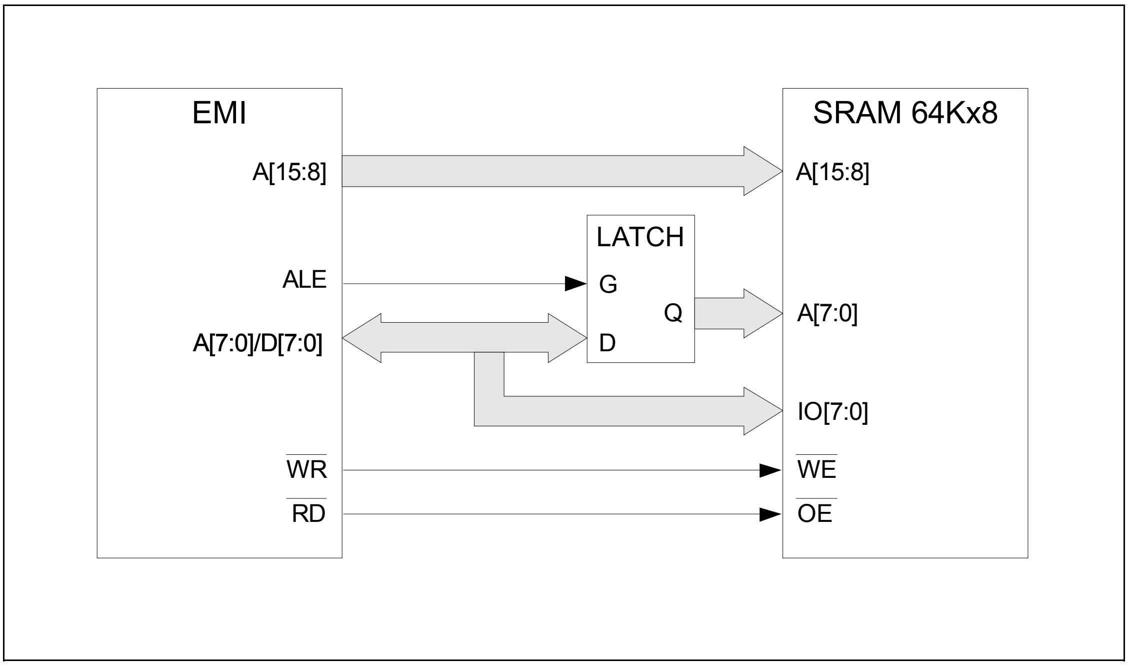


Figure 4-1: Connection topology

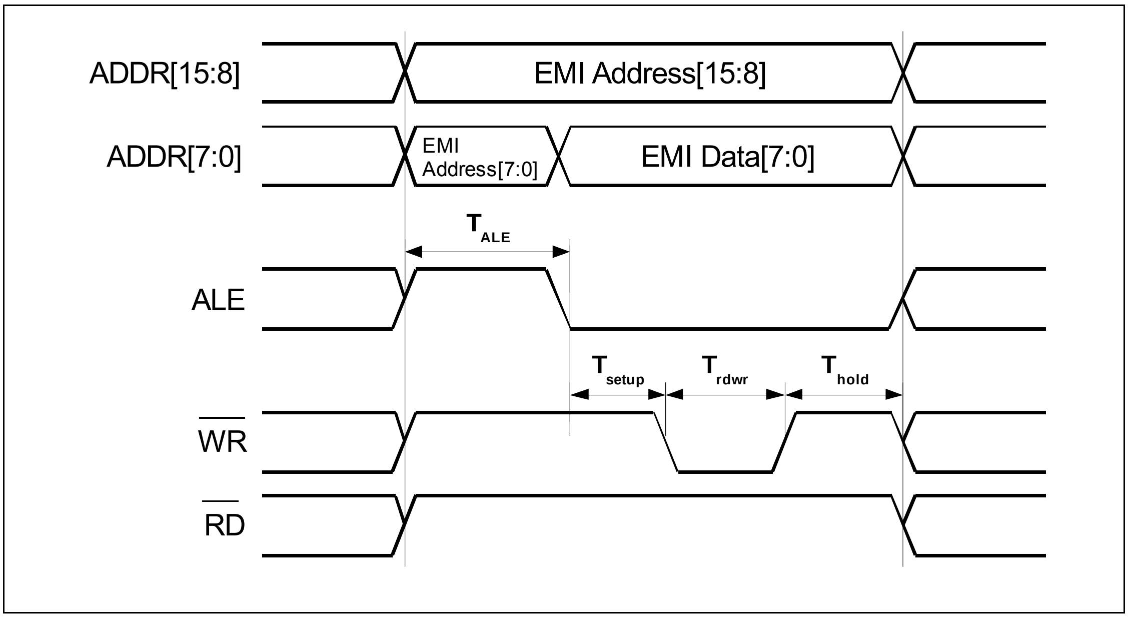


Figure 4-2: Timing of write operation

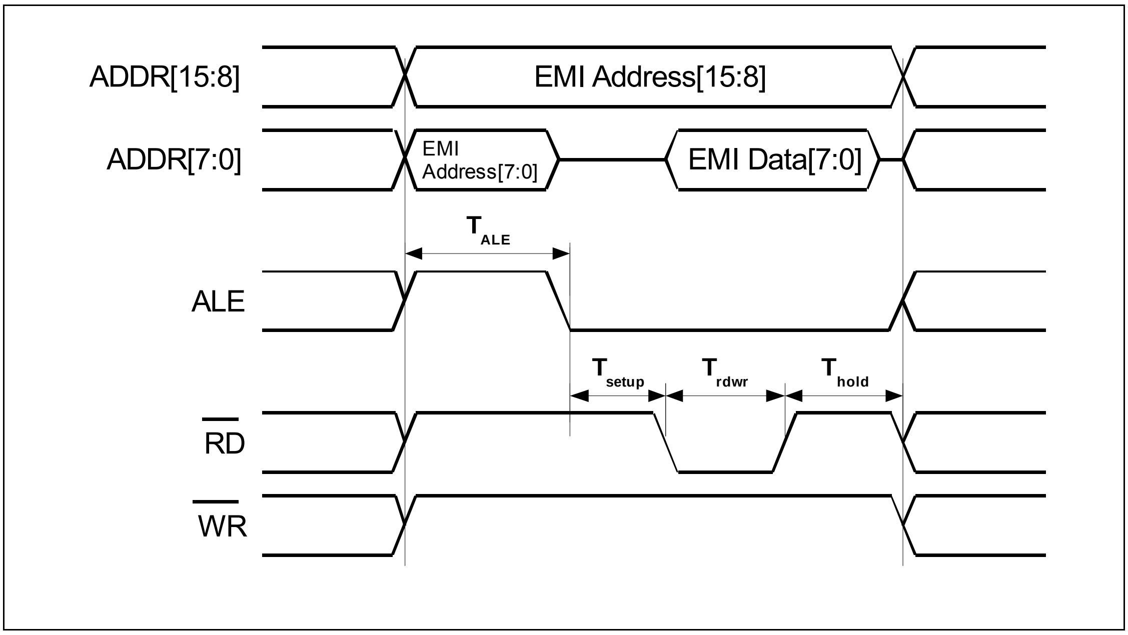


Figure 4-3: Timing of read operation when PSON = 0

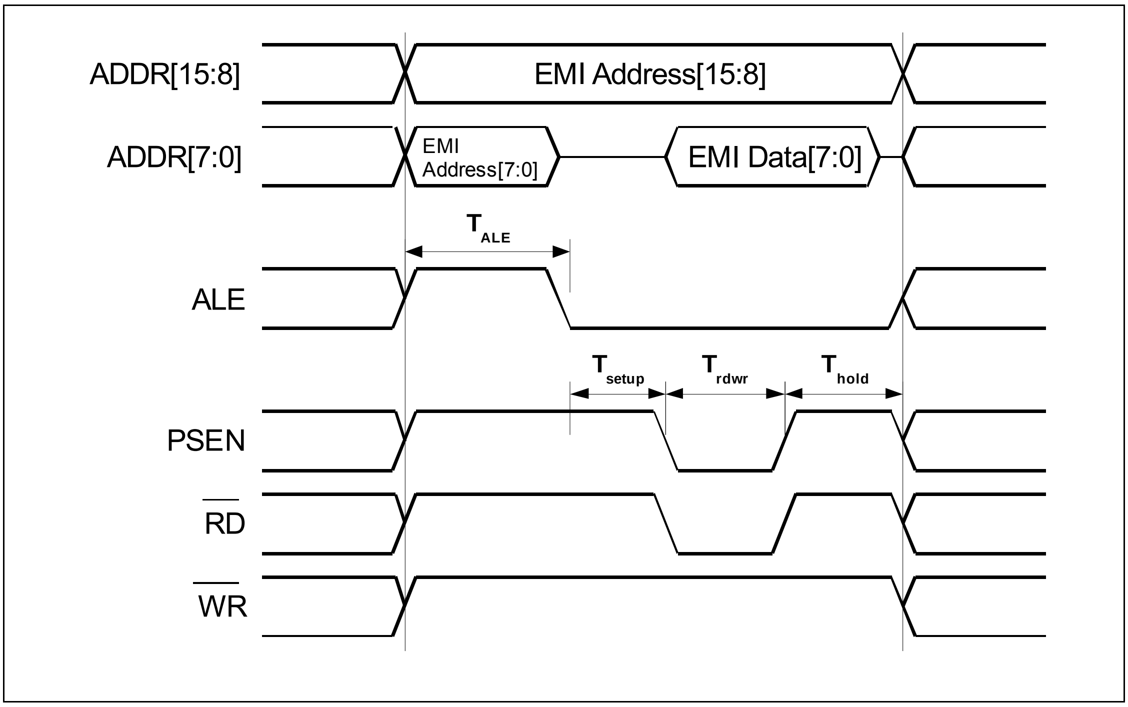


Figure 4-4: Timing of read operation when PSON = 1

4.2 Non-Multiplexed Address Mode

Table 4-2: Required IO resources

EMI	Port
A[15:8]	P2
A[7:0]	P1
D[7:0]	P0

/WR	P3.6
/RD	P3.7

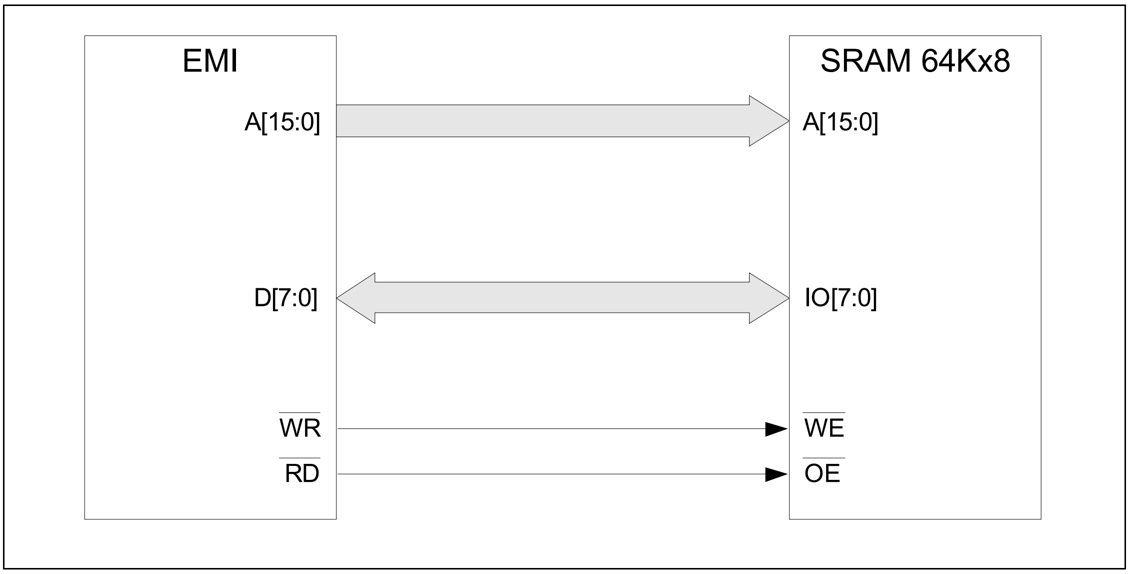


Figure 4-5: Connection topology

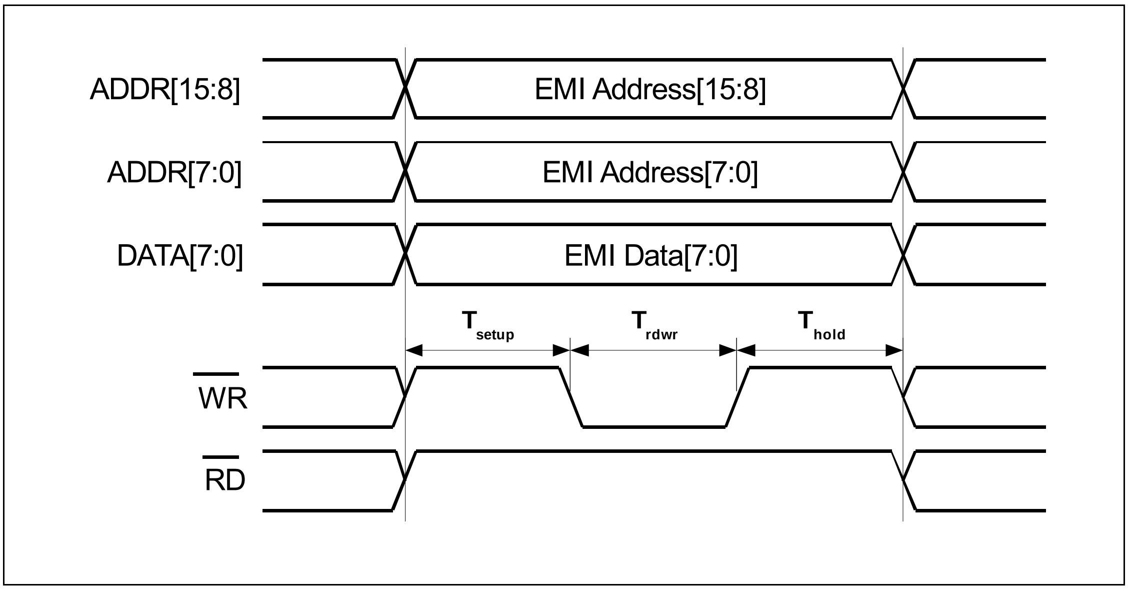


Figure 4-6: Timing of write operation

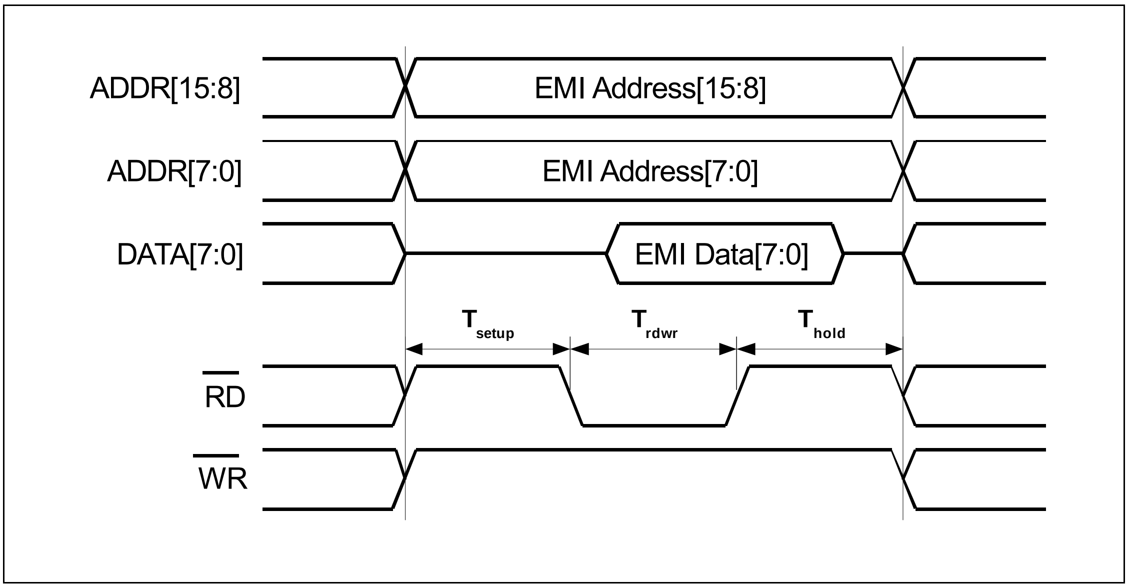


Figure 4-7: Timing of read operation

4.3 8-bit Address Mode

Table 4-3: Required IO resources

EMI	Port
A[7:0]	P1
D[7:0]	P0
/WR	P3.6
/RD	P3.7

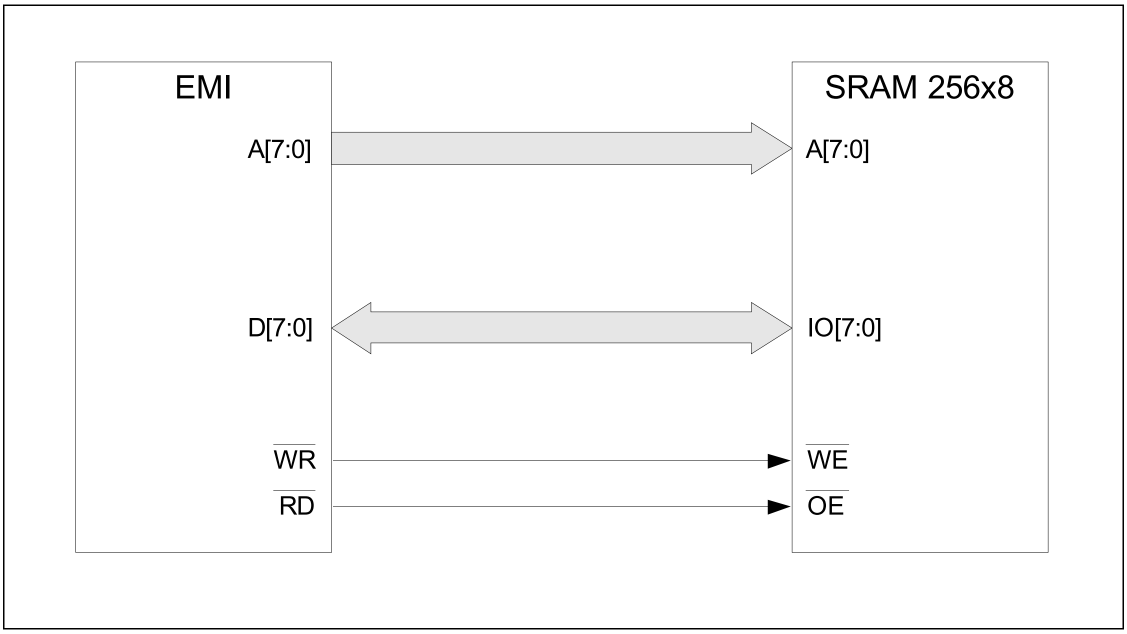


Figure 4-8: Connection topology

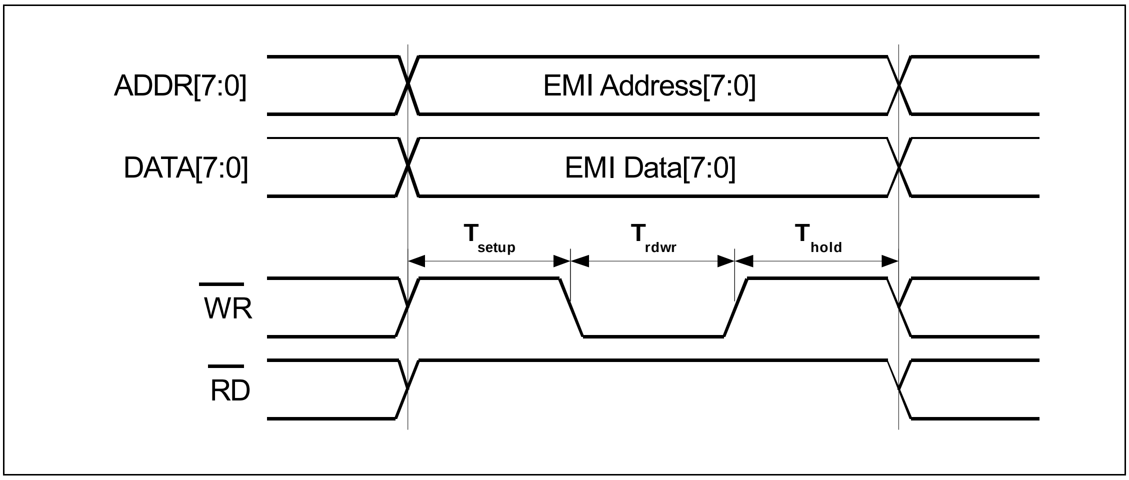


Figure 4-9: Timing of write operation

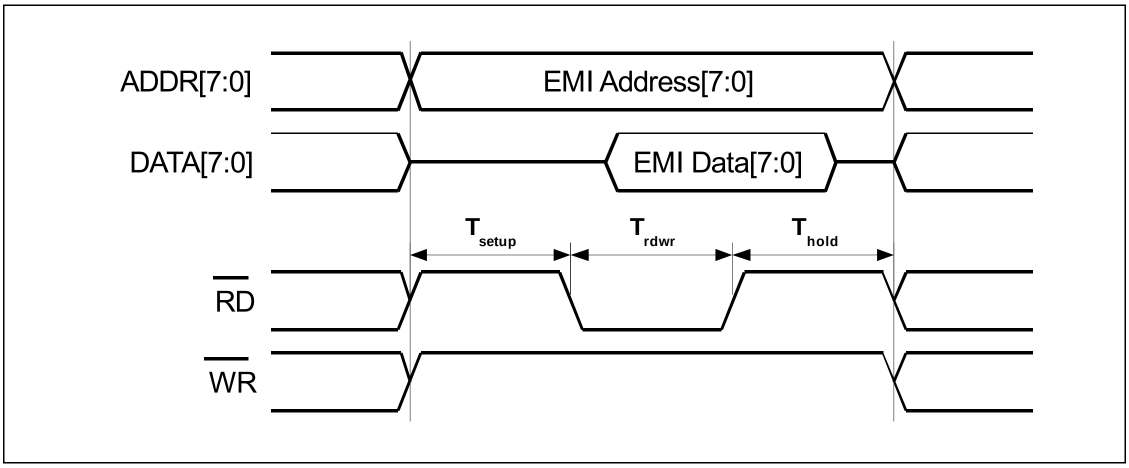


Figure 4-10: Timing of read operation

4.4 FIFO Mode

Table 4-4: Required IO resources

EMI	Port
D[7:0]	P0
/WR	P3.6
/RD	P3.7

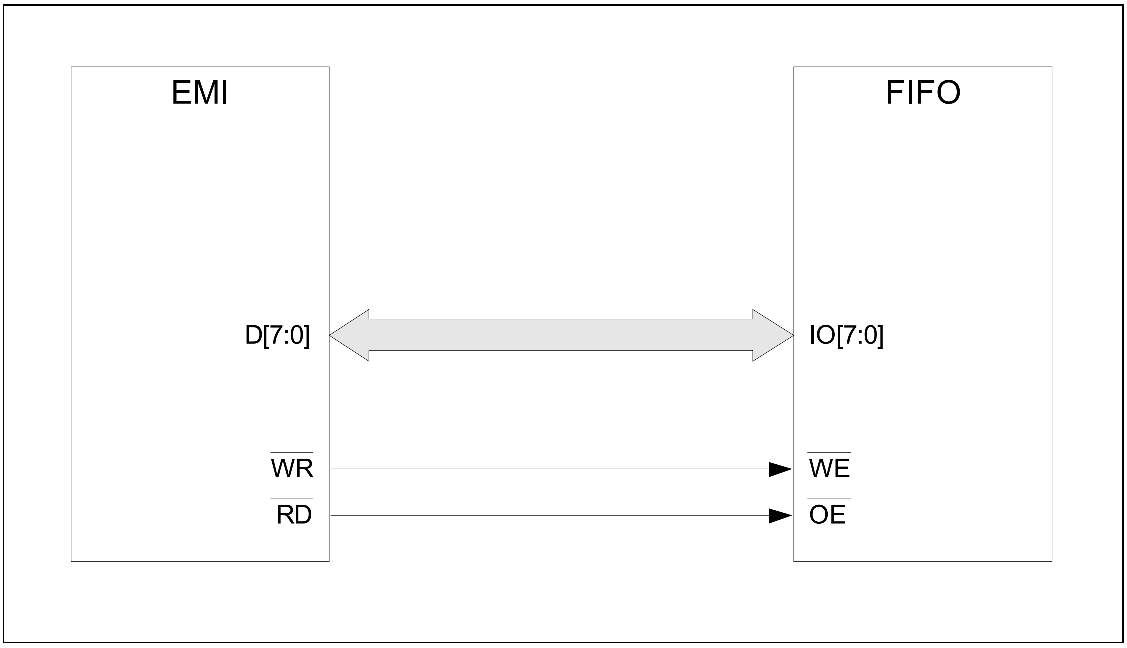


Figure 4-11: Connection topology

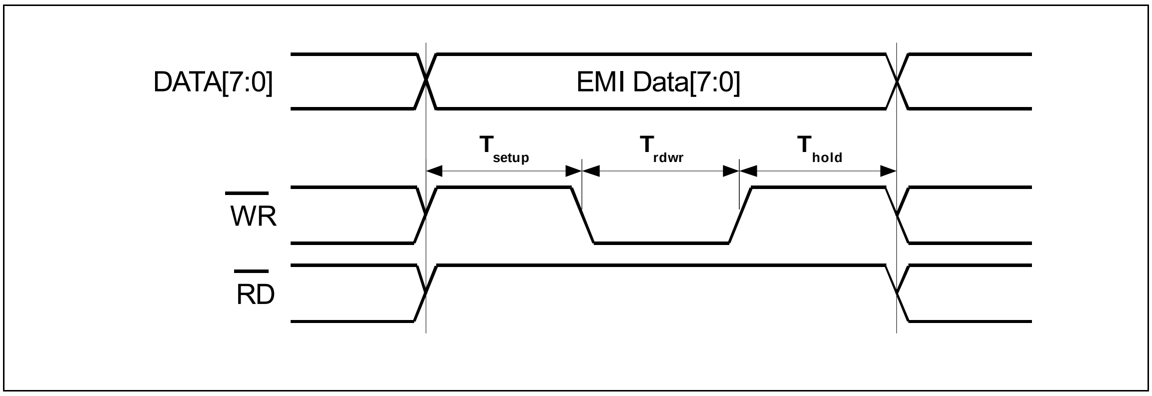


Figure 4-12: Timing of write operation

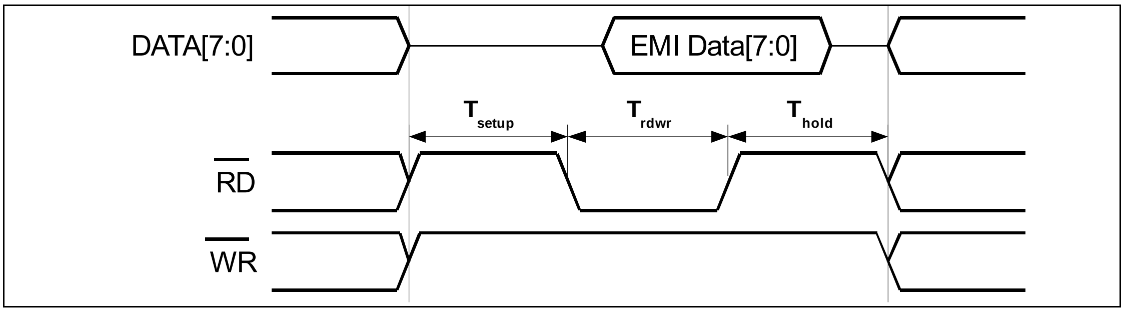


Figure 4-13: Timing of read operation

5 Interrupt Processing

CFP5102 extends the interrupt system to support totally 14 interrupt sources 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 interrupts are 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 (IEN0.7) 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.

IEN0: **Interrupt Enable Register 0**
TYPE: **R/W**
ADDRESS: **0xA8**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**EA**	**ERT**	**ET2**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**
Default	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Name	Mode	Description	Setting
EA	RW	Global interrupt enable bit	0: disable 1: enable
ERT	RW	Real-time wakeup interrupt enable bit	0: disable 1: enable
ET2	RW	Timer 2 interrupt enable bit	0: disable 1: enable
ES	RW	Serial port interrupt enable bit	0: disable 1: enable
EX1	RW	External interrupt 1 enable bit	0: disable 1: enable
ET1	RW	Timer 1 interrupt enable bit	0: disable 1: enable
EX0	RW	External interrupt 0 enable bit	0: disable 1: enable
ET0	RW	Timer 0 interrupt enable bit	0: disable

			1: enable

IEN1: **Interrupt Enable Register 1**
TYPE: **R / W**
ADDRESS: **0xA9**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	EGPSI	EADC	EUD	EUI	ESI	EPW
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
EGPSI	RW	GPSI interrupt enable bit	0: disable 1: enable
EADC	RW	ADC interrupt enable bit	0: disable 1: enable
EUD	RW	USB DMA interrupt enable bit	0: disable 1: enable
EUI	RW	USB interface interrupt enable bit	0: disable 1: enable
ESI	RW	Serial protocol interface interrupt enable bit	0: disable 1: enable
EPW	RW	Port wakeup interrupt enable bit	0: disable 1: enable

5.1 Interrupt Sources and Vectors

The CPU supports 14 interrupt sources. 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. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behaviour 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 (IP0 or IP1) 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.

Caution:
(1) Watchdog interrupt is in ‘SUPER’ which has higher priority than any interrupt source having priority bit = ‘1’

IP0: Interrupt Priority Register 0
TYPE: R / W
ADDRESS: 0xB8

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	PRT	PT2	PS	PT1	PX1	PT0	PX0
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
PRT	RW	Real-time wakeup interrupt priority select	0: disable 1: enable
PT2	RW	Timer 2 interrupt priority select	0: disable 1: enable
PS	RW	Serial port interrupt priority select	0: disable 1: enable
PX1	RW	External interrupt 1 priority select	0: disable 1: enable
PT1	RW	Timer 1 interrupt priority select	0: disable 1: enable
PX0	RW	External interrupt 0 priority select	0: disable 1: enable
PT0	RW	Timer 0 interrupt priority select	0: disable 1: enable

IP1: Interrupt Priority Register 1
TYPE: R / W
ADDRESS: 0xB9

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	PGPSI	PADC	PUD	PUI	PSI	PPW
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
PGPSI	RW	GPSI interrupt priority select	0: disable 1: enable
PADC	RW	ADC interrupt priority select	0: disable 1: enable
PUD	RW	USB DMA interrupt priority select	0: disable 1: enable
PUI	RW	USB interface interrupt priority select	0: disable 1: enable
PSI	RW	Serial protocol interface interrupt priority select	0: disable 1: enable
PPW	RW	Port wakeup interrupt priority select	0: disable 1: enable

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.

Table 5-1: Summary of interrupts

Interrupt Source	Interrupt Vector	Intrinsic Priority	Pending Flag	Bit-addressable	Cleared Automatically	Enable Bit	Priority Select
External Interrupt 0	0x03	1 (lowest)	TCON.1	Y	Y	IEN0.0	IP0.0
Timer 0 Overflow	0x0B	2	TCON.5	Y	Y	IEN0.1	IP0.1
External Interrupt 1	0x13	3	TCON.3	Y	Y	IEN0.2	IP0.2
Timer 1 Overflow	0x1B	4	TCON.7	Y	Y	IEN0.3	IP0.3
Serial Port	0x23	5	SCON.0 SCON.1	Y	N	IEN0.4	IP0.4
Timer 2 Overflow	0x2B	6	T2CON.6 T2CON.7	Y	N	IEN0.5	IP0.5
Real-timer Wakeup	0x33	7	RTCON.6	Y	N	IEN0.6	IP0.6
Port Wakeup	0x3B	8	WKPND.0 WKPND.1 WKPND.2 WKPND.3	N	N	IEN1.0	IP1.0
SPI	0x43	9	SPICON.7	N	N	IEN1.1	IP1.1
USB Interface	0x4B	10	USBCON0.6	Y	N	IEN1.2	IP1.2
USB DMA	0x53	11	USBDR USBDM	N	N	IEN1.3	IP1.3
ADC	0x5B	12	ADCCON0.7	N	N	IEN1.4	IP1.4
GPSI	0x63	13	GPSICON1.3 GPSICON1.6 GPSICON1.7 GPSICON2.0 GPSICON2.1	N	N	IEN1.5	IP1.5
WDT	0x6B	Super	WDTCON.5	N	N	FSET.1	-
Reset	0x00	**Top (highest)**	-	-	-	-	-

6 Clocks and Resets and Power Management

6.1 Clock System

CFP5102 possesses a configurable clock system aiming to provide balance between performance and power consumption in different applications. It provides programmable clock divider to allow dynamic selecting clock frequency depending on the workload to eliminate unnecessary power consumption in clock. It also provides fine-grain clock gating to shutdown unused part completely. The organization of clock system is illustrated in Figure 6-1.

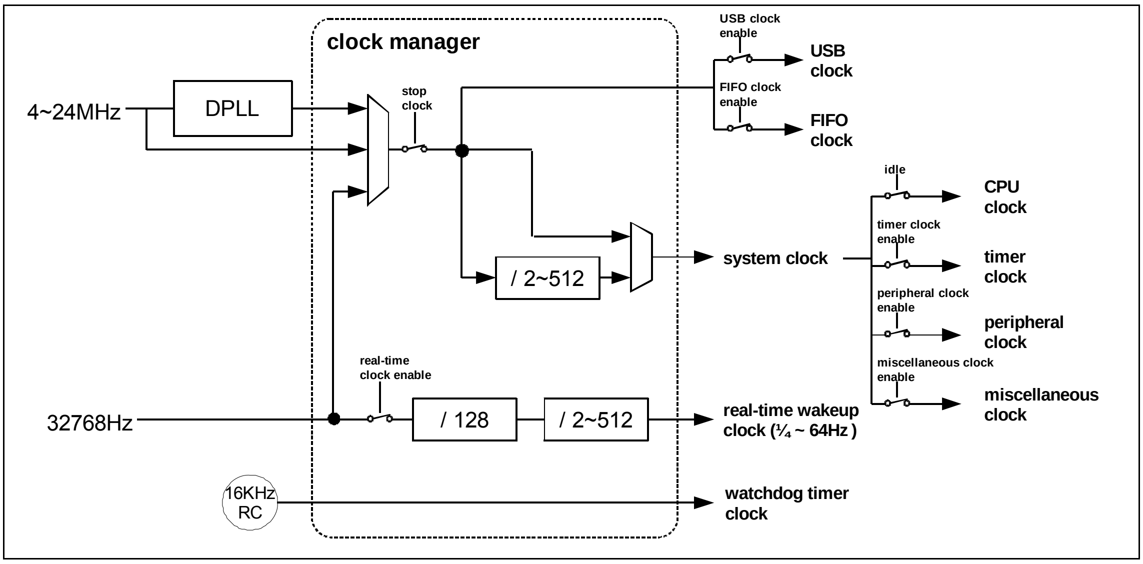


Figure 6-1: Block diagram of clock system

6.1.1 Clock Control

The major clock source is a high speed crystal oscillator that suits for 4MHz to 24MHz crystals. This clock is used directly 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. Other than that, a digital PLL dedicated for USB module is provided. It doubles 24MHz clock to 48MHz which is necessary to operate USB.

The selection of clock sources is controlled by setting register CKCON0. 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. (Please refer to Section 6.1.3)

To activate system clock divider, SCKD the bit 0 of CKCON0 has to be set. Similarly, the activation takes 2 periods of the divided clock, and system clock suspends, meanwhile. With regard of this, it is recommended to activate system clock divider before choosing high divide ratio. The clock divide ratio is defined by system clock divider control register CKDIV. The frequency of the system clock is

$F_{system\_clock} = F_{clock\_in}/2*(n+1)$, where n is the value of CKDIV

CKCON0: **Clock Control Register 0**
TYPE: **R / W**
ADDRESS: **0x91**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**-**	**-**	**LPOSC**	**PLLEN**	**-**	**SCPLL**	**SCS**	**SCKD**
Default	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Name	Mode	Description	Setting
LPOSC	RW	32,768 Hz crystal low power mode enable bit	0: disable 1: enable
PLLEN	RW	DPLL enable bit	0: disable 1: enable
SCPLL	RW	Select DPLL output as high-speed clock source	0: disable 1: enable
SCS	RW	System clock select	0: select high-speed clock 1: select 32,768 Hz clock
SCKD	RW	System clock divider enable	0: disable 1: enable

CKDIV: **System Clock Divider Control Register**
TYPE: **W**
ADDRESS: **0x94**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**CKDIV[7:0]**							
Default	**0x00**							

6.1.2 Clock Gating

CFP5102 provides comprehensive clock gating options for eliminating power-wasting activities. As shown in Figure 6-1, the clock system is basically organized into 6 clocks with individual gating. The 6 clocks are CPU clock, timer clock, peripheral clock, USB clock, FIFO clock and miscellaneous clock. Real-time wakeup clock is used for event triggering only, will be discussed in the following section.

The influence of each clock is summarised in Table 6-1.

Table 6-1: Scope of clocks

Clock	**Influence**
CPU clock	1. CPU 2. USB DMA
Timer clock	1. timer 0 2. timer 1 3. timer 2
Peripheral clock	1. serial port 2. SPI 3. GPSI 4. ADC
Miscellaneous clock	1. port synchronizers 2. serial port data input 3. SPI data input 4. GPSI data input 5. USB interface unit 6. timer 0 / 1 / 2 clock input 7. timer 0 / 1 / 2 external pin edge detection
Real-time clock	1. real-time wakeup

USB clock	1. USB controller
FIFO clock	1. USB FIFO 2. FIFO interface
Watchdog timer clock	1. watchdog timer

Some modules require more than 1 clock to operate as described below:

1. For timer 0, timer 1 and timer 2, timer clock is needed for counting. Miscellaneous clock is needed for capturing external clock and detecting external pin edge.
2. For serial port, SPI and GPSI, both peripheral clock and miscellaneous clock are needed.
3. For ADC, only peripheral clock is needed.
4. For USB, USB clock, FIFO clock and miscellaneous clock are needed. To operate USB DMA, CPU clock is also needed.
5. All port synchronizers are clocked by miscellaneous clock.

The clock gating is controlled by register CKCON1. The setting takes effect immediately at the coming rising edge of the corresponding clocks. CPU clock is controlled by register PCON. It can be stopped in idle mode only.

CKCON1: **Clock Control Register 1**
TYPE: **R/W**
ADDRESS: **0x92**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	FFCKE	32CKE	RTCKE	MCKE	PCKE	-	TCKE	UCKE
Default	0	0	0	1	1	1	1	0

Name	Mode	Description	Setting
FFCKE	RW	FIFO clock enable	0: disable 1: enable
32CKE	RW	32,768 Hz clock enable Enable only when the 32,768 Hz crystal is stable. Disable before turn off the 32,768 Hz crystal	0: disable 1: enable
RTCKE	RW	Real-time clock enable	0: disable 1: enable
MCKE	RW	Miscellaneous clock enable	0: disable 1: enable
PCKE	RW	Peripheral clock enable	0: disable 1: enable
TCKE	RW	Timer clock enable	0: disable 1: enable
UCKE	RW	USB clock enable	0: disable 1: enable

6.1.3 Crystal Oscillators

Register CKCON2 is used to control the oscillator. It has different meaning for reading from and writing to this register. Reading CKCON2 reports the current status of the oscillators. There are 3 active bits: CKS reports the condition of the resuming oscillator. When the stopped oscillator is re-activated, this flag is cleared. It is set again when the oscillator is stable and ready to use. Stopping an oscillator take no effect to this flag. 32CE shows the status of the 32,768 Hz oscillator, it is set when this oscillator is in active state and is cleared in stop state. As 32,768 Hz oscillator is rather slow comparing to high speed oscillator, there is a certain latency to actual state switching from a stop or resume command. HSCE, similar to 32CE, shows the status of the high speed oscillator.

Writing keywords to CKCON2 issues command to stop or resume the oscillators. Special keywords are assigned to each operation as tabulated in Table 6-2. To avoid accidental operation to the oscillators, keywords must be entered in bit-bang way (first enter the keyword and then enter the bit inverted keyword). For example, write 0xAA to CKCON2 and then write 0x55 immediately to stop the high speed oscillator.

CKCON2: **Clock Control Register 2**
TYPE: **R / W**
ADDRESS: **0x93**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	-	CKS	32CE	HSCE
Default	0	0	0	0	1	1	0	1

Name	Mode	Description	Setting
CKS	R	Clock stability flag. When a stopped clock is re-activated, this flag is cleared until the resumed clock becomes stable	0: unstable 1: stable
32CE	R	32,768 Hz oscillator status flag	0: disable 1: enable
HSCE	R	High speed oscillator status flag	0: disable 1: enable

Table 6-2: Keywords for operating oscillators

	Stop	Resume
High-speed oscillator	0xAA	0x3C
32,768 Hz oscillator	0xF0	0x66

There is another guard bit for restricting the stop oscillator operation. PD, the power down mode enable bit, is used to protect the system clock from unintended suspension. When this bit is cleared, it is unable to stop the oscillator that is using as the clock source of the system. This bit takes no effect to the oscillator that is not selected for system clock source.

PCON: **Processor Control Register**
TYPE: **R / W**
ADDRESS: **0x87**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic							PD	
Default							0	

Name	Mode	Description	Setting
PD	RW	Power down mode enable bit When this bit is cleared, it is unable to stop the oscillator that is using as the clock source of the system.	0: disable 1: enable

To summarised, the procedures to operate the oscillators are described below:

Turn on 32,768 Hz oscillator when using high speed oscillator for system clock source

1. Configure LPOSC in CKCON0 to enable low power mode if necessary (please refer to Section 6.3.6)
2. Write 0x66 to CKCON2
3. Write 0x99 to CKCON2
4. Poll CKS until it is set
5. Write '1' to 32CKE of CKCON1
6. When 32CE is set, 32,768 Hz oscillator is ready to use

Turn off 32,768 Hz oscillator when using high speed oscillator for system clock source

1. Write '0' to 32CKE of CKCON1
2. Write 0xF0 to CKCON2
3. Write 0x0F to CKCON2
4. 32,768 Hz oscillator is stopped when 32CE is cleared

Turn on high speed oscillator when using 32,768 Hz oscillator for system clock source

1. Write 0x3C to CKCON2
2. Write 0xC3 to CKCON2
3. Poll CKS until it is set
4. When HSCE is set, high speed oscillator is ready to use

Turn off high speed oscillator when using 32,768 Hz oscillator for system clock source

1. Write 0xAA to CKCON2
2. Write 0x55 to CKCON2
3. High speed oscillator is stopped when HSCE is cleared

CKCON1: **Clock Control Register 1**
TYPE: **R/W**
ADDRESS: **0x92**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic		32CKE						
Default		0						

Name	Mode	Description	Setting
32CKE	RW	32,768 Hz clock enable Enable only when the 32,768 Hz crystal is stable. Disable before turn off the 32,768 Hz crystal	0: disable 1: enable

Oscillator configurations

Table 6-3 illustrates recommended configuration of crystal/resonator oscillator at different operating frequency. R_M is the motional resistance of the crystal/resonator and can be found in crystal/resonator vendor's datasheet.C1 and C2 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 C1 and C2. The tuning range is highly dependent on crystal/resonator and users need to consult the crystal/resonator vendor for details.

Table 6-3: Use of R_M and C_1, C_2

Crystal	Maximum R_M (ohm)	Loading capacitor C_1, C_2 (pF)
32KHz	50K	30
1MHz – 13MHz	100	10-30
13MHz – 19MHz	40	10-30
20MHz – 24MHz	40	10-30

PCB layout recommendation

Precaution should be taken when drawing printed-circuit board (PCB) layout for crystal. The crystal/resonator and the loading capacitor C_1//C_2 should be placed closest to CFP5102 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.

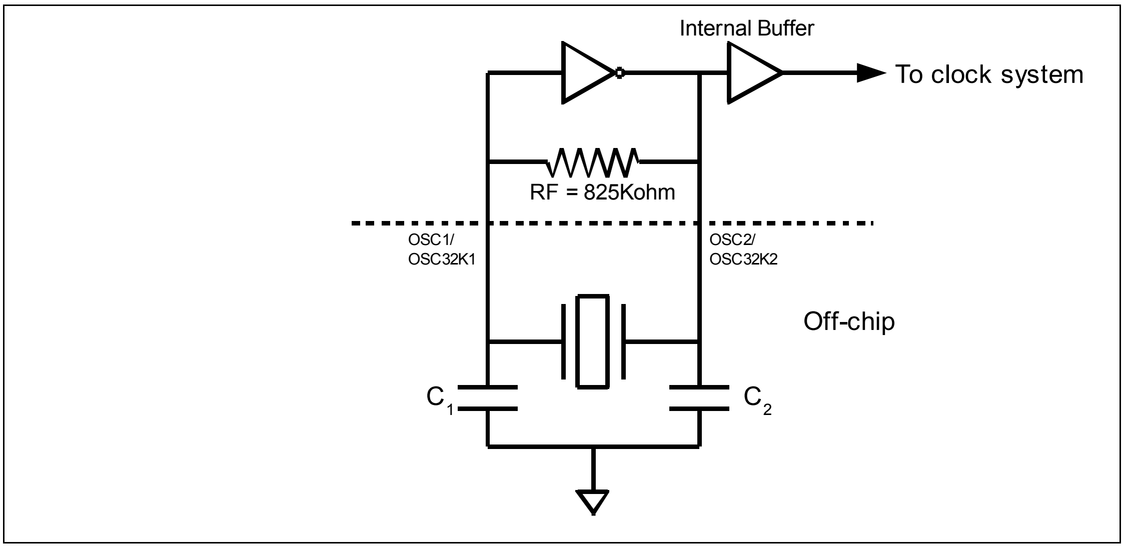


Figure 6-2: Crystal oscillator connection diagram

6.1.4 Using External Clock Generators

User has an opportunity to use external clock generator to completely bypass CFP5102 oscillator circuit. In order to do this, user can safely connect the clock source output to the OSC2 pin while floating OSC1 pin. Figure 6-3 illustrates the connection details.

When using external clock sources, the system clock has the same phase with the input clock. Figure 6-4 illustrates the phase relationship between input clock and the system clock.

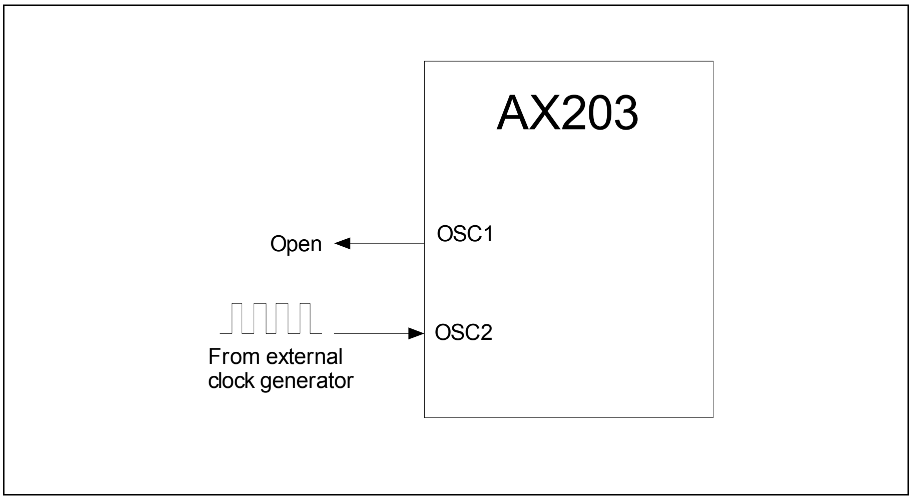


Figure 6-3: Connection diagram when using external clock generator

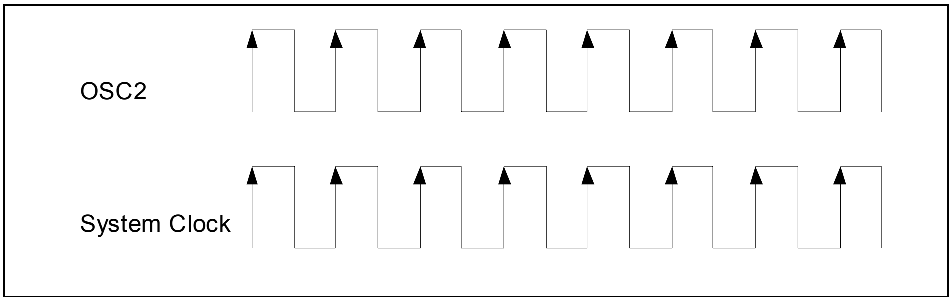


Figure 6-4: The phase relationship between input clock and the system clock

6.1.5 RC Oscillator

16KHz 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.

6.1.6 Digital Phase Lock Loop

To generate 48MHz clock for USB module, a digital phase lock loop (DPLL) is provided. This DPLL is designed for generating 48MHz clock from 24MHz input clock only. On the other hand, it does not require stability time. It can be used immediately after enable. To enable DPLL, user has to write ‘1’ to PLLEN in CKCON0.

When using USB, the system clock must be 48MHz; however, the recommend clock frequency for CPU is 24MHz. User is recommended to enable clock divider before switch to DPLL.

CKCON0: **Clock Control Register 0**
TYPE: **R/W**
ADDRESS: **0x91**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic				PLLEN				
Default				0				

Name	Mode	Description	Setting
PLLEN	RW	DPLL enable bit	0: disable 1: enable

6.1.7 Real-Time Wakeup Clock

Real-time wakeup clock is a special clock. It does not clock modules as other clocks. The only purpose of this clock is to trigger real-time wakeup module for implementing a software real-time counter. The period of the real-time wakeup clock can be tuned from 1/64 second to 4 seconds using register RTDIV.

RTDIV: **Real-time Clock Divider Control Register**
TYPE: **W**
ADDRESS: **0xD4**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	RTDIV[7:0]							
Default	0x00							

Description	Setting
Real-time clock is controlled by RTDIV and divided by $2\times(n+1)$, where n is the value of RTDIV.	

6.1.8 Supply Clock to Off-Chip Peripherals

For single clock on-board system, CFP5102 can share its clock with off-chip peripherals by outputting the clock through pin P1.0. The output clock is controlled by OCSEL, bit 4 and bit 5 of register FSET. In addition, the pin P1.0 has to be configured as output. As the driving of the IO is not designed for outputting clock, supplying clocks to couples of off-chip peripherals may deteriorate the quality of the output clock.

FSET: **Special Function Setting Register**
TYPE: **R / W**
ADDRESS: **0xBF**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic			OCSEL[1]	OCSEL[0]				
Default			0	0				

Name	Mode	Description	Setting
OCSEL	RW	Output clock selection	00: disable 01: USB clock 10: high speed oscillator 11: high speed oscillator ÷ 2

6.2 Resets

CFP5102 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.

6.2.1 Reset Sequence

When the reset happens, CFP5102 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

CFP5102 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

CFP5102 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 DVDD=3.3V, the POR threshold voltage is set to be about 2.2V.

Sometimes, when the VDD is power-off and quickly power-on again, there might be cases that the POR will work improperly and internal reset might not be generated. For this reason, CFP5102 POR circuit incorporates an internal self-reset module to discharge PORB output during power-off to ensure each power cycle will work properly.

However, it is also highly recommended user should have a long time between power-off and next power-on to ensure proper start-up. The time depends on actual system board environment and how much decoupling

capacitors between power and ground. User has to take into account this effect during board level design.

Figure 6-5 illustrates the power-on and reset signals waveform during proper power-on and power-off. Internally, there is T_{POR} and T_{RC} time for both the POR circuit and the internal counter. T_{POR} is the time for the POR circuit to stay at zero voltage until it reaches V_{POR} and the time varies for different VDD rise-up time. It can be assumed to be about 2/3 of the VDD rise-up time. T_{RC} is the time for internal counter to count at least 16ms using internal RC-oscillator when the counter sees a high logic from PORB signal. As a result, the overall internal reset time is the sum of T_{POR} and T_{POR}. Such a long time is required to ensure the crystal oscillator has been started up properly to provide a clean and stable clock source for system use and Phase-Locked Loop (PLL). It also ensures all internal logics are properly reset.

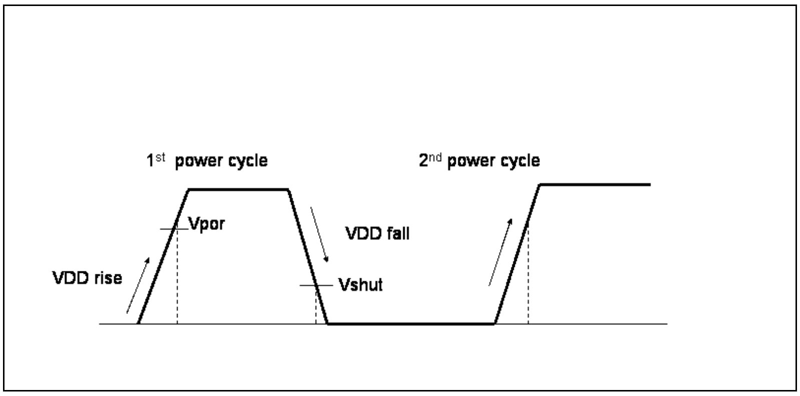


Figure 6-5: Waveform of power on reset

PCON: **Processor Control Register**
TYPE: **R/W**
ADDRESS: **0x87**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic				**POF**				
Default				**1**				

Name	Mode	Description	Setting
POF	RW	Power on reset flag This bit is used to indicate the last reset source is power on reset. To do that, user clear this bit to '0' in the program, if CFP5102 has undergone power on reset, this bit will be set after reset sequence. Note that this bit cannot be altered by other resets	0: inactive 1: reset by POR

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 7.11.

6.3 Power Management

6.3.1 Operating in 3.3V or 5V Systems

CFP5102 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 3.3V systems, the on-chip regulator should be turned off by tying LDOE to VDDIO. 3.3V power should be connected to both DVDD and VDDIO.

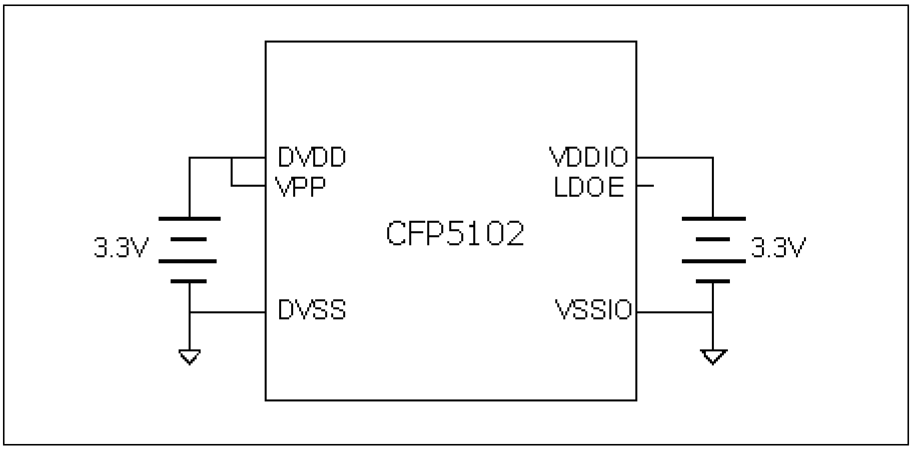


Figure 6-6: Topology of supplying 3.3V to CFP5102

Operating in 5V System

In 5V systems, the on-chip regulator should be turned on by tying LDOE to VSSIO. 5V power should be connected to VDDIO only. A compensation network should be attached to DVDD and DVSS to ensure good quality of on-chip regulator output. For details, please refer to Section 6.3.2.

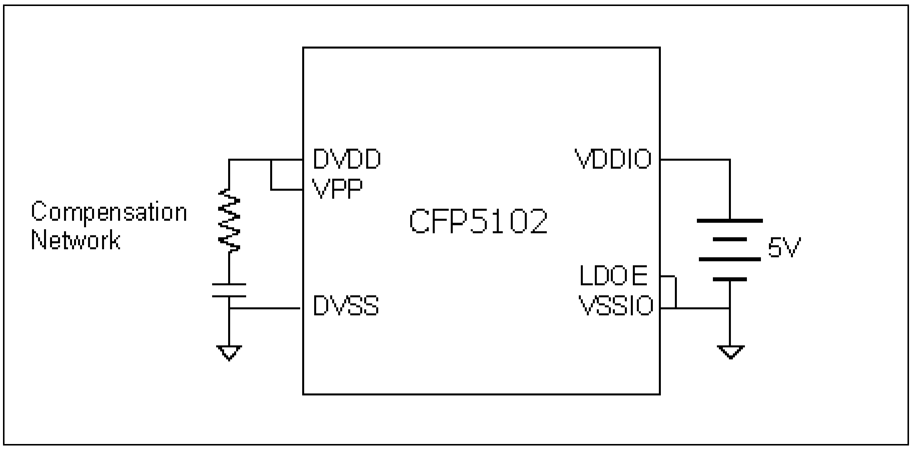


Figure 6-7: Topology of supplying 3.3V to CFP5102

6.3.2 On-chip Regulator

CFP5102 provides an on-chip low drop-out regulator (LDO) to convert from 5.0 V to 3.3 V for internal core power use. To use internal LDO, LDOE pin must be tied to 0V in order to enable the LDO externally. If user wants to use external 3.3 V power supply to DVDD pin, it is necessary to tie LDOE to VDDIO to disable the LDO. The LDO employs pole-zero cancellation frequency compensation technique through external small resistor for the stability of the LDO.

Frequency compensation through external RC components

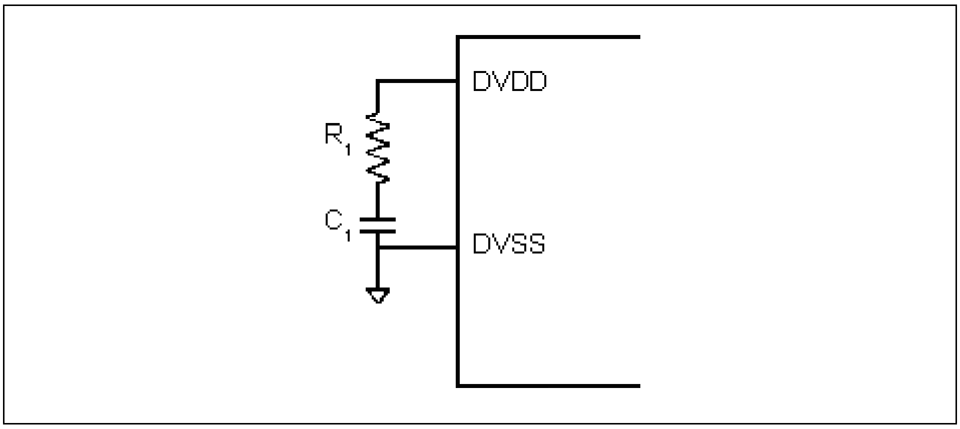


Figure 6-8: Frequency compensation using RC components

Table 6-4: Recommendation for RC values

R_1	C_1
1Ω	1uF
0.5Ω	2uF
0.25Ω	4.7uF
0.1Ω	10uF
0.05Ω	20uF
0.5Ω	1uF

6.3.3 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 IDL, bit 0 of PCON to '1'. Any enabled interrupt and reset sources can resume the core clock and deactivate idle mode.

PCON: **Processor Control Register**
TYPE: **R / W**
ADDRESS: **0x87**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic								IDL
Default								0

Name	Mode	Description	Setting
IDL	RW	Idle mode enable bit Write '1' to this bit to enter idle mode. This bit is reset to '0' automatically after wakeup	0: disable 1: enable

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

6.3.4 Halt Mode

Halt mode aggressively shuts down the whole system clock from the "stop clock" switch showing in Figure 6-1. At this moment, there is no activity in any module. Halt mode is activated by setting HLT, bit 5 of PCON to '1'. It is recommended to disable all peripherals and also DPLL. Only selected interrupt and reset sources can resume the clocks and deactivate halt mode.

PCON: **Processor Control Register**
TYPE: **R / W**
ADDRESS: **0x87**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic			HLT					
Default			0					

Name	Mode	Description	Setting
HLT	RW	Halt mode enable bit Write '1' to this bit to enter halt mode. This bit is reset to '0' automatically after wakeup	0: disable 1: enable

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

6.3.5 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 CFP5102 has dual crystal oscillators, the condition of power down is tabulated in Table 6-5. The procedures of stopping crystal oscillators are shown in Section 6.1.3. 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 6-5: 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

6.3.6 Additional Low Power Options

To tackle the static power consumption in OTP and crystal oscillators, 2 additional low power options are designed.

OTP low power mode

This option suppresses the speed of the OTP to reduce its static power consumption. It suits for CPU clock less than 5 MHz. As long as the CPU clock reduces in this mode, OTP static power consumption reduces linearly.

CCON: **CPU Control Register**
TYPE: **R / W**
ADDRESS: **0xAB**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic			LPM					
Default			0					

Name	Mode	Description	Setting
LPM	RW	Low power mode enable bit. Reduce the power consumption by OTP memory. Caution: this mode can only be enabled when system clock is under 5 MHz	0: disable 1: enable

32,768 Hz crystal oscillator low power mode

This option puts 32,768 Hz crystal oscillator in low power mode, at the cost of reducing compatibility to different crystals. The power factor can be reduced four times compared to normal mode in oscillator.

CKCON0: **Clock Control Register 0**
TYPE: **R / W**
ADDRESS: **0x91**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic			LPOSC					
Default			0					

Name	Mode	Description	Setting
LPOSC	RW	32,768 Hz oscillator low power mode enable bit For low power mode, set this bit before enable 32,768 Hz oscillator	0: disable 1: enable

7 Ports and Peripherals

7.1 Ports

CFP5102 has 32 IO pins organized as 4 byte-wide ports. User can completely control over the operation mode, the direction, the presence of pull-up of each pin individually through control registers. The configuration of the control registers are described in the following section.

CFP5102 are designed to work in both 3.3V and 5V systems and all port pins can output either 3.3V or 5V depending on VDDIO voltage.

P3.2 and P3.3 are used to enter special mode of CFP5102. Their voltage levels at the pads are captured in the reset state to determine the working mode. In normal operation, P3.3 must be kept at high voltage level during reset. If it is not desired to force the CFP5102 to use external memory interface, P3.2 should be also kept high during reset.

7.1.1 Push-Pull Mode and Open-Drain Mode

There are two operation modes of the port for interfacing different off-chip peripherals. The default one is push-pull mode showing in Figure 7-1. In the initial state after reset, all pins are configured as input. All pull-ups are disabled and no port synchronizer is in use.

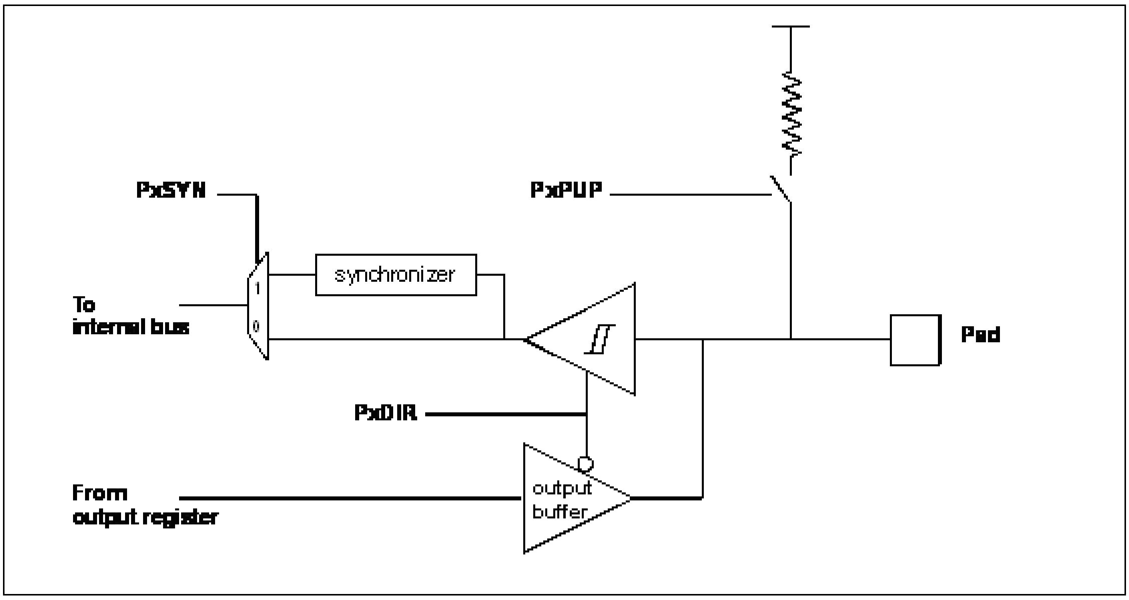


Figure 7-1: Block diagram of push-pull mode

The other operation mode of the port is open-drain mode, showing in Figure 7-2. This mode is selected by manipulating registers P0OD, P1OD, P2OD and P3OD. In addition, the corresponding pins have to be configured as input (refer to Section 7.1.2). Open-drain mode does not enable the internal pull-ups automatically. User should enable pull-ups in registers P0PUP, P1PUP, P2PUP and P3PUP (refer to Section 7.1.4).

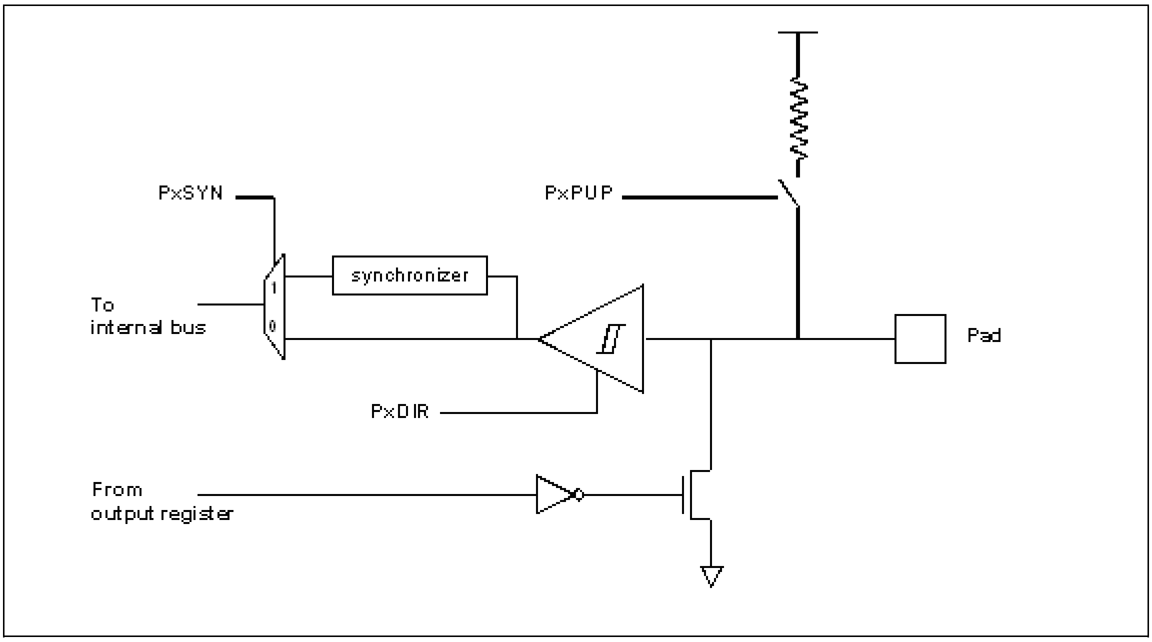


Figure 7-2: Block diagram of open-drain mode

PxDIR	Description	Type	Address	Default
P0OD	Port 0 Open-drain Mode Register	R/W	0xA4	0x00
P1OD	Port 1 Open-drain Mode Register	R/W	0xA5	0x00
P2OD	Port 2 Open-drain Mode Register	R/W	0xA6	0x00
P3OD	Port 3 Open-drain Mode Register	R/W	0xA7	0x00

Name	Mode	Description	Setting
PxOD[7]	RW	Port x pin 7 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[6]	RW	Port x pin 6 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[5]	RW	Port x pin 5 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[4]	RW	Port x pin 4 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[3]	RW	Port x pin 3 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[2]	RW	Port x pin 2 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[1]	RW	Port x pin 1 open-drain mode select bit	0: push-pull mode 1: open-drain mode
PxOD[0]	RW	Port x pin 0 open-drain mode select bit	0: push-pull mode 1: open-drain mode

7.1.2 Setting Port Direction

In push-pull mode, the direction of the ports is defined in registers P0DIR, P1DIR, P2DIR and P3DIR. 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.

In open-drain mode, port output is always available. With regards of this, registers P0DIR, P1DIR, P2DIR and P3DIR only control the input buffer. When the pin is set as output, the input buffer is disabled and the readout from the pin is always ‘0’. To obtain the voltage level at the pad, keep the input buffer on and write ‘1’ to port data register to force the pad to high impedance state before reading the port data register.

PxDIR	Description	Type	Address	Default
P0DIR	Port 0 Direction Control Register	R/W	0xAC	0xFF
P1DIR	Port 1 Direction Control Register	R/W	0xAD	0xFF
P2DIR	Port 2 Direction Control Register	R/W	0xAE	0xFF
P3DIR	Port 3 Direction Control Register	R/W	0xAF	0xFF

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	PxDIR[7:0]							

Name	Mode	Description	Setting	
PxDIR[7]	RW	Port x pin 7 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[6]	RW	Port x pin 6 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[5]	RW	Port x pin 5 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[4]	RW	Port x pin 4 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[3]	RW	Port x pin 3 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[2]	RW	Port x pin 2 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[1]	RW	Port x pin 1 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer
PxDIR[0]	RW	Port x pin 0 direction control bit	Push-pull mode	0: output 1: input
			Open-drain mode	0: disable input buffer 1: enable input buffer

7.1.3 Reading from and Writing to Port

Port 0 to Port 3 are memory-mapped into the registers P0, P1, P2 and P3 in SFR space. Writing to a port data register sets the voltage levels of the corresponding port pins that have been configured to operate as outputs. Reading from a data register reads the voltage levels of the corresponding port pins.

There is major difference of reading the port values when the port is set as input or is set as output. When the port is set as output, the CPU will read the port value from port data register instead of the voltage level at the pad. When the port is set as input, the CPU will read the value from in directly instead of the port value from port data register. As a result, the user should be very careful when using ***Read-then-Write*** instructions to access the ports and change port direction control register before write the output value to port data register when using port as output. For example:

```
Assembler:
ANL P0DIR, #0FEH
MOV P0, #01h
C language:
P0DIR &= 0xFE;
P0 = 0x01;
```

The first instruction in this example configures P0.0 as output, and then the second instruction writes the Port 0 data register (P0), which controls the output levels of the P0.0.

There are port synchronizers associate to each port. These synchronizers are controlled by register PSYN. When the port synchronizers are enabled, the voltage level at the pin will be passed to port synchronizers, and the readout is the output of the synchronizers instead of the voltage level at the pad.

Px	Description	Type	Address	Default
P0	Port 0 Data Register	R/W	0x80	0xXX
P1	Port 1 Data Register	R/W	0x90	0xXX
P2	Port 2 Data Register	R/W	0xA0	0xXX
P3	Port 0 Data Register	R/W	0xB0	0xXX

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	Px[7:0]							

Name	Mode	Description	Setting
Px	RW	Port x data register In output mode: - outputs data to pads when writing to Px - reads in register data when reading Px In input mode: - output to pads is prohibited - reads in data at pads when reading Px	

PSYN: **Port Synchronizer Enable Register**
TYPE: **R / W**
ADDRESS: **0xB6**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**-**	**-**	**-**	**-**	**P3SYN**	**P2SYN**	**P1SYN**	**P0SYN**
Default	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Name	Mode	Description	Setting
P3SYN	RW	Port 3 synchronizers enable bit	0: disable 1: enable
P2SYN	RW	Port 2 synchronizers enable bit	0: disable 1: enable
P1SYN	RW	Port 1 synchronizers enable bit	0: disable 1: enable
P0SYN	RW	Port 0 synchronizers enable bit	0: disable 1: enable

7.1.4 Using Pull-Ups

Each port pin associates with a 20Kohm pull-up resistor. The pull-up is disabled by default and it is enabled through register P0PUP, P1PUP, P2PUP and P3PUP. 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.

PxPUP	**Description**	**Type**	**Address**	**Default**
P0PUP	Port 0 Pull-up Control Register	R/W	0xB1	0x00
P1PUP	Port 1 Pull-up Control Register	R/W	0xB2	0x00
P2PUP	Port 2 Pull-up Control Register	R/W	0xB3	0x00
P3PUP	Port 3 Pull-up Control Register	R/W	0xB4	0x0C

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**PxPUP[7:0]**							

Name	Mode	Description	Setting
PxPUP[7]	RW	Port x pin 7 pull-up enable bit	0: disable 1: enable
PxPUP[6]	RW	Port x pin 6 pull-up enable bit	0: disable 1: enable
PxPUP[5]	RW	Port x pin 5 pull-up enable bit	0: disable 1: enable
PxPUP[4]	RW	Port x pin 4 pull-up enable bit	0: disable 1: enable
PxPUP[3]	RW	Port x pin 3 pull-up enable bit	0: disable 1: enable
PxPUP[2]	RW	Port x pin 2 pull-up enable bit	0: disable 1: enable
PxPUP[1]	RW	Port x pin 1 pull-up enable bit	0: disable 1: enable
PxPUP[0]	RW	Port x pin 0 pull-up enable bit	0: disable 1: enable

Caution:
(1) In push-pull mode, the pull-up resistor can be enabled only when input

7.1.5 Configuring for Analog Input

To configurable 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 open-drain mode
2. Write ‘1’ to port data register to force the pad to high impedance state
3. Disable the input buffer by clearing port direction control register

After the above procedure, the digital buffers of the pin are completely off. At this moment, analog channel can be enabled by manipulating register P0AIE, P1AIE, P2AIE and P3AIE. To avoid leakage current, always keeps analog channels disabled when it is not being used.

P0AIE: **Port 0 Analog Input Enable Register**
TYPE: **R/W**
ADDRESS: **0xE1**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	P0AIE[7:0]							
Default	00000000							

P1AIE: **Port 1 Analog Input Enable Register**
TYPE: **R/W**
ADDRESS: **0xE2**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	-	P1AIE[2]	-	-
Default	0	0	0	0	0	0	0	0

P3AIE: **Port 3 Analog Input Enable Register**
TYPE: **R/W**
ADDRESS: **0xE4**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	-	-	-	P3AIE[0]
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
PxAIE[7]	RW	Port x pin 7 analog input channel enable bit	0: disable 1: enable
PxAIE[6]	RW	Port x pin 6 analog input channel enable bit	0: disable 1: enable
PxAIE[5]	RW	Port x pin 5 analog input channel enable bit	0: disable 1: enable
PxAIE[4]	RW	Port x pin 4 analog input channel enable bit	0: disable 1: enable
PxAIE[3]	RW	Port x pin 3 analog input channel enable bit	0: disable 1: enable
PxAIE[2]	RW	Port x pin 2 analog input channel enable bit	0: disable 1: enable
PxAIE[1]	RW	Port x pin 1 analog input channel enable bit	0: disable 1: enable
PxAIE[0]	RW	Port x pin 0 analog input channel enable bit	0: disable 1: enable

Caution:
(1) To avoid leakage current, always keeps analog input channel disable when it is not being used.

7.1.6 IO Shared with Peripherals

Most port IO are shared with different peripherals. The functionality of the pins switch when the corresponding peripheral is enabled. (please refer to operation flow of each peripheral for details.) Table 7-1 shows the allocation of IO to the peripherals.

Table 7-1: The list of IO shared with peripherals

Port Pin	Serial Interface	External Interrupt	Waveform Output	Analog Peripherals
P0.0				ADC p.92
P0.1				
P0.2				
P0.3				
P0.4				
P0.5				
P0.6				
P0.7				
P1.0		Port Wakeup p.62	Clock Output p.47	
P1.1	GPSI p.92			
P1.2				ADC
P1.3				
P1.4				
P1.5				
P1.6				
P1.7	SPI p.80			
P2.0				
P2.1				
P2.2			8-bit PWM p.73	
P2.3				
P2.4				
P2.5				
P2.6			16-bit PWM p.73	
P2.7				
P3.0	UART p.80			Reference Output
P3.1				
P3.2		8051 External Interrupt p.64		
P3.3				
P3.4				
P3.5				
P3.6				
P3.7				

7.1.7 Interrupt and Wakeup

There are 4 dedicated circuits associating with P0.5, P0.6, P0.7 and USB data pin USBDP for capturing transitions at the pad. When an interested transition (defined in register WKEDG) appears at the pad, these circuits generate a port wakeup interrupt pending. If the corresponding interrupt is enabled (EPW = 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 0.

These circuits are controlled by port wakeup control register, WKEN. The detecting event is defined in register 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.

WKEN: **Port Wakeup Control Register**
TYPE: **R / W**
ADDRESS: **0xBC**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**-**	**-**	**-**	**-**	**WKEN[3:0]**			
Default	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Name	Mode	Description	Setting
WKEN[3]	RW	USB OTG DP wakeup enable bit	0: disable 1: enable
WKEN[2]	RW	P0.7 wakeup enable bit	0: disable 1: enable
WKEN[1]	RW	P0.6 wakeup enable bit	0: disable 1: enable
WKEN[0]	RW	P0.5 wakeup enable bit	0: disable 1: enable

Caution:

(1) The wakeup pending of port wakeup are in undefined state after reset. User must clear the pending bit before using port wakeup as interrupt. Otherwise unexpected interrupt will be resulted.

(2) Procedure of using port wakeup as interrupt
i. disable the interrupt if it is enabled
ii. select wakeup event (rising or falling edges)
iii. assign priority
iv. clear the wakeup pending ***MUST!***
v. enable the interrupt

WKPND: **Port Wakeup Pending Register**
TYPE: **R/W**
ADDRESS: **0xBB**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	WKPND[3:0]			
Default	0	0	0	0	X	X	X	X

Name	Mode	Description	Setting
WKPND[3]	RW	USB OTG DP event occurs. Interrupt source. Clear by software	0: inactive 1: active
WKPND[2]	RW	P0.7 wakeup event occurs. Interrupt source. Clear by software	0: inactive 1: active
WKPND[1]	RW	P0.6 wakeup event occurs. Interrupt source. Clear by software	0: inactive 1: active
WKPND[0]	RW	P0.5 wakeup event occurs. Interrupt source. Clear by software	0: inactive 1: active

WKEDG: **Port Wakeup Edge Select Register**
TYPE: **R/W**
ADDRESS: **0xBD**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	WKEDG[3:0]			
Default	0	0	0	0	X	X	X	X

Name	Mode	Description	Setting
WKEDG[3]	RW	Select the wakeup event of USB OTG DP	0: rising edge 1: falling edge
WKEDG[2]	RW	Select the wakeup event of P0.7	0: rising edge 1: falling edge
WKEDG[1]	RW	Select the wakeup event of P0.6	0: rising edge 1: falling edge
WKEDG[0]	RW	Select the wakeup event of P0.5	0: rising edge 1: falling edge

IEN1: **Interrupt Enable Register 1**
TYPE: **R/W**
ADDRESS: **0xA9**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic								EPW
Default								0

Name	Mode	Description	Setting
EPW	RW	Port wakeup interrupt enable bit	0: disable 1: enable

Classic 8051/8052 external interrupt

CFP5102 reserves the external interrupts of classic 8051/8052. The related control registers are summarized below. Unlike CFP5102's port wakeup circuitry, these external interrupts cannot be wakeup source.

TCON: **Timer Control Register**
TYPE: **R/W**
ADDRESS: **0x88**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic					IE1	IT1	IE0	IT0
Default					0	0	0	0

Name	Mode	Description	Setting
IT0	RW	External interrupt 0 type select bit This bit selects whether the configured INT0(P3.2) signal will detect falling edge or active-low	0: level triggered. 1: edge triggered
IE0	RW	External Interrupt 0 This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. This flag is the inverse of the INT0 input signal's logic level when IT0 = 0	0: inactive 1: Interrupt pending
IT1	RW	External interrupt 1 type select bit This bit selects whether the configured INT0(P3.3) signal will detect falling edge or active-low	0: level triggered. 1: edge triggered
IE1	RW	External Interrupt 1 This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT1 = 1. This flag is the inverse of the INT1 input signal's logic level when IT1 = 0	0: inactive 1: Interrupt pending

IEN0: **Interrupt Enable Register 0**
TYPE: **R/W**
ADDRESS: **0xA8**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic						EX1		EX0
Default						0		0

Name	Mode	Description	Setting
EX1	RW	External interrupt 1 enable bit	0: disable 1: enable
EX0	RW	External interrupt 0 enable bit	0: disable 1: enable

Caution:

(1) The wakeup pending of port wakeup are in undefined state after reset. User must clear the pending bit before using port wakeup as interrupt. Otherwise unexpected interrupt will be resulted.

(2) Procedure of using port wakeup as interrupt
i. disable the interrupt if it is enabled
ii. select wakeup event (low level or falling edges)
iii. assign priority
iv. clear the wakeup pending ***MUST!***
v. enable the interrupt

7.2 Timer 0/1

Timer 0 and Timer 1 are accessed and controlled through SFRs. Each counter/timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control (TCON) register is used to enable Timer 0 and Timer 1 as well as indicate their status. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits M1-M0 in the Counter/Timer Mode (TMOD) register. Each timer can be configured independently. Following is a detailed description of each operating mode.

7.2.1 Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as a 13-bit counter/timer in Mode 0. The following describes the configuration and operation of Timer0. However, both timers operate identically and Timer 1 is configured in the same manner as described for Timer0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if enabled. The C/T0 bit (TMOD.2) selects the counter/timer's clock source. Clearing C/T selects the system clock as the input for the timer. When C/T0 is set to logic 1, high-to-low transitions at the selected input pin increment the timer register.

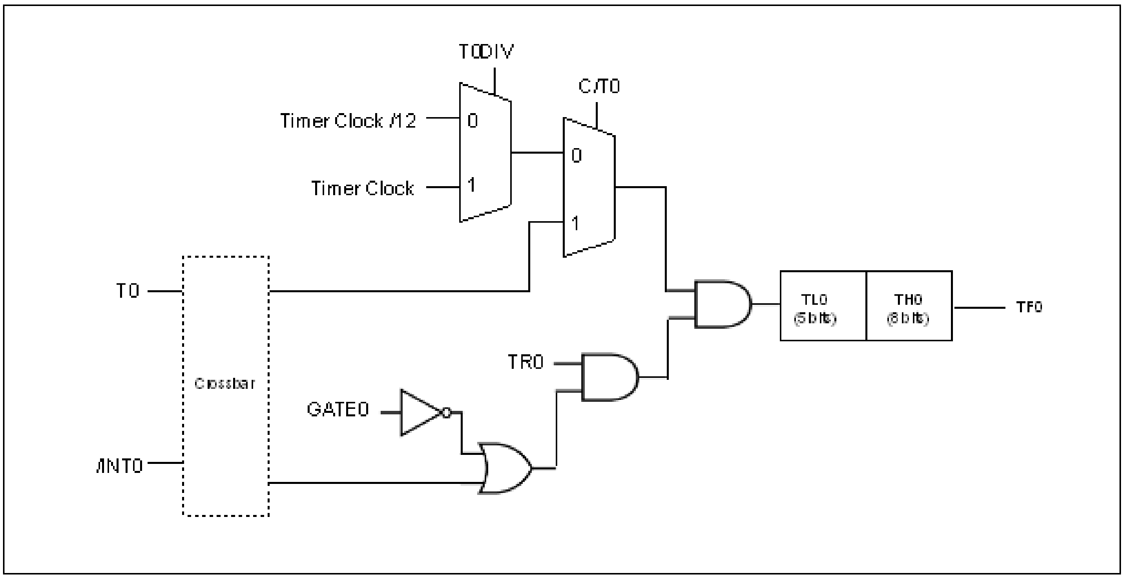


Figure 7-3: Block diagram of mode 0

7.2.2 Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

7.2.3 Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. The TL0 holds the count and TH0 holds the reload value. When the count in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer1 operates identically to Timer0. Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0.

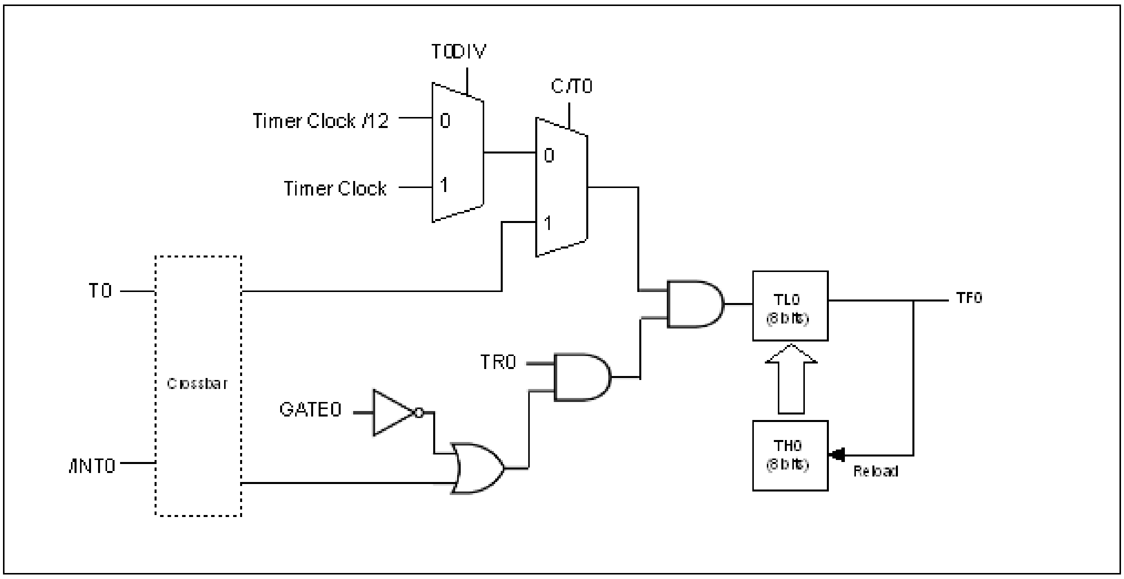


Figure 7-4: Block diagram of mode 2

7.2.4 Mode 3: Two 8-bit Counter/Timers (Timer0 Only)

Timer 0 and Timer 1 behave differently in Mode 3. Timer0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. It can use either the system clock or an external input signal as its time base. The TH0 register is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer1 run control bit TR1. TH0 sets the Timer1 overflow flag TF1 on overflow and thus controls the Timer1 interrupt. Timer1 is inactive in Mode 3, so with Timer0 in Mode 3, Timer1 can be turned off and on by switching it into and out of its Mode 3. When Timer0 is in Mode 3, Timer1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer1 overflow can be used for baud rate generation. Refer to Section 7.5 (serial port) for information on configuring Timer1 for baud rate generation.

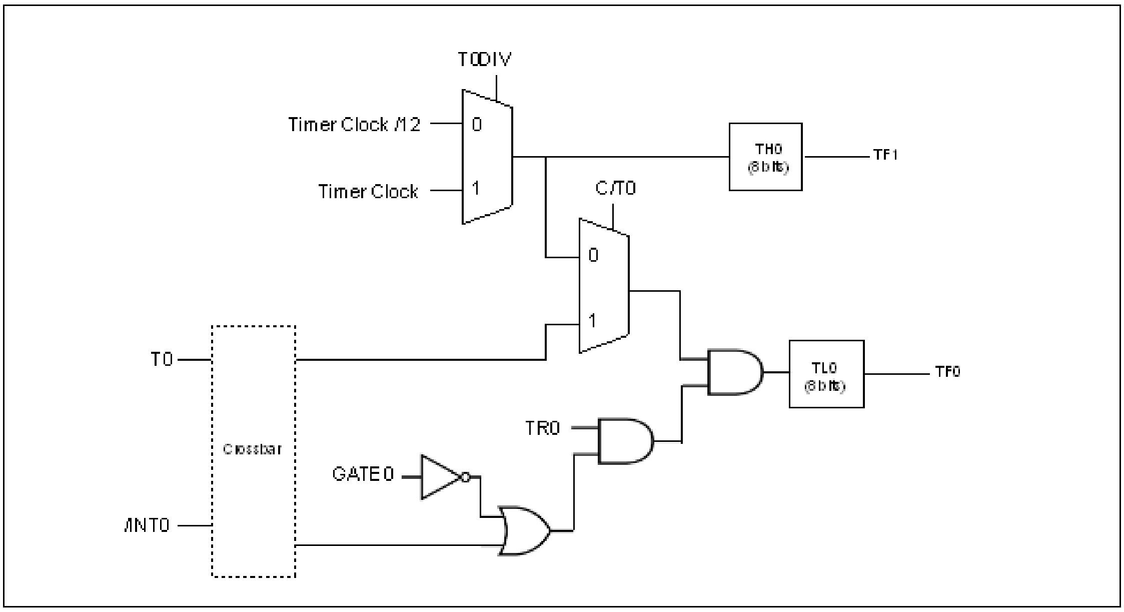


Figure 7-5: Block diagram of mode 3

TCON: **Timer Control Register**
TYPE: **R / W**
ADDRESS: **0x88**

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**TF1**	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
Default	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Name	Mode	Description	Setting
TF1	RW	Timer 1 overflow flag Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.	0: inactive 1: overflow occurs
TR1	RW	Timer 1 run control	0: timer 1 disable 1: timer 1 enable
TF0	RW	Timer 0 overflow flag Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.	0: inactive 1: overflow occurs
TR0	RW	Timer 0 run control	0: timer 0 disable 1: timer 0 enable
IE1	RW	External Interrupt 1 This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT1 = 1. This flag is the inverse of the INT1 input signal's logic level when IT1 = 0	0: inactive 1: Interrupt pending
IT1	RW	External interrupt 1 type select bit This bit selects whether the configured INT0(P3.3) signal will detect falling edge or active-low	0: level triggered. 1: edge triggered
IE0	RW	External Interrupt 0 This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0	0: inactive 1: Interrupt pending

		= 1. This flag is the inverse of the INT0 input signal's logic level when IT0 = 0	
IT0	RW	External interrupt 0 type select bit This bit selects whether the configured INT0(P3.2) signal will detect falling edge or active-low	0: level triggered. 1: edge triggered

TMOD: **Timer Mode Register**
TYPE: **R/W**
ADDRESS: **0x89**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	GATE1	C/T1	T1M[1]	T1M[0]	GATE0	C/T0	T0M[1]	T0M[0]
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
GATE1	RW	Timer 1 Gate control When set to '1' timer 1 is enabled only when TR1 = 1 AND INT1 = logic level one; While clear to '0' timer 1 is enabled when TR1 = 1 irrespective of INT1 (P3.3) logic level.	
C/T1	RW	Counter/Timer 1 select bit Timer Function: Timer 1 incremented by clock defined by FSET[2]. Counter Function: Timer 1 incremented by high-to-low transitions on external input pin T1(P3.5)	0: timer function 1: counter function
T1M	RW	Timer 1 Mode Select	

T1m	Mode
00	Mode 0: 13-bit counter/timer
01	Mode 1: 16-bit counter/timer
10	Mode 2: 8-bit counter/timer with auto-reload
11	Mode 3: Timer 1 Inactive/stopped

GATE0	RW	Timer 0 Gate control When set to '1' timer 1 is enabled only when TR1 = 1 AND INT1 = logic level one; While clear to '0' timer 1 is enabled when TR1 = 1 irrespective of INT1 (P3.3) logic level.	
C/T0	RW	Counter/Timer 0 select bit Timer Function: Timer 0 incremented by clock defined by FSET[6]. Counter Function: Timer 0 incremented by high-to-low transitions on external input pin T0(P3.4)	0: timer function 1: counter function
T0M	RW	Timer 0 Mode Select	

T0M	Mode
00	Mode 0: 13-bit counter/timer
01	Mode 1: 16-bit counter/timer
10	Mode 2: 8-bit counter/timer with auto-reload
11	Mode 3: Timer 1 Inactive/stopped

FSET: **Special Function Setting Register**
TYPE: **R/W**
ADDRESS: **0xBF**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic		T0DD				T1DD		
Default		1				1		

Name	Mode	Description	Setting
T0DIV	RW	Timer 0 clock divider select bit	0: system clock / 12 1: system clock
T1DIV	RW	Timer 1 clock divider select bit	0: system clock / 12 1: system clock

TL0: **Timer 0 Counter Low Byte Register**
TYPE: **R / W**
ADDRESS: **0x8A**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TL0							
Default	0x00							

TL1: **Timer 1 Counter Low Byte Register**
TYPE: **R / W**
ADDRESS: **0x8B**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TL1							
Default	0x00							

TH0: **Timer 0 Counter High Byte Register**
TYPE: **R / W**
ADDRESS: **0x8C**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TH0							
Default	0x00							

TH1: **Timer 1 Counter High Byte Register**
TYPE: **R / W**
ADDRESS: **0x8D**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TH1							
Default	0x00							

IEN0: **Interrupt Enable Register 0**
TYPE: **R / W**
ADDRESS: **0xA8**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic					ET1		ET0	
Default					0		0	

Name	Mode	Description	Setting
ET1	RW	Timer 1 interrupt enable bit	0: disable 1: enable
ET0	RW	Timer 0 interrupt enable bit	0: disable 1: enable

7.3 Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate as a timer or an event counter. This is selectable by bit C/T2 in the SFR T2CON. Timer 2 consists of two 8-bit registers, TH2 and TL2. In Timer function, TL2 register is incremented every clock cycle. In Counter function, TL2 register is incremented in response to a 1-to-0 transition at its external input pin T2 (P2.4). Timer 2 has three operating modes: capture, auto-reload (up or down counting) and PWM.

7.3.1 Capture Mode

In capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which sets bit TF2 in T2CON when overflow occurs. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input pin T2EX (P2.5) also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 7-6.

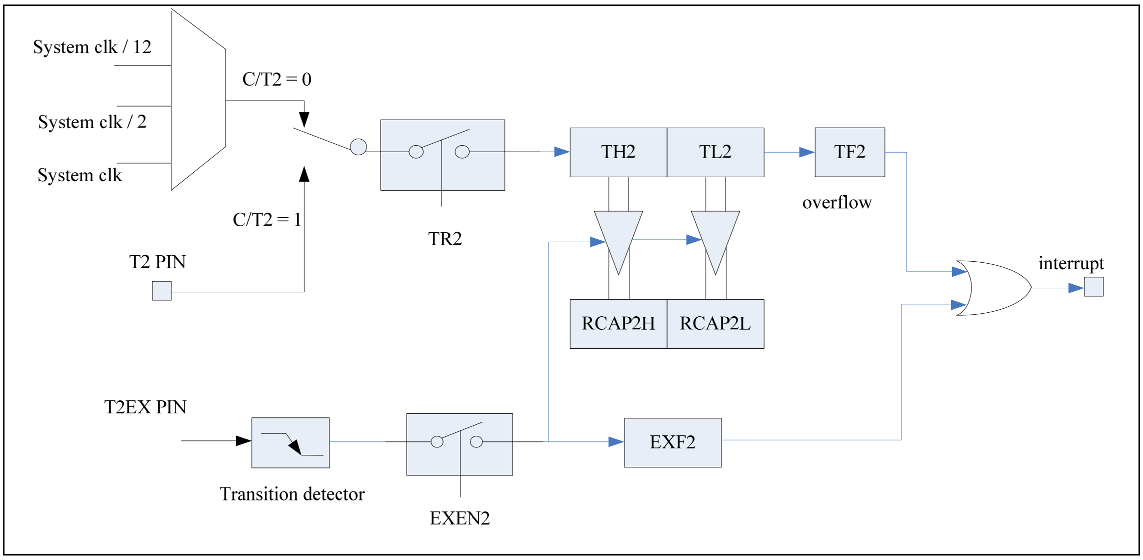


Figure 7-6: Block diagram of timer 2 in capture mode

7.3.2 Auto-reload Mode (Up or Down Counter)

Timer 2 can be programmed to count up or down in 16-bit auto-reload mode by using a bit named DCEN (Down Counter Enable) located in SFR T2MOD. Upon reset, DCEN bit is set to 0 so that Timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down depending on the value of T2EX pin.

Figure 7-7 shows that Timer 2 is counting up when DCEN=0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to

0FFFFH and sets TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L can be defined by software. If EXEN2 = 1, a 16-bit reload can be triggered by either an overflow or a 1-to-0 transition at external pin T2EX. This transition also sets the EXF2 bit. Both TF2 and EXF2 can generate an interrupt if enabled.

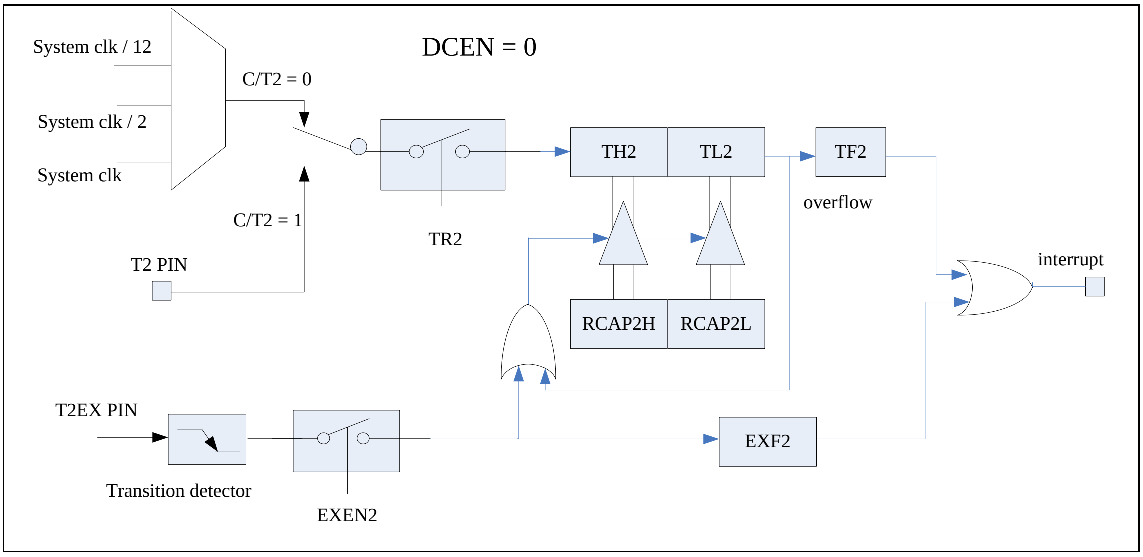


Figure 7-7: Block diagram of timer 2 in auto-reload mode when DCEN = 0

Figure 7-8 shows that Timer 2 can count up or down when DCEN=1. In this mode, T2EX pin controls the counting direction. Logic 1 at T2EX enables Timer 2 to count up. The timer will overflow at 0FFFFH and set TF2 bit. The overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded to TH2 and TL2, respectively.

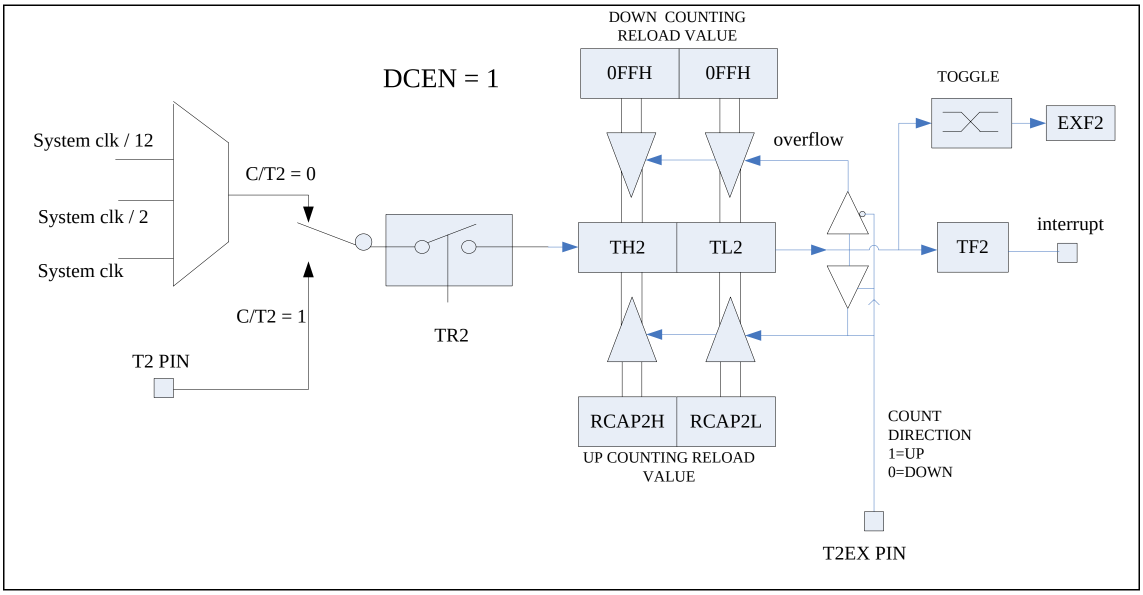


Figure 7-8: Block diagram of timer 2 in auto-reload mode when DCEN = 1

Logic 0 at T2EX enables Timer 2 to count down. In this case, Timer 2 underflows when TH2 and TL2 equal to the values stored in RCAP2H and RCAP2L. The underflow sets TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows only when DCEN=1. This bit can be used as the 17th bit for the timer registers if desired. In this case, EXF2 does not trigger an interrupt. Also, capture mode is not available when DCEN=1.

7.3.3 Pulse Width Modulation (PWM)

Timer 2 PWM has two operating modes: 8-bit and 16-bit modes. In 16-bit mode, TPR2L and TPR2H become a 16-bit period register (TPR2). Similarly, RCAP2L and RCAP2H become a 16-bit duty cycle register (RCAP2). Figure 7-9 shows the Timer 2 operation in 16-bit PWM mode.

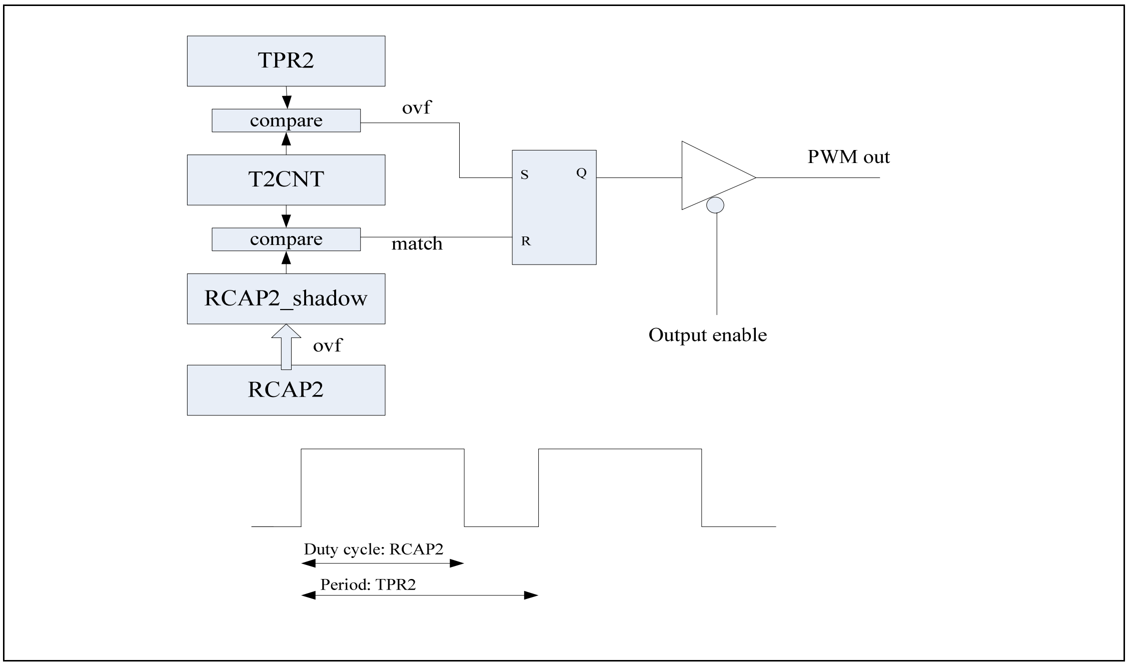


Figure 7-9: Block diagram of timer 2 in pulse width modulation mode

Timer register TL2 is incremented every clock cycle. When value of timer register is equal to period register TPR2, PWM output will be set to 1 and timer register will be reset to 0000H. The content of RCAP will be reloaded into a shadow register for comparison purpose. When timer register reaches the value of duty cycle register RCAP2, PWM output will be set to 0. Bit T2OE in T2MOD is the master control for all PWM output signals. Once T2OE is set, PWM16P bit and PWM16N bit can control the output at PWMP (P2.6) and PWMN (P2.7), respectively. When PWM16P bit is set, PWMOP will output PWM signal according to the settings of TPR2 and RCAP2. When

PWM16N bit is set, PWMN will output an inverted signal of the original PWMP. Both PWM16P and PWM16N can be set simultaneously.

PWM 8-bit mode works exactly the same as PWM 16-bit mode, except that the 16-bit period register TPR2 and duty cycle register RCAP2 are divided into two 8-bit registers to separately control two 8-bit PWM output signals. The 8-bit period register TPR2L and 8-bit duty cycle register RCAP2L control the output of PWM1 (P2.2). The 8-bit period register TPR2H and 8-bit duty cycle register RCAP2H control the output of PWM2 (P2.3). When PWM8 bit and T2OE bit are set, it enables the output of PWM1 and PWM2.

T2CON: Timer 2 Control Register
TYPE: R/W
ADDRESS: 0xC8

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TF2	EXF2	-	-	EXEN2	TR2	C/T2	CP/RL2
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
TF2	RW	Timer 2 overflow flag is set when Timer 2 overflows and must be cleared by software.	0: inactive 1: overflow occurs
EXF2	RW	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2=1. When Timer 2 interrupt is enabled, EXF2=1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN=1).	0: inactive 1: Negative transition on T2EX occurs
EXEN2	RW	Timer 2 external event enable. When EXEN2=1, it allows a capture or reload to occur as a result of a negative transition on T2EX. EXEN2=0 causes Timer 2 to ignore events at T2EX.	0: Timer 2 ignores external events at T2EX 1: Timer 2 detects external events at T2EX
TR2	RW	Start/Stop control for Timer 2. TR2=1 starts the timer.	0: Disable Timer 2 1: Enable Timer 2
C/T2	RW	Timer 2 counter/timer select bit Timer Function: Timer 2 is incremented every clock cycle. Counter Function: Timer 2 is incremented by a negative transition on external input pin T2 (P2.4).	0: Timer function 1: Counter function
CP/RL2	RW	Capture/Reload select bit Capture mode: Timer 2 captures on negative transitions at T2EX if EXEN2=1. Reload mode: Timer 2 reloads automatically when Timer 2 overflows or negative transitions occur at T2EX when EXEN2=1	0: Reload mode 1: Capture mode

T2MOD: Timer 2 Mode Register
TYPE: R/W
ADDRESS: 0xC9

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	PWM8	T2CLK[1]	T2CLK[0]	PWM16N	PWM16P	T2OE	DCEN
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
PWM8	RW	8-bit PWM output enable bit. When PWM8 is set, it	0: Disable 8-bit PWM

		enables 8-bit PWM output signals at PWM1 (P2.2) and PWM2 (P2.3).	output 1: Enable 8-bit PWM output
T2CLK[1:0]	RW	Timer 2 clock divider select	00: system clock 01: system clock / 2 10: system clock / 12 11: inactive
PWM16N	RW	16-bit PWM inverted output enable bit. When PWM16N is set, it enables 16-bit PWM output signal at PWMN (P2.7).	0: Disable output at PWMON 1: Enable output at PWMON
PWM16P	RW	16-bit PWM output enable bit. When PWM16P is set, it enables 16-bit PWM output signal at PWMP (P2.6).	0: Disable output at PWMOP 1: Enable output at PWMOP
T2OE	RW	Timer 2 PWM output enable bit. It is the master control of all 8-bit and 16-bit PWM output signals.	0: Disable PWM output 1: Enable PWM output
DCEN	RW	Down counter enable bit. When DCEN is set, Timer 2 can be configured as up/down counter depending on the value of T2EX pin.	0: Disable down counter 1: Enable down counter

RCAP2L: **Timer 2 Capture / Reload / PWM duty cycle Low Byte Register**
TYPE: **R / W**
ADDRESS: **0xCA**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	RCAP2L							
Default	0x00							

RCAP2HL: **Timer 2 Capture / Reload / PWM duty cycle High Byte Register**
TYPE: **R / W**
ADDRESS: **0xCB**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	RCAP2H							
Default	0x00							

TL2: **Timer 2 Counter Low Byte Register**
TYPE: **R / W**
ADDRESS: **0xCC**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TL2							
Default	0x00							

TH2: **Timer 2 Counter High Byte Register**
TYPE: **R / W**
ADDRESS: **0xCD**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TH2							
Default	0x00							

TPR2L: **Timer 2 PWM Period Low Byte Register**
TYPE: **R / W**
ADDRESS: **0xCE**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TPR2L							
Default	0xFF							

TPR2H: **Timer 2 PWM Period High Byte Register**
TYPE: **R / W**
ADDRESS: **0xCF**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	TPR2H							

Default	**0xFF**

IEN0: **Interrupt Enable Register 0**
TYPE: **R/W**
ADDRESS: **0xA8**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic			ET2					
Default			0					

Name	Mode	Description	Setting
ET2	RW	Timer 2 interrupt enable bit	0: disable 1: enable

7.3.4 Operation Flow

Capture or reload mode

1. Set counter value TH2 and TL2
2. Set limit value TPR2H and TPR2L
3. Set reload value RCAP2H and RCAP2L (reload mode only)
4. Configure T2MOD (DCEN and T2CLK)
5. Configure T2CON (EXEN2, C/T2, CP/RL2) and start Timer 2 (TR2=1)
6. Enable Timer 2 interrupt (set ET2 to 1)
7. When Timer 2 interrupt occurs (by external event, counter overflow or counter reaches limit value), clear Timer 2 overflow flag (TF2)
8. Read RCAP2H and RCAP2L if necessary (in capture mode only)

PWM mode

1. Disable Timer 2 interrupt (set ET2=0)
2. Set counter value TH2 and TL2
3. Set period register value TPR2H and TPR2L
4. Set duty cycle register RCAP2H and RCAP2L
5. Configure T2MOD (PWM8, T2CLK, PWM16N, PWM16P, T2OE, and set DCEN=0)
6. Configure T2CON (set EXEN2=0, C/T2=0, CP/RL2=0, and TF2=0) and start Timer 2 (TR2=1)

7.4 Real Time Wakeup

Real time wakeup module provides a regular, periodic interrupt 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 0/1 and Timer 2.

The clock source for real time wakeup module comes from 32,768 Hz oscillator. User should refer to *Clock System* in section 6.1 for more details on enabling or disabling 32,768 Hz clock. Once enabled, RTDIV controls frequency of the real time wakeup clock.

Since real time wakeup module has internal wakeup circuitry, it can work with IDLE, HALT and Power Down power-saving modes. The following scenarios may occur when real time wakeup module is enabled.

Normal running mode with interrupt enabled

When real time wakeup module is enabled (RTWE=1) and ERT is set to 1, it sets the pending flag RTWK and triggers an interrupt periodically according to the setting of RTDIV.

Idle mode with interrupt enabled

When real time wakeup module sets RTWK flag and triggers an interrupt, CFP5102 exits IDLE mode and enters interrupt service routine at vector 0x33. Once the routine is served, CFP5102 will run the next instruction immediately after entering IDLE mode.

Halt mode with interrupt enabled

Similar to in idle mode, the CFP5102 exits IDLE mode and enters interrupt service routine at vector 0x33.

Halt mode with interrupt disabled

Real time wakeup module sets RTWK flag and does not trigger an interrupt. CFP5102 will run the next instruction immediately after entering IDLE mode.

Power down mode

The module triggers a reset when real time wakeup event occurs.

Caution:

(1) As the 32,768 Hz clock runs freely after enabled, the first wakeup after enabling real-time wakeup is possibly earlier than expected. After that, the time between the second and the first wakeup will be exactly the period defined in RTDIV.

RTCON: **Real Time Control Register**
TYPE: **R/W**
ADDRESS: **0xD8**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	RTWK	-	-	RTWE	-	-	-
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
RTWK	RW	Real time wakeup pending flag It can be cleared by software.	0: inactive 1: Interrupt pending
RTWE	RW	Real time wakeup enable bit	0: disable 1: enable

IEN0: **Interrupt Enable Register 0**
TYPE: **R/W**
ADDRESS: **0xA8**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic		ERT						
Default		0						

Name	Mode	Description	Setting
ERT	RW	Real-time wakeup interrupt enable bit	0: disable 1: enable

7.4.1 Operation Flow

1. Enable 32,768 Hz clock (refer to section 6.1.3 for details)
2. Enable real time clock (set RTCKE to 1 in CKCON1)
3. Configure RTDIV
4. Clear real time wakeup pending flag (set RTWK to 0)
5. Enable real time wakeup module (set RTWE to 1)
6. Enable real time wakeup interrupt (set ERT to 1 in IEN0) if necessary
7. When real time wakeup event occurs, clear pending flag (set RTWK to 0) and ready for next event

7.5 Serial Port (UART)

UART is a serial port capable of asynchronous transmission. The UART0 can function in full duplex mode. In all modes, receive data is buffered in a holding register. This allows the UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

Receive pin (RXD) – P3.0

Transmit pin (TXD) – P3.1

UART provides four operating modes selected by setting configuration bits in the SCON register. These four modes offer different baud rates and communication protocols.

Table 7-2: Operation modes of UART

Mode	Synchronization	Baud Clock	Data Bits	Start/Sto Bits
0	Synchronous	System clock / 12	8	None
1	Asynchronous	Timer 1 overflow	8	1 Start, 1 Stop
2	Asynchronous	System clock / 32 or / 64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1 overflow	9	1 Start, 1 Stop

7.5.1 Mode 0

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX pin. The TX pin provides the shift clock for both transmit and receive. The MCU must be the master since it generates the shift clock for transmission in both directions.

Data transmission begins when an instruction writes a data byte to the SBUF register. Eight data bits are transferred LSB first, and the TI Transmit Interrupt Flag is set at the end of the eighth bit time. Data reception begins when the REN Receive Enable bit is set to logic 1 and the RI Receive Interrupt Flag is cleared. One cycle after the eighth bit is shifted in, the RI flag is set and reception stops until software clears the RI bit. An interrupt will occur if enabled when either TI or RI are set.

7.5.2 Mode 1

Mode 1 provides standard asynchronous, full duplex communication using a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted from the TX pin and received at the RX pin. On receive, the eight data bits are stored in SBUF and the stop bit goes into RB8.

Data transmission begins when an instruction writes a data byte to the SBUF register. The TI Transmit Interrupt Flag is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF receive register if the following conditions are met:

RI must be logic 0, and if SM2 is logic 1, the stop bit must be logic 1.

If these conditions are met, the eight bits of data are stored in SBUF, the stop bit is stored in RB8 and the RI flag is set. If these conditions are not met, SBUF and RB8 will not be loaded and the RI flag will not be set. An interrupt will occur if enabled when either TI or RI is set.

7.5.3 Mode 2

Mode 2 provides asynchronous, full-duplex communication using a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. Mode 2 supports multiprocessor communications. On transmit, the ninth data bit is determined by the value in TB8. It can be assigned the value of the parity flag P in the PSW or used in multiprocessor communications. On receive, the ninth data bit goes into RB8 and the stop bit is ignored. Data transmission begins when an instruction writes a data byte to the SBUF register. The TI Transmit Interrupt Flag is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF receive register if RI is logic 0 and one of the following requirements are met:

1. SM2 is logic 0

2. SM2 is logic 1, the received 9th bit is logic 1.

If the above conditions are satisfied, the eight bits of data are stored in SBUF, the ninth bit is stored in RB8 and the RI flag is set. If these conditions are not met, SBUF and RB8 will not be loaded and the RI flag will not be set. An interrupt will occur if enabled when either TI or RI is set.

7.5.4 Mode 3

Mode 3 uses the Mode 2 transmission protocol with the Mode 1 baud rate generation. Mode 3 operation transmits 11 bits: 1 start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. Multiprocessor communications is supported.

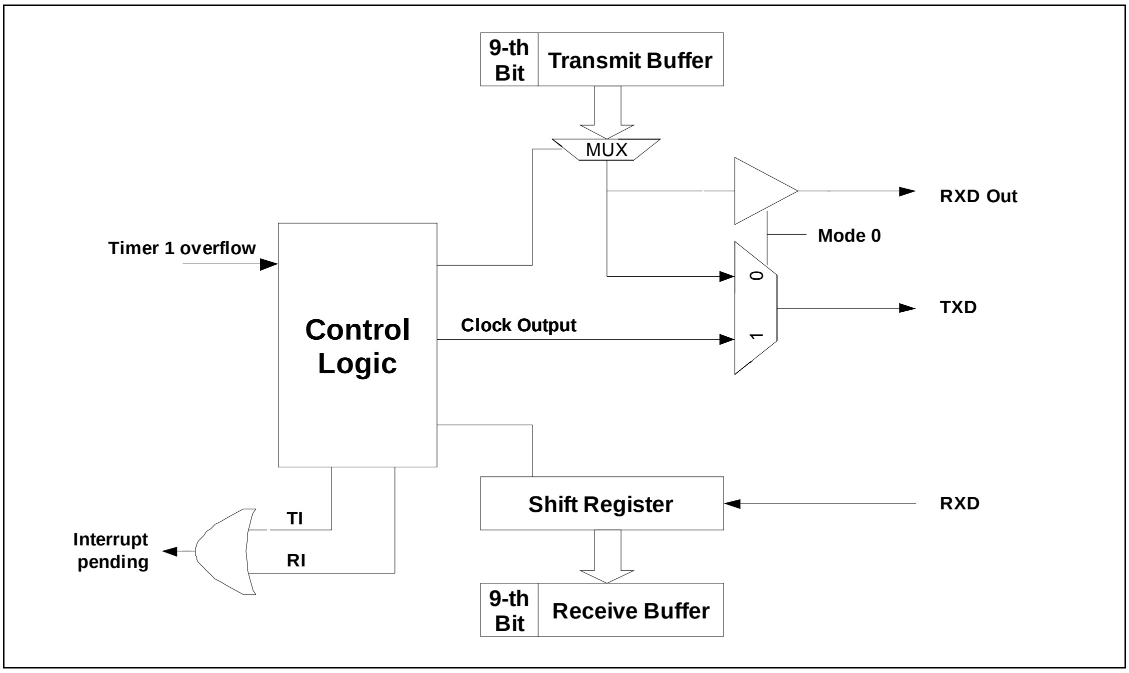


Figure 7-10: UART block diagram

SCON: Serial Port Control Register
TYPE: R/W
ADDRESS: 0x98

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
Default	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

<table>
<tr><th>Name</th><th>Mode</th><th>Description</th><th>Setting</th></tr>
<tr><td>SM0,
SM1</td><td>RW</td><td>Serial port operation mode</td><td>
<table>
<tr><th>SM0</th><th>SM1</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>mode 0, synchronous, fixed baudrate: system clock / 12</td></tr>
<tr><td>0</td><td>1</td><td>mode 1, 8 bit UART, Variable baudrate</td></tr>
<tr><td>1</td><td>0</td><td>mode 2, 9bit UART, fixed baudrate: system clock / 32 or system clock / 64</td></tr>
<tr><td>1</td><td>1</td><td>mode 3, 9bit UART, Variable baudrate</td></tr>
</table>
</td></tr>
<tr><td>SM2</td><td>RW</td><td colspan="2">Serial port multiprocessor communications bit
The function of this bit is dependent on the Serial Port Operation Mode
<table>
<tr><th>Mode</th><th>SM2</th><th></th></tr>
<tr><td>0</td><td>0 / 1</td><td>No effect</td></tr>
<tr><td>1</td><td>0</td><td>Logic level of stop bit is ignored</td></tr>
<tr><td></td><td>1</td><td>RI will only be activated if stop bit is logic level 1</td></tr>
<tr><td>2, 3</td><td>0</td><td>Logic level of ninth bit is ignored</td></tr>
<tr><td></td><td>1</td><td>RI is set and an interrupt is generated only when the ninth bit is logic 1</td></tr>
</table>
</td></tr>
<tr><td>REN</td><td>RW</td><td>Receive enable bit</td><td>0: disable
1: enable</td></tr>
<tr><td>TB8</td><td>RW</td><td>Ninth transmission bit
The logic level of this bit will be assigned to the ninth transmission bit in Modes 2 and 3.
It is not used in Modes 0 and 1. Set or cleared by software as required.</td><td></td></tr>
<tr><td>RB8</td><td>RW</td><td>Ninth receive bit
The bit is assigned the logic level of the ninth bit received in Modes 2 and 3.
In Mode 1, if SM2 is logic 0, RB8 is assigned the logic level of the received stop bit.
RB8 is not used in Mode 0.</td><td></td></tr>
<tr><td>TI</td><td>RW</td><td>Transmit interrupt flag
Please refer to Section 7.5.6 for details</td><td>0: inactive
1: Interrupt pending</td></tr>
<tr><td>RI</td><td>RW</td><td>Receive interrupt flag
Please refer to Section 7.5.6 for details</td><td>0: inactive
1: Interrupt pending</td></tr>
</table>

SBUF: Serial Port Buffer Register
TYPE: R/W
ADDRESS: 0x99

<table>
<tr><td><b>Position</b></td><td><b>Bit7</b></td><td><b>Bit6</b></td><td><b>Bit5</b></td><td><b>Bit4</b></td><td><b>Bit3</b></td><td><b>Bit2</b></td><td><b>Bit1</b></td><td><b>Bit0</b></td></tr>
<tr><td><b>Mnemonic</b></td><td colspan="8"><b>SBUF</b></td></tr>
<tr><td><b>Default</b></td><td colspan="8"><b>0x00</b></td></tr>
</table>

Description	Setting
Write this location will load the data to transmitter buffer. Read this location will read the data from the receiver buffer.	

7.5.5 Baudrate Setting

UART does not have a dedicated baudrate generator. It refers to the period of timer 1 overflow to generate local clock for both transmit and receive. The relationship of transmit and receive baudrate and the overflow frequency of timer 1 is as follows:

$$Baudrate = (2^{SMOD} \times f_{overflow})/32$$

PCON: **Processor Control Register**
TYPE: **R / W**
ADDRESS: **0x87**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SMOD							
Default	0							

Name	Mode	Description	Setting
SMOD	RW	Serial port baudrate select bit	0: frequency of timer 1 overflow / 32 1: frequency of timer 1 overflow / 16

7.5.6 Interrupt

UART asserts interrupts in two situations: (1) when one byte of data is received; (2) when one byte of data is transmitted. The corresponding pending flags are located in bit 0 and bit 1 of SCON. Interrupt enable bit ES controls both transmit and receive interrupt pending together. User should check the pending bit in the interrupt service subroutine to distinguish the source of interrupt.

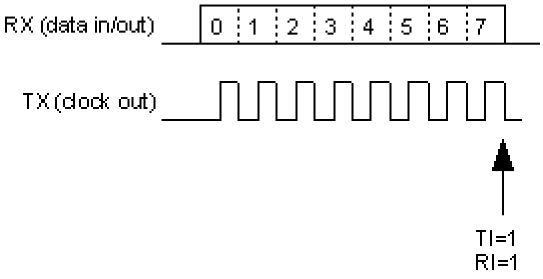


Figure 7-11 Interrupt timing for 8-bit synchronous mode (mode 0)

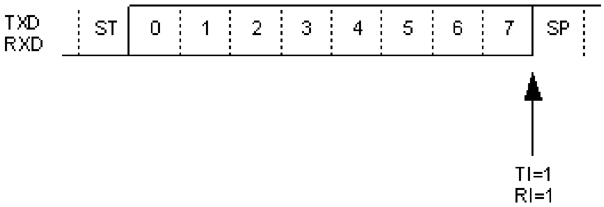


Figure 7-12 Interrupt timing for 8-bit asynchronous mode (mode 1)

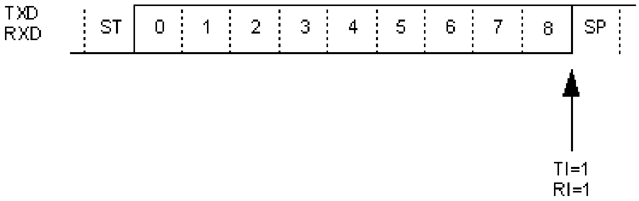


Figure 7-13 Interrupt timing for 9-bit asynchronous mode (mode 2, 3)

IEN0: **Interrupt Enable Register 0**
TYPE: **R/W**
ADDRESS: **0xA8**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic				ES				
Default				0				

Name	Mode	Description	Setting
ES	RW	Serial port interrupt enable bit	0: disable 1: enable

SCON: **Serial Port Control Register**
TYPE: **R/W**
ADDRESS: **0x98**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic							TI	RI
Default							0	0

Name	Mode	Description	Setting
TI	RW	Transmit interrupt flag Set by hardware when a byte of data has been transmitted by the UART (after the 8th bit in Mode 0 or at the beginning of the stop bit in other modes). When the UART interrupt is enabled, setting this bit causes the CPU to vector to the UART interrupt service routine. This bit must be cleared manually by software.	0: inactive 1: Interrupt pending
RI	RW	Receive interrupt flag Set by hardware when a byte of data has been received by the UART (after the 8th bit in Mode 0, or after the stop bit in other modes – see SM2 bit for exception). When the UART interrupt is enabled, setting this bit causes the CPU to vector to the UART interrupt service routine. This bit must be cleared manually by software.	0: inactive 1: Interrupt pending

7.5.7 Operation Flow

Mode 0 transmit data

1. Configure timer 1 and SMOD to generate desired baudrate
2. Write '1' to P3.0 port buffer
3. Set P3.0 and P3.1 as output
4. Configure SCON

5. Set ES and global interrupt enable EA if necessary
6. Fill transmit data to SBUF
7. Poll for TI changing to ‘1’, or wait for interrupt
8. Clear TI ,go to step 6 if intends for transmitting another byte of data

Mode 0 receive data

1. Configure timer 1 and SMOD to generate desired baudrate
2. Set P3.0 as input and P3.1 as output
3. Configure SCON
4. Set ES and global interrupt enable EA if necessary
5. Poll for RI changing to ‘1’, or wait for interrupt
6. Read received data from SBUF, clear RI
7. Go to step 5 to wait for another byte of data, or turn off UART by clearing REN and ES

Mode 1, 2, 3 transmit data

1. Configure timer 1 and SMOD to generate desired baudrate
2. Write ‘1’ to P3.0 port buffer
3. P3.1 as output
4. Configure SCON
5. Set ES and global interrupt enable EA if necessary
6. Fill transmit data to SBUF
7. Poll for TI changing to ‘1’, or wait for interrupt
8. Clear TI ,go to step 6 if intends for transmitting another byte of data

Mode 1, 2, 3 receive data

1. Configure timer 1 and SMOD to generate desired baudrate
2. Set P3.0 as input
3. Configure SCON
4. Set ES and global interrupt enable EA if necessary
5. Poll for RI changing to ‘1’, or wait for interrupt
6. Read received data from SBUF, clear RI
7. Go to step 5 to wait for another byte of data, or turn off UART by clearing REN and ES

7.6 Serial Protocol Interface (SPI)

SPI provides full-duplex, synchronous serial communication between CFP5102 and peripherals. It is a 3-wire protocol, consisting of clock (SCK), data output (SDO) and data input (SDI). The system architecture is illustrated in Figure 7-14: System diagram of SPI between two devices showing the connection of a master device and a slave device. CFP5102 uses the following pins for SPI.

Serial clock (SCK) – P1.7

Serial data input (SDI) – P2.0

Serial data output (SDO) – P2.1

It also supports 2-wire mode, which is a variation for reducing the footprint in on-board connections.

Serial clock (SCK) – P1.7

Serial data input / output (SD) – P2.0

The SPI can be configured to operate as master and slave by SM bit (bit 6 of SPICON). The transmission format can also be defined through SPICON. The details of master and slave mode are described in the following sections.

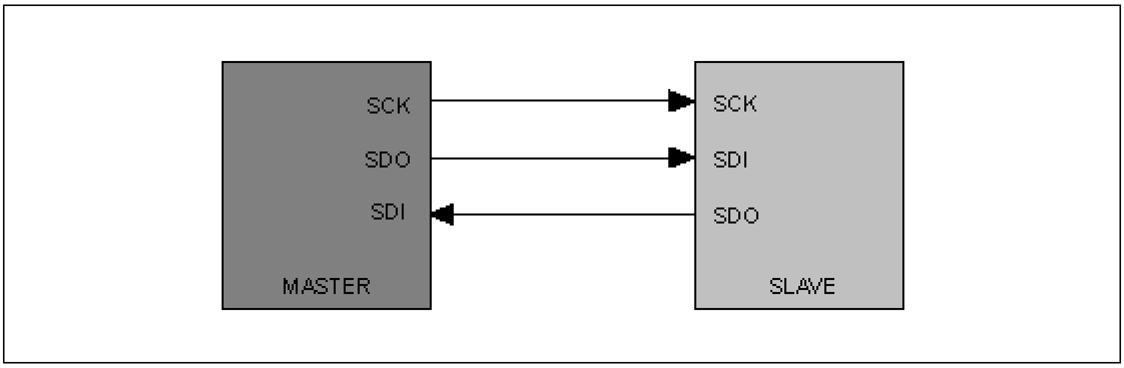


Figure 7-14: System diagram of SPI between two devices

7.6.1 Master Mode

In master mode, clock (SCK) is generated from internal system clock by setting SM bit to 0. SCK pin (P1.7) must be configured as an output pin for master mode. The transmit clock frequency is determined by the value of SPIBAUD. IDS bit defines the logic level of SCK in idle state. When IDS=0, logic level of SCK will be 0 when data is not transmitting. SMP bit determines whether data is transmitted or sampled at rising or falling edge of the clock. Selection of rising of falling edge of the clock depends on IDS bit. When IDS and SMP have the same value (IDS=SMP=0 or IDS=SMP=1), falling edge is selected for transmission or reception. On the other hand, if IDS and SMP are have different values (IDS=0, SMP=1; or IDS=1, SMP=0), rising edge is selected. Figure 7-15 SPI waveform shows the relationship between SMP, IDS, and the SPI output waveform when IDS=0 and IDS=1.

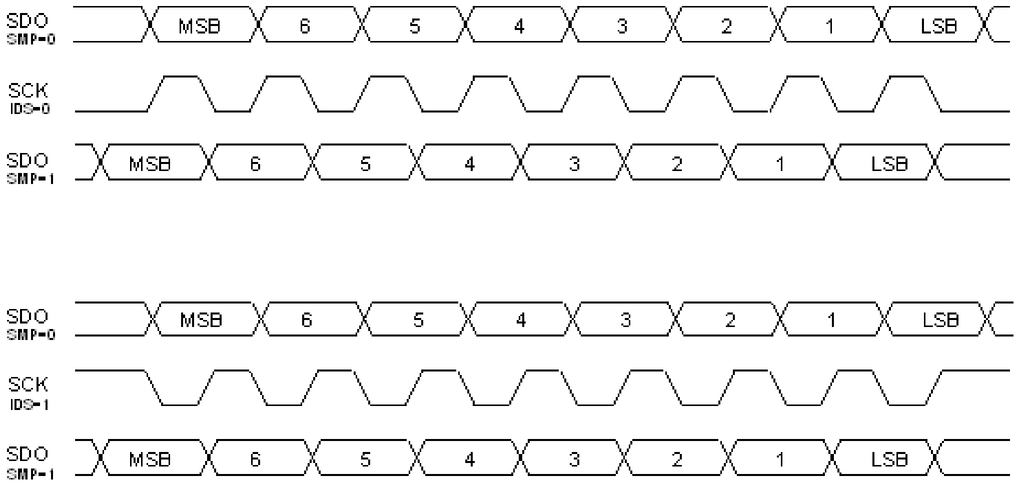


Figure 7-15 SPI waveform

7.6.2 Slave Mode

Slave mode operation is the same as master code, except that the input clock signal SCK is from an external source. Therefore, SCK pin (P1.7) must be configured as an input pin for slave mode. It is highly recommended that SCK frequency is 1/8 of CFP5102 system clock frequency or slower to ensure proper operation.

7.6.3 Two-wire Mode

Typical SPI connection requires three wires, with separate data input and output lines. SPI also supports 2-wire mode, which uses a single wire for bi-directional data operation. Two-wire mode can be selected by setting WS bit (bit 4 of SPICON) to 1. RTS bit controls whether data is transmitting or receiving in 2-wire mode. During data transmission, set RTS to 0 and SD (P2.0) as output pin. For data reception, set RTS to 1 and SD as input pin. RTS status may be disregarded when operating in 3-wire mode. Figure 7-16 SPI two-wire connection illustrates the SPI 2-wire mode connection.

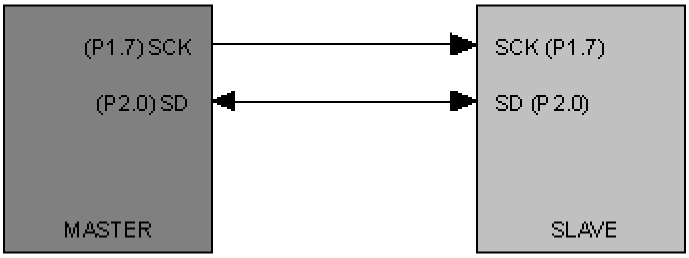


Figure 7-16 SPI two-wire connection

7.6.4 Receive-only Mode

When SPI is used for receiving data only, receive-only can be enabled by setting RXO (bit 3 of SPICON) to 1. Receive-only mode allows fewer instructions to be run by software when data is received continuously. During normal data reception, data must be written to SPIBUF to initiate reception. In receive-only mode, data reception can be initiated once only. After the first time, data reception will start automatically again once data is loaded from receive buffer by reading SPIBUF. Pending bit (SPIPND) does not need to be cleared by software nor by writing data to SPIBUF to initiate reception. Receive-only mode is applicable to any combination of master/slave and 2/3-wire modes.

Caution:

(1) In receive-only mode, the data in the transmit data buffer is also output in data reception. To define the level of the data output wire, user has to initial the transmit data buffer to output desired level. For example, to keep the output wire high, write 0xFF to SPIBUF before data reception.

SPICON: **SPI Control Register**
TYPE: **R/W**
ADDRESS: **0x95**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SPIPND	SM	RTS	WS	RXO	SMP	IDS	SPIEN
Default	0	0	0	0	0	0	0	0

<table>
<tr><th>Name</th><th>Mode</th><th>Description</th><th>Setting</th></tr>
<tr><td>SPIPND</td><td>RW</td><td>SPI interrupt pending flag
Please refer to Section 7.6.5 for details</td><td>0: inactive
1: Interrupt pending</td></tr>
<tr><td>SM</td><td>RW</td><td>Master mode or slave mode select bit</td><td>0: master mode
1: slave mode</td></tr>
<tr><td>RTS</td><td>RW</td><td>Receive / Transmit select bit in 2-wire mode
Status of RTS is disregarded for 3-wire mode</td><td>0: transmit
1: receive</td></tr>
<tr><td>WS</td><td>RW</td><td>2-wire mode/3-wire mode select bit
SPI uses SD (P2.0) only for data input and output in 2-wire mode</td><td>0: 3-wire mode
1: 2-wire mode</td></tr>
<tr><td>RXO</td><td>RW</td><td>Receive-only mode enable bit
Reduce number of operations for receive data in this mode.</td><td>0: disable
1: enable</td></tr>
<tr><td>SMP</td><td>RW</td><td>SPI sampling edge select bit
Define the sampling edge of the SPI clock</td><td><table><tr><td>IDS</td><td>SMP = 0</td><td>SMP = 1</td></tr><tr><td>0</td><td>Falling edge</td><td>Rising edge</td></tr><tr><td>1</td><td>Rising edge</td><td>Falling edge</td></tr></table></td></tr>
<tr><td>IDS</td><td>RW</td><td>SPI clock signal idle state
Define the logic level of clock signal when idle</td><td>0: clock signal stay at 0
1: clock signal stay at 1</td></tr>
<tr><td>SPIEN</td><td>RW</td><td>SPI enable bit</td><td>0: disable
1: enable</td></tr>
</table>

SPIBUF: **SPI Data Buffer Register**
TYPE: **R/W**
ADDRESS: **0x97**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SPIBUF							
Default	0x00							

7.6.5 Baudrate Setting

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

$$Baudrate = \frac{f_{system\_clock}}{2\times(SPIBAUD+1)}$$

SPIBAUD: **SPI Baud Control Register**
TYPE: **R/W**
ADDRESS: **0x96**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SPIBAUD							
Default	0x00							

Description	Setting
SPIBAUD register defines the baud rate of the SPI. This is effective in master mode only. The effective output clock frequency is equal to *system clock frequency / [2(SPIBAUD+1)]*.	

7.6.6 Interrupt

SPI asserts an interrupt when ESI is enabled and SPI pending bit (SPIPND, bit 7 of SPICON) is set to 1. SPI sets pending bit to 1 when one byte of data is received or transmitted. SPIPND can be cleared by one of the following actions:

1. Write transmit data to SPIBUF
2. Read received data from SPIBUF in receive-only mode (RXO=1)
3. Write 0 to SPIPND bit

IEN1: Interrupt Enable Register 1
TYPE: R/W
ADDRESS: 0xA9

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic							ESI	
Default							0	

Name	Mode	Description	Setting
ESI	RW	Serial protocol interface interrupt enable bit	0: disable 1: enable

SPICON: SPI Control Register
TYPE: R/W
ADDRESS: 0x95

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	SPIPND							
Default	0							

Name	Mode	Description	Setting
SPIPND	RW	SPI interrupt pending flag. SPIPND is set to 1 when one byte of data is transmitted or received. It can be cleared by: (1) write transmit data to SPIBUF; (2) read received data from SPIBUF in receive-only mode; (3) write '0' to this bit	0: inactive 1: Interrupt pending

7.6.7 Operation Flow

Receive and transmit data in normal mode

1. Configure P1.7, P2.0, and P2.1 input/output direction
2. Configure baud rate (SPIBAUD)
3. Configure operation mode (IDS and SMP)
4. Disable Receive-only mode
5. Clear pending flag
6. Set SPI interrupt enable bit in IEN1 if necessary
7. Enable the SPI
8. Write transmit data to SPIBUF
9. When pending bit is set to 1, read received data from SPIBUF
10. Repeat step 8

Receive data in Receive-only mode

1. Configure P1.7, P2.0, and P2.1 input/output direction
2. Configure baud rate (SPIBAUD)
3. Configure operation mode (IDS and SMP)
4. Enable receive-only mode (set RXO to 1)
5. Clear pending flag
6. Set SPI interrupt enable bit in IEN1 if necessary
7. Enable the SPI
8. Write dummy data to SPIBUF for the first time (content will not be changed after data is received)
9. When pending bit is set to 1, read received data from SPIBUF
10. Repeat step 9

7.7 General Purpose Serial Interface (GPSI)

GPSI is a flexible, bi-directional serial interface which provides a straight forward connection to a communication controller through synchronous serial data stream for transmit and receive data. GPSI module supports the following features:

- Master or slave mode
- Four different transmit modes: GPSI, SPI-like, USART-like, and UART-like
- 16-bit or 32-bit header function
- Adjustable transmit clock baud rate divider

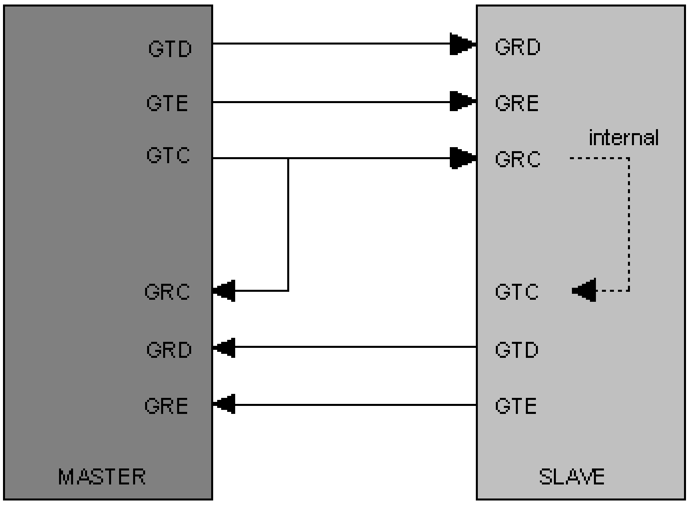


Figure 7-17: Typical GPSI master and slave mode connection

Figure 7-17 shows the typical master and slave mode connection. GPSI uses 6 pins:

Transmit clock (P1.1) – GTC

Transmit enable (P1.2) – GTE

Transmit data (P1.3) – GTD

Receive data (P1.4) – GRD

Receive clock (P1.5) – GRC

Receive enable (P1.6) – GRE

In master mode, transmit clock (GTC) is generated from internal system clock by setting TXCLKSEL bit to 0. The transmit clock frequency is determined by the value of GPSIBAUD. In slave mode, GPSI uses receive clock (GRC) as transmit clock by setting TXCLKSEL to 1. In this case, transmit clock bypasses internal baud rate divider, so value of GPSIBAUD is disregarded. TXEDSEL and RXEDSEL can be used for selecting either rising edge or falling edge of its clock during data transmission and reception.

7.7.1 Header

GPSI supports 16-bit and 32-bit header. Header function can be enabled by setting TXHDREN or RXHDREN when transmitting or receiving data. HEAD32 defines whether header length is 16 or 32 bits long. When transmit header is enabled by setting TXHDREN to 1, a header, defined in GPSIHDR, will be transmitted, immediately followed by the data stored in GPSIBUF. Once enabled, TXHDREN HDRTXLOAD flag will be set to 1 once the header is transmitted. HDRTXLOAD flag can be cleared by writing 0 to HDRTXCLR bit. Writing 1 to HDRTXCLR bit will have no effect on HDRTXLOAD flag. The following figures illustrate the waveform for data transmission without header (Figure 7-18), 16-bit header (Figure 7-19), and 32-bit header (Figure 7-20).

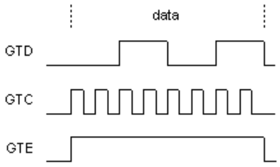


Figure 7-18 GPSI waveform without header

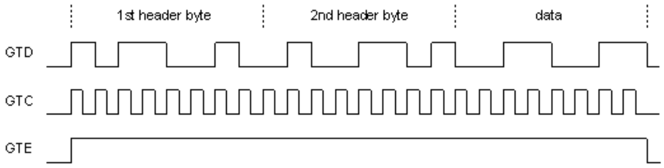


Figure 7-19 GPSI waveform with 16-bit header

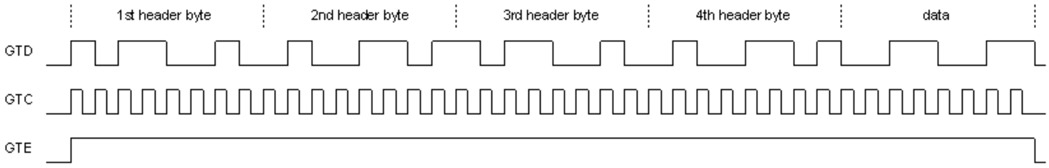


Figure 7-20 GPSI waveform with 32-bit header

For 16-bit header, GPSI transmits the content of GPSIHDR for the first byte in header. The second byte is the inverted content of GPSIHDR. As an example illustrated in Figure 7-19, if value of GPSIHDR is 0x4D, the first header byte to be transmitted is 0x4D and the second header byte is 0xB2, followed by the data stored in GPSIBUF (0xCC). For 32-bit header (Figure 7-20 GPSI waveform with 32-bit header), four bytes of header is transmitted in the following order: 0x4D, 0xB2, 0x4D, 0xB2, where the first and third byte is the content of GPSIHDR, and the second and fourth byte is the inverted content of GPSIHDR.

If receive header is enabled, GPSI expects a header before the incoming data and checks the received header with the value stored in GPSIHDR. If the received header matches with the content in GPSIHDR, HDRRXMATCH flag will be set to 1. Using the above example, the content of GPSIHDR is 0x4D. If the received header has 0x4D for the first byte (i.e. content of GPSIHDR) and 0xB2 for the second byte (i.e. the inverted content of the GPSIHDR), the receive header matches. For 32-bit header, GPSI checks for the following 4 bytes: 0x4D, 0xB2, 0x4D, 0xB2 (i.e. the first and third byte is the content of GPSIHDR, the second and fourth byte is the inverted content of GPSIHDR). HDRRXMATCH flag can be cleared by writing 0 to HDRRXCLR bit. When HDRTXCLR bit is set to 1, it will have no effect on HDRRXMATCH flag.

GPSIHDR: **GPSI Header Register**
TYPE: **R/W**
ADDRESS: **0xC3**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	GPSIHDR							
Default	0x00							

7.7.2 Transmit Protocol

GPSI module supports 4 different transmit modes: GPSI, SPI-like, USART-like, and UART-like modes.

GPSI mode

When data is not transmitted, GTD remain logic 0, but GTC is running at all times. GTE is logic 1 for the duration of data transmission. This is the standard GPSI interface.

SPI-like mode

GTD and GTC remain logic 0 when data is not transmitted. The state of GTE does not matter and GPSI does not occupy GTE (P1.2).

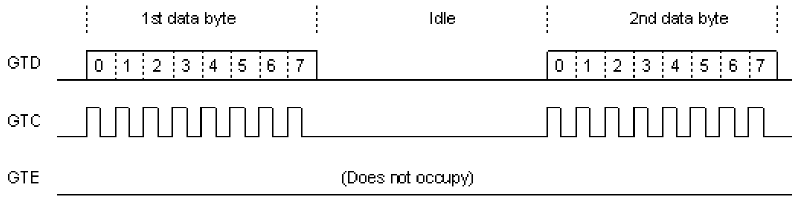


Figure 7-21 GPSI SPI-like mode

USART-like mode

When data is not transmitted, GTD remains logic 0, but GTC is running at all times. The state of GTE does not matter and GPSI does not occupy GTE (P1.2).

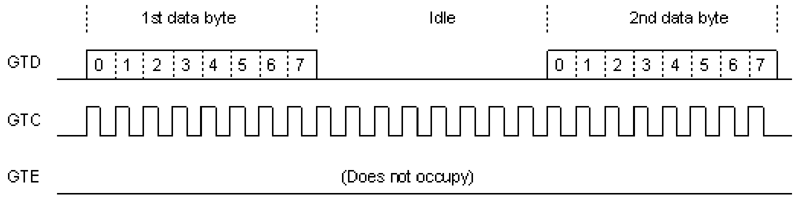


Figure 7-22 GPSI USART-like mode

UART-like mode

When data is not transmitted, GTD remains logic 0. The state of GTE and GTC does not matter and GPSI does not occupy GTE (P1.2) and GTC (P1.1). Note that GPSIBAUD still controls the data rate even when GTC is not in use.

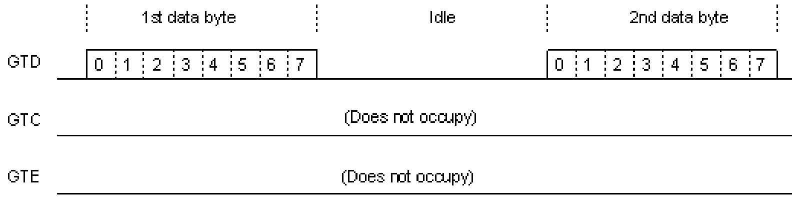


Figure 7-23 GPSI UART-like mode

Caution:

(1) Note that the least significant bit of each byte is transmitted / received first.

7.7.3 Receive Protocol

When RXENOFF (bit 2 of GPSICON1) is set to 1, GRE pin (P1.6) is not used by GPSI module. Internally, GPSI always has RX enabled. In this case, EOP will not be detected and GPSI does not occupy P1.6. When RXENOFF is set to 0, GRE pin will indicate if valid data is available for reception. EOP flag will be ineffective when RXENOFF is set to 1.

GPSICON0: GPSI Control Register 0
TYPE: R/W
ADDRESS: 0xC0

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	GPSI RXIE	GPSI TXIE	TXCLK SEL	RXED SEL	TXED SEL	GPSIEN
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
GPSIRXIE	RW	GPSI RX interrupt enable bit If enabled, GPSI triggers an interrupt if RXBUFFULL is set to 1.	0: Disable RX interrupt 1: Enable RX interrupt
GPSITXIE	RW	GPSI TX interrupt enable bit If enabled, GPSI triggers an interrupt if TXBUFEPTY is set to 1.	0: Disable TX interrupt 1: Enable TX interrupt
TXCLKSEL	RW	GPSI transmit clock source select bit When TXCLKSEL=0, TX clock uses system clock and GPSIBAUD determines the actual clock frequency. When TXCLKSEL=1, TX clock uses RX clock any bypasses internal baud rate divider.	0: Select system clock 1: Select RX clock
RXEDSEL	RW	GPSI receive clock edge select bit When RXEDSEL=0, GPSI receives data at RX clock rising edge. When RXEDSEL=1, GPSI receives data at RX clock rising edge.	1: Rising edge 0: Falling edge
TXEDSEL	RW	GPSI transmit clock edge select bit When TXEDSEL=0, GPSI transmits data at TX clock rising edge. When TXEDSEL=1, GPSI transmits data at TX clock rising edge.	1: Rising edge 0: Falling edge
GPSIEN	RW	GPSI enable bit	0: Disable GPSI 1: Enable GPSI

GPSICON1: GPSI Control Register 1
TYPE: R/W
ADDRESS: 0xC1

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	RXBUF FULL	TXBUF EPTY	RX HDREN	TX HDREN	EOP	RXEN OFF	TXMODE [1]	TXMODE [0]
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
RXBUFFULL	RW	RX buffer full flag This flag will be set to 1 when RX buffer is full. It must be cleared by software.	0: RX buffer is not full 1: RX buffer is full
TXBUFEPTY	R	TX buffer empty flag This flag will be set to 1 when TX buffer is empty. It will be cleared to 0 when data is written into GPSIBUF.	0: TX buffer is not empty 1: TX buffer is empty
RXHDREN	W	RX header enable bit Once enabled, it cannot be disabled by software. It will be disabled automatically after a header is received.	0: Disable RX header 1: Enable RX header
TXHDREN	W	TX header enable bit Once enabled, it cannot be disabled by software. It will be disabled automatically after a header is transmitted.	0: Disable TX header 1: Enable TX header
EOP	RW	End of packet (EOP) flag EOP flag will be set to 1 when 1-to-0 transition is detected on GRE pin (P1.6). It must be cleared by software.	0: EOP is not detected 1: EOP is detected
RXENOFF	RW	RX enable off bit When RXENOFF=1, GRE pin (P1.6) is not used by GPSI. Internally, GPSI always has RX enabled. In this case, EOP will not be detected	0: GRE pin is enabled 1: GRE pin is disabled

		and GPSI does not occupy P1.6.	
TXMODE[1:0]	RW	TX mode select	00: GPSI mode 01: SPI-like mode 10: USART-like mode 11: UART-like mode

GPSICON2: **GPSI Control Register 2**
TYPE: **R/W**
ADDRESS: **0xC2**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	HDR RXCLR	HDR TXCLR	GPSI EOPIE	GPSI RXHIE	GPSI TXHIE	HEAD 32	HDRRX MATCH	HDRTX LOAD
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
HDRRXCLR	W	RX header matched flag clear bit HDRRXMATCH can be cleared by writing 0 to HDRRXCLR. When HDRRXCLR is set to 1, it has no effect.	0: Clear RX header matched flag 1: Inactive
HDRTXCLR	W	TX header loaded flag clear bit HDRTXLOAD can be cleared by writing 0 to HDRTXCLR. When HDRTXCLR is set to 1, it has no effect.	0: Clear TX header loaded flag 1: Inactive
GPSIEOPIE	RW	EOP interrupt enable bit If enabled, GPSI triggers an interrupt when EOP is set to 1.	0: Disable EOP interrupt 1: Enable EOP interrupt
GPSIRXHIE	RW	RX header interrupt enable bit If enabled, GPSI triggers an interrupt when HDRRXMATCH is set to 1.	0: Disable RX header interrupt 1: Enable RX header interrupt
GPSITXHIE	RW	TX header interrupt enable bit If enabled, GPSI triggers an interrupt when HDRTXLOAD is set to 1.	0: Disable TX header interrupt 1: Enable TX header interrupt
HEAD32	RW	32-bit header enable bit When HEAD32 is set to 1, header length is defined to be 32 bits.	0: Header length is 16-bit 1: Header length is 32-bit
HDRRXMATCH	R	RX header matched flag This flag will be set to 1 when RX header received is matched with the content of GPSIHDR. It can only be cleared by writing 0 to HDRRXCLR.	0: RX header is not matched 1: RX header is matched
HDRTXLOAD	R	TX header loaded flag This flag will be set to 1 when TX header is transmitted. It can only be cleared by writing 0 to HDRTXCLR.	0: TX header is not transmitted 1: TX header is transmitted

GPSIBUF: **GPSI Data Buffer Register**
TYPE: **R / W**
ADDRESS: **0xC4**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	GPSIBUF							
Default	0x00							

7.7.4 Interrupts

GPSI has 5 different interrupt sources. When GPSI master interrupt (EGPSI, bit 5 of IE1) is enabled, one of the following GPSI interrupts will be triggered:

TX interrupt (GPSITXIE)

When GPSITXIE is set to 1, GPSI triggers TX interrupt when TXBUFEPTY flag is set to 1. If data in GPSIBUF is loaded to the internal shift register for transmission, GPSIBUF becomes empty and TXBUFEPTY flag is set to 1. It indicates that GPSIBUF is ready for more data. TXBUFEPTY flag will be cleared automatically when data is written to GPSIBUF.

TX header interrupt (GPSITXHIE)

When GPSITXHIE is set to 1, GPSI triggers TX header interrupt when HDRTXLOAD flag is set to 1. In 16-bit header case, GPSI sets HDRTXLOAD to 1 after the first header byte is transmitted. In 32-bit header case, GPSI sets HDRTXLOAD to 1 after the third header byte is transmitted. It indicates that TX header is just about finish transmission. HDRTXLOAD flag can be cleared only by writing 0 to HDRTXCLR (bit 6 of GPSICON2).

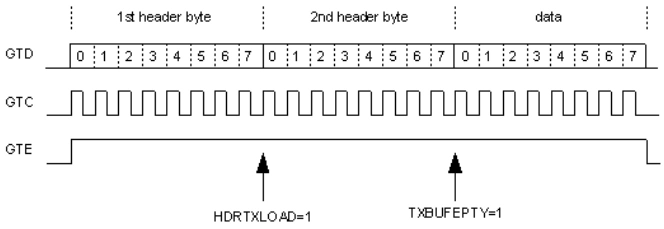


Figure 7-24 GPSI 16-bit header TX interrupt timing

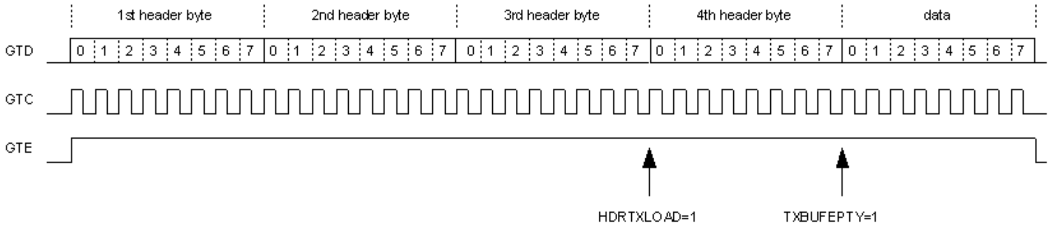


Figure 7-25 GPSI 32-bit header TX interrupt timing

RX interrupt (GPSIRXIE)

When GPSIRXIE is set to 1, GPSI triggers RX interrupt when RXBUFFULL flag is set to 1. Once the last data bit is received and stored in GPSIBUF, the receive buffer becomes full and GPSI sets RXBUFFULL flag to 1. It indicates that receive buffer contains valid data. RXBUFFULL flag can be cleared by software.

RX header interrupt (GPSIRXHIE)

When GPSIRXHIE is set to 1, GPSI triggers RX header interrupt when HDRRXMATCH flag is set to 1. For 16-bit header, the first byte must match with the content of GPSIHDR and the second byte must match with the inverted content of GPSIHDR. For 32-bit header, the first and third byte must match with the content of GPSIHDR and the second and fourth byte must match with the inverted content of GPSIHDR. When the received header matches with the defined header pattern, HDRRXMATCH flag will be set to 1. HDRRXMATCH can be cleared only by writing 0 to HDRRXCLR (bit 7 of GPSICON2).

RX EOP interrupt (GPSIEOPIE)

When GPSIEOPIE is set to 1, GPSI triggers RX EOP (end of packet) interrupt when EOP flag is set to 1 (bit 3 of GPSICON1). If a 1-to-0 transition is detected on receive enable pin (GRE), GPSI sets EOP flag to 1. EOP flag can be cleared by software. GPSIEOPIE becomes ineffective when RXENOFF is set to 1.

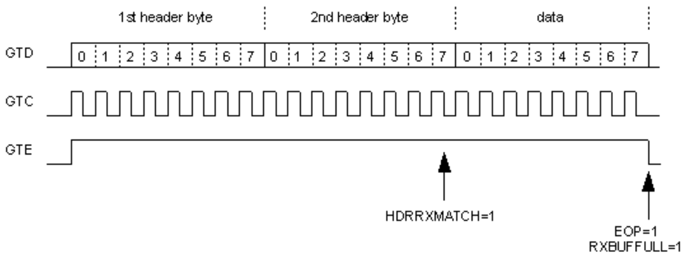


Figure 7-26 GPSI 16-bit header RX interrupt timing

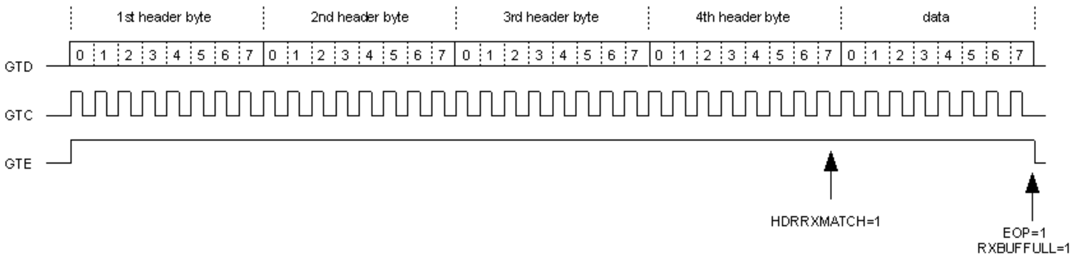


Figure 7-27 GPSI 32-bit header RX interrupt timing

IEN1: **Interrupt Enable Register 1**
TYPE: **R / W**
ADDRESS: **0xA9**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic			EGPSI					

Default			**0**					

Name	Mode	Description	Setting
EGPSI	RW	GPSI interrupt enable bit	0: disable 1: enable

7.7.5 Baudrate Setting

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

$$Baudrate = \frac{f_{system\_clock}}{2 \times (GPSIBAUD + 1)}$$

GPSIBAUD: GPSI Baud Rate Divider Register
TYPE: R / W
ADDRESS: 0xC5

Position	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Mnemonic	**GPSIBAUD**							
Default	**0x00**							

Description	Setting
GPSIBAUD register defines the baud rate of the GPSI. This is effective in master mode only. The effective output clock frequency is equal to *system clock frequency / [8(GPSIBAUD+1)]*.	

7.7.6 Operation Flow

TX Operation Flow

1. Set P1.1, P1.2, P1.3 as output, and P1.4, P1.5, P1.6 as input
2. Select master or slave mode
3. Select TX mode and TX clock edge
4. Enable TXHDREN and HEAD32 if necessary
5. Write to GPSIHDR if TXHDREN is enabled
6. Enable GPSITXIE and GPSITXHIE if necessary
7. Set GPSIEN to 1
8. Write to GPSIBUF to start transmission

RX Operation Flow

1. Set P1.1, P1.2, P1.3 as output, and P1.4, P1.5, P1.6 as input
2. Select master or slave mode
3. Select RX clock edge
4. Enable RXHDREN and HEAD32 if necessary
5. Write to GPSIHDR if RXHDREN is enabled
6. Enable GPSIRXIE, GPSIRXHIE, and GPSIEOPIE if necessary
7. Set GPSIEN to 1 to begin reception

7.8 Analog-to-Digital Converter (ADC)

CFP5102 provides an eight-channel moderate conversion speed and a moderate resolution 8-bit successive approximated register Analog-to-Digital Converter (ADC) for users to develop applications in Measurement requiring moderate performance and speed

ADC has an 8-to-1 analog MUX channel for selecting one of the input channels to ADC circuitry. **Channels 0 to 7** are multiplexed with GPIO pin: **P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7 and P1.2**. To use either one of the channels, the GPIO has to be configured as analog input (please refer to Section 7.1) and the conversion results are stored in ADCB register. Control register ADCCON0 and ADCCON1 registers control the operation of the ADC and the ADCSC register controls the ADC sampling clock frequency for controlling the speed of conversion.

7.8.1 Reference Voltage

For the sake of flexibility, the ADC reference voltage can be selected to use either DVDD or REFO. The reference voltage defines the highest possible conversion voltage the ADC can obtain during conversion. Two options can be chosen. One is REFO which has been multiplexed to P0.0 and the other is DVDD which is the internal DVDD power supply, which is different to the external system power during LDO in use. **Precaution should be taken to check LDO output voltage to ensure reference voltage is properly required.**

Caution:

(1) Power supply voltage (DVDD) and ADC external reference voltage (REFO) should be about 3.3V. High voltage to ADC may cause permanent physical damage.

7.8.2 Input Unity-Gain Buffer

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 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.

7.8.3 Conversion and Interrupt

For ADC conversion, a general procedure can be like in the following:

1. Configure ADCCON0 register by

   i. clear the ADC done pending bit

   ii. select the ADC trigger mode either manual or timer-trigger

   iii. select whether to use input buffer (make sure the input impedance from the signal is not too high)

   iv. select the reference voltage

2. Configure ADCCON1 register by

i. select the reference voltage buffer if necessary

ii. select the channel to use

3. Select the ADC clock frequency by configuring ADCSC register
4. Kick-start the ADC conversion by selecting timer 0/1/2 under timer-trigger mode or setting ADCGO under manual mode
5. During conversion, ADCGO bit keeps '1', and after each conversion, ADCGO bit will change to 0. In addition, ADC done pending bit will change to 1 after conversion, and interrupt pending will be asserted.
6. The conversion result is captured in register ADCB. The value retains until the next completion of conversion

ADCCON0: ADC Control Register 0
TYPE: R/W
ADDRESS: 0xF4

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	ADCPND	INBUF	REFS	TM	TC[1]	TC[0]	GO	ADCEN
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
ADCPND	RW	ADC interrupt pending flag. Interrupt source. Reset to '0' when write a '0' to this bit	0: inactive 1: Interrupt pending
INBUF	RW	Input buffer enable bit	0: disable 1: enable
REFS	RW	Reference voltage selection bit	0: VDDCORE 1: P00
TM	RW	Trigger mode select bit	0: manual kick start 1: automatic trigger
TC	RW	Trigger source selection bit Select a source to trigger the conversion automatically	00: timer 0 overflow 01: timer 1 overflow 10: timer 2 overflow
GO	RW	Conversion kick start bit Write '1' to kick start conversion. This bit is set to '1' once the conversion begin and is cleared when the conversion is done	0: idle 1: start
ADCEN	RW	ADC enable bit	0: disable 1: enable

ADCCON1: ADC Control Register 1
TYPE: R/W
ADDRESS: 0xF5

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	CM	GT	-	REFBE	-	AS[2]	AS[1]	AS[0]
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
CM	RW	Compare mode enable bit.	0: disable 1: enable
GT	RW	Interrupt condition selection bit in compare mode	0: less and equal to threshold 1: greater than threshold
REFBE	RW	Reference voltage buffer enable bit	0: disable 1: enable
AS	RW	Channel selection for group A	000: P01 001: P02 010: P03 011: P04 100: P05

			101: P06 110: P07 111: P12

ADCB: **ADC Buffer Register**
TYPE: **R**
ADDRESS: **0xEE**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	ADCB							
Default	0xXX							

IEN1: **Interrupt Enable Register 1**
TYPE: **R/W**
ADDRESS: **0xA9**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic				EADC				
Default				0				

Name	Mode	Description	Setting
EADC	RW	ADC interrupt enable bit	0: disable 1: enable

7.8.4 Comparison Mode

To facilitate real-time monitoring, ADC supports automatically comparing the measured result and the defined threshold. ADC keeps converting until the defined condition is fulfilled. This operation mode is called comparison mode. This is useful, for example during monitoring the voltage output of batteries. User can monitor the battery life by setting up a detection threshold at ADC threshold register ADCT, and then configuring the ADC to interrupt when the measured result is lower than the threshold. As the operation is automatic, CPU does not involve in decision making, thus ensuring the degree of real-time.

To operate in comparison mode, the procedure is as the following:

1. Configure ADCCON0 register by
   i. clear the ADC done pending bit
   ii. select the ADC manual trigger mode
   iii. select whether to use input buffer (make sure the input impedance from the signal is not too high)
   iv. select the reference voltage
2. Configure ADCCON1 register by
   i. enable comparison mode
   ii. select interrupt condition
   iii. select the reference voltage buffer if necessary
   iv. select the channel to use
3. Select the ADC clock frequency by configuring ADCSC register
4. Kick-start the ADC conversion by setting ADCGO under manual mode
5. Interrupt pending asserts when the interrupt condition is fulfilled.

6. The conversion result is captured in register ADCB. The value retains until the next completion of conversion
7. To stop during conversion,

   i. disable ADC

   ii. disable ADC interrupt

   iii. Invert control bit GT in ADCCON1 to force to fulfil the interrupt condition

ADCT: **ADC Threshold Register**
TYPE: **W**
ADDRESS: **0xF1**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	ADCT							
Default	0xXX							

7.8.5 Conversion Time and Sampling Clock

A conversion in 8-bit resolution takes 13.5 ADC sampling cycles to complete. The ADC conversion speed depends not only on resolution mode but also on ADC clock frequency. The ADC clock source is generated from system clock and is divided by a separate 8-bit ADC clock divider to give out a flexible ADC sampling clock frequency range so that user can decide the actual ADC input clock and the final conversion speed based on different applications.

ADCSC register is used to control the clock-divide ratio of the ADC clock divider. The effective sampling clock frequency is formulated as

$$F_{ADC\_CLOCK} = F_{SYSTEM\_CLOCK} / 2(ADCSR+1)$$

For example, F_{SYSTEM_CLOCK} = 48MHz, ADCSC=0x0B, then F_{ADC_CLOCK} =2MHz (the recommended frequency)

Note:

Maximum allowable ADC clock frequency is 2MHz and frequency higher than 2MHz is not recommended.

ADCSC: **ADC Sampling Clock Control Register**
TYPE: **W**
ADDRESS: **0xF3**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	ADCSR							
Default	0x00							

7.9 Universal Serial Port (USB)

7.9.1 USB Device

CFP5102 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/un-stuffing, 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 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.

Endpoints

CFP5102 supports three endpoints, EP0 for control, EP1 endpoint (IN/OUT) and EP2 endpoint (IN/OUT) for data transmission. Every endpoint supports three modes: Interrupt mode, Bulk mode and Isochronous mode. All endpoints share a 208 bytes FIFO buffer. The buffer sizes of the endpoints are summarized in Table 7-3.

Table 7-3: Endpoint buffer size

Endpoint	Input Buffer	Output Buffer
EP0	16 bytes (shared)	
EP1	64 bytes	64 bytes
EP2	32 bytes	32 bytes

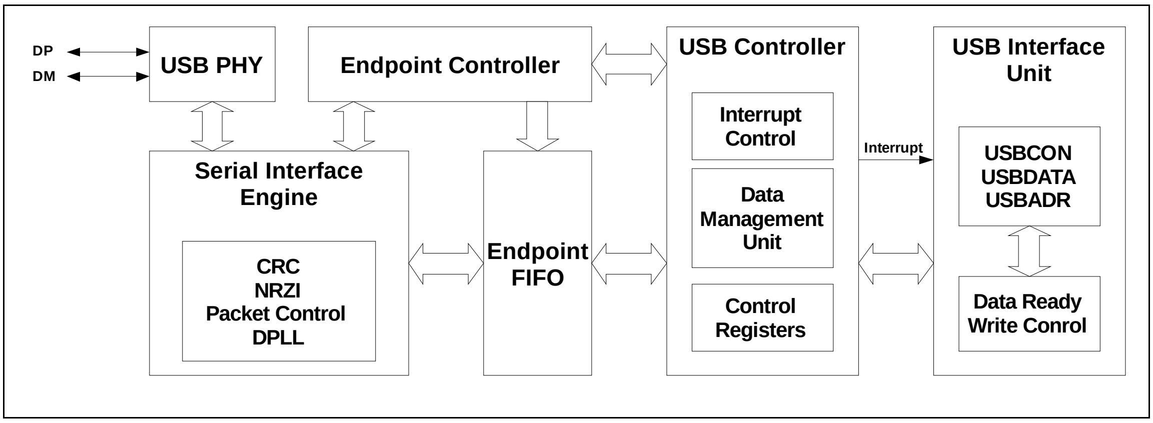


Figure 7-28: System architecture of USB module

7.9.2 USB Interface Unit

CPU controls the USB module through the registers defined in USB interface unit. The registers are assigned to dedicated address in SFR, shown in Table 7-4. All functions of the USB module are accessed through operating USBCON0, USBADR and USBDATA. USBCON1 is used to control the physical layer. The rest are for direct memory access (DMA) control. The detailed description of USBCON0 and USBCON1 are tabulated below.

Table 7-4: Registers in USB interface unit

Mnemonic	Address	Type	Description
USBCON0	0xF8	RW	USB control register 0
USBCON1	0xF9	RW	USB control register 1
USBDATA	0xFA	RW	USB access data register
USBADR	0xFB	RW	USB access address register
USBDADR	0xFC	RW	USB DMA base address register
USBDM	0xFD	RW	USB DMA interrupt mask register
USBDR	0xFE	RW	USB DMA request register
USBDD	0xFF	RW	USB DMA done register

USBCON0: USB Control Register 0
TYPE: R/W
ADDRESS: 0xF8

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	UCD	URST	-	-	URW	UDS	UCS
Default	0	1	0	0	0	X	0	0

Name	Mode	Description	Setting
UCD	RW	USB controller done flag Interrupt source. Set to ‘1’ when USB controller completes the operation (not including DMA). Reset to ‘0’ when write ‘0’ to this bit. *Caution: when DMA is operating, USB controller interrupt is blocked.*	0: inactive 1: Interrupt pending
URST	WO	USB controller reset bit This bit resets during power-on reset and master reset only	0: force to reset state 1: release from reset
URW	RW	Read/Write select bit. Select the access mode to USB controller. It also defines the direction of DMA channel	0: write 1: read
UDS	WO	USB DMA transfer kick-start bit Write ‘1’ to kick start DMA transfer	
UCS	WO	USB interface kick-start bit Write ‘1’ to kick start manipulation to USB controller	

IEN1: Interrupt Enable Register 1
TYPE: R/W
ADDRESS: 0xA9

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic					EUD	EUI		
Default					0	0		

Name	Mode	Description	Setting
EUD	RW	USB DMA interrupt enable bit	0: disable 1: enable
EUI	RW	USB interface interrupt enable bit	0: disable 1: enable

USBCON1: **USB Control Register 1**
TYPE: **R / W**
ADDRESS: **0xF9**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	UDMU	UDPU	UPE	USPD
Default	0	0	0	0	0	0	0	1

Name	Mode	Description	Setting
UDMU	WO	Data pin DM pull-up control	0: pull-up disable 1: pull-up enable
UDPU	RW	Data pin DP pull-up control	0: pull-up disable 1: pull-up enable
UPE	RW	USB physical layer enable bit	0: disable 1: enable
USPD	RW	USB physical layer speed select bit	0: low speed 1: high speed

7.9.3 Access USB Controller

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

Write operations

To perform a write operation, the address and data are stored to USBADDR and USBDATA registers, respectively. For a write operation, clear URW bit to ‘0’. Write operation to most of the registers takes 2 USB clock cycles. Writing to CSR0 takes 7 USB clock cycles while writing to other CSR registers (InCSR1, OutCSR1, InCSR2, and OutCSR2) takes 6 USB clock cycles. The write operation will begin in background once it is kick-started by setting UCS to ‘1’. UCS is write-only bit. Reading it will get a ‘0’. When write operation is kick-started, UCD will be cleared. User can poll the UCD bit and it will change to ‘1’ to indicate that write operation is complete.

Read operations

Reading data from USB controller is similar to writing data to it. Store the register address to USBADDR from which data is going to be read. For a read operation, set URW to ‘1’. Then, set UCS to ‘1’ to start the read operation. The data of the register, pointed by USBADDR, will be stored in USBDATA when the read operation is complete. When read operation is kick-started, UCD will be cleared. User can poll the UCD bit and it will change to ‘1’ when read operation is complete. Reading from most registers takes 2 USB clock cycles and reading from FIFO registers takes 5 USB clock cycles.

Controlling USB controller

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

Common USB control registers (addresses 0x00 to 0x0F): These registers provide control and status information for the entire controller (see Table 7-9).

Indexed registers (addresses 0x10 to 0x1F): These registers provide control and status information for different endpoints (see Table 7-10). Endpoint can be selected by writing the endpoint number to the Index register (0x0E). 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.

FIFO registers (addresses 0x20 to 0x22): The FIFO registers for the IN endpoints appear as single bytes, consecutively in the memory map starting at address 20h (see Table 7-11). The FIFO registers for the OUT endpoints also appear consecutively at the same set of addresses. Writing address 0x21 results in loading a byte into the FIFO buffer for IN Endpoint 1. Reading from address 0x21 results in unloaded a byte from the FIFO buffer for OUT Endpoint 1.

DMA Operation: Endpoint1~2's IN/OUT DMA function is controlled by InCSR2/OutCSR2 registers. Reserving spaces in DATA or in XDATA is a must before doing DMA operations. The starting address of the memory block is set by the lower byte of USBDADR. The bit 6 of USBDADR selects the location: selects DATA when setting '0'; selects XDATA when setting '1'. The effective starting address is formulated as $ADDR_{DMA}$ = 8*USBDADR[4:0]. For example, when USBDADR is 0x27, the starting address is 0x38 in XDATA.

The size of the block should not be less than the size of the FIFO that prepares for DMA. "OUT DMA" transfers data from EndPoint's "OUT FIFO" to this block. "IN DMA" transfers data from the block to EndPoint's "IN FIFO". USBADR decides the EndPoint FIFO which undergoes DMA Operation.

USB's DMA requests can be read from USBDR. Interrupt will be generated once DMA request presents providing that DMA interrupt is enabled and the corresponding interrupt mask (USBDM) is cleared. Similarly, when USB DMA done, interrupt will also be generated if the corresponding interrupt mask is clear. USB DMA done status can be read from USBDD. The description of USBDR, USBDD and USBDM are shown below.

Setting URW means DMA operates Endpoint's "OUT FIFO", while clearing URW meaning DMA operates Endpoint's "IN FIFO". DMA will start automatically after setting UDS. Interrupt will be generated after DMA has finished if enable DMA Interrupt. The operating flow of "OUT DMA" is shown below. The difference between "OUT DMA" and "IN DMA" is that writing data to allocated memory block before setting UDS.

1. Poll until DMA request is active (or wait for DMA request interrupt)
2. Set the destination address through register USBDADR
3. Select the desired endpoint by setting USDADR and URW

4. Write '1' to UDS to kick start the DMA operation
5. Poll for the assertion of DMA done flag (or wait for DMA done interrupt)
6. Clear the DMA done flag and the data is ready for manipulation

USBDR: **USB DMA Request Register**
TYPE: **R/W**
ADDRESS: **0xFE**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	RI2	RI1	RO2	RO1
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
RI2	RW	USB Endpoint 2 IN DMA request	0: inactive 1: request occurs
RI1	RW	USB Endpoint 1 IN DMA request	0: inactive 1: request occurs
RO2	RW	USB Endpoint 2 OUT DMA request	0: inactive 1: request occurs
RO1	RW	USB Endpoint 1 OUT DMA request	0: inactive 1: request occurs

USBDD: **USB DMA Done Register**
TYPE: **R/W**
ADDRESS: **0xFF**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	DI2	DI1	DO2	DO1
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
DI2	RW	USB Endpoint 2 IN DMA done	0: inactive 1: DMA completes
DI1	RW	USB Endpoint 1 IN DMA done	0: inactive 1: DMA completes
DO2	RW	USB Endpoint 2 OUT DMA done	0: inactive 1: DMA completes
DO1	RW	USB Endpoint 1 OUT DMA done	0: inactive 1: DMA completes

USBDM: **USB DMA Interrupt Mask Register**
TYPE: **R/W**
ADDRESS: **FDh**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	-	-	-	-	MI2	MI1	MO2	MO1
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
MI2	RW	USB Endpoint 2 IN DMA interrupt mask	0: clear mask 1: set mask
MI1	RW	USB Endpoint 1 IN DMA interrupt mask	0: clear mask 1: set mask
MO2	RW	USB Endpoint 2 OUT DMA interrupt mask	0: clear mask 1: set mask
MO1	RW	USB Endpoint 1 OUT DMA interrupt mask	0: clear mask 1: set mask

Table 7-5: Common USB control registers

USBADR	Name	Description
0x00	FAddr	Function address register.
0x01	Power	Power management register.
0x02	IntrIn1	Interrupt register for Endpoint 0 plus IN Endpoints 1 & 2.
0x03	RESERVED	
0x04	IntrOut1	Interrupt register for OUT Endpoints 1 & 2.
0x05	RESERVED	
0x06	IntrUSB	Interrupt register for common USB interrupts.
0x07	IntrIn1E	Interrupt enable register for IntrIn1.
0x08	RESERVED	
0x09	IntrOut1E	Interrupt enable register for IntrOut1.
0x0A	RESERVED	
0x0B	IntrUSBE	Interrupt enable register for IntrUSB.
0x0C	Frame1	Frame number bits 0 to 7.
0x0D	Frame2	Frame number bits 8 to 10.
0x0E	Index	Index register for selecting the endpoint status and control registers.
0x0F	RESERVED	

Table 7-6: Indexed Endpoint control registers

USBADR	Name	Description
0x10	InMaxP	Maximum packet size for IN endpoint. (Endpoints 1 & 2)
0x11	CSR0	Control Status register for Endpoint 0. (Control Endpoint)
	InCSR1	Control Status register 1 for IN endpoint. (Endpoints 1 & 2)
0x12	InCSR2	Control Status register 2 for IN endpoint. (Endpoints 1 & 2)
0x13	OutMaxP	Maximum packet size for OUT endpoint. (Endpoints 1 & 2)
0x14	OutCSR1	Control Status register 1 for OUT endpoint. (Endpoints 1 & 2)
0x15	OutCSR2	Control Status register 2 for OUT endpoint. (Endpoints 1 & 2)
0x16	Count0	Number of received bytes in Endpoint 0 FIFO. (Control Endpoint)
	OutCount1	Number of bytes in OUT endpoint FIFO (lower byte). (Endpoints 1 & 2)
0x17	OutCount2	Number of bytes in OUT endpoint FIFO (upper byte). (Endpoints 1 & 2)

USBADR	Name	Description
0x20	FIFO0	FIFO register of EP0
0x21	FIFO1	FIFO register of EP1
0x22	FIFO2	FIFO register of EP2

Please refer to Appendix B for detailed description for the USB registers.

7.9.4 Wakeup Operations

CFP5102 provides a perfect management for power consumption. It reduces power consumption by banding SLEEP, HOLD and USB WAKEUP together closely. Table 7-7 shows every operating mode and requirement.

Table 7-7: USB operation mode

USB mode	CPU Clock	System Clock	USB Clock	USB PHY	Description
Normal Mode	√	√	√ 48MHz	√	Normal operation.
Keeping Mode	X IDLE	√	√ 48MHz	√	All USB Interrupt can wakeup CPU.
Suspend Mode	X IDLE	√	√	√	Only USB RESET/RESUME can wake up CPU. Make sure that USB clock is set back to 48MHz after wakeup sequence.
Sleep Mode	X Power Down	X	X	X	Only USB DP wakeup takes effect. CPU resets while USB keeps its status.

USB Normal Mode: CPU and USB Module run in full speed mode. OSC and PLL must be turned on.

USB Keeping Mode: CPU enters IDLE mode. OSC and DPLL keep turn-on in order to provide clock to USB module. USB still run normally and can send or receive data at this mode. If USB interrupt enable has been set, CPU will wakeup when USB interrupt happens. Moreover, USB can wake up only after USB leaving SUSPEND Mode.

USB Suspend Mode: USB can wakeup only after USB leaving SUSPEND Mode at this mode. CPU enters idle mode and DPLL is off, but system clock must keep running to provide clock to USB Module. USB will generate RESUME Interrupt to wakeup CPU if USB interrupt enable has been set and USB RESUME Interrupt has been enabled. And USB will also generate RESUME Interrupt to wakeup on condition that USBINTWE or USB RESUME has been close but SUSPEND has been turned on.

USB Sleep Mode: This mode will provide the maximal power saving. CPU enters power down Mode and system clock stops. Enable USBDP wakeup before entering Halt Mode. CPU wakes up after receiving signal from DP pin which turns according to appointed edge. CPU would be reset while USB Module doesn’t.

7.10 FIFO Interface

CFP5102 provides a FIFO interface to access USB FIFO as generic FIFO memory when USB is not in use. This extra 208 bytes of memory is excellent for data buffer. However, FIFO memory is a slow memory which takes at most 4 CPU clock cycles to access from read/write operation kick-start.

FIFOCON: FIFO Control Register
TYPE: R/W
ADDRESS: 0x9C

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	FEN	ONLY	-	-	-	-	WR	RD
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
FEN	RW	FIFO interface enable bit	0: disable 1: enable
ONLY	RW	FIFO only mode enable bit This mode isolates the FIFO from USB. In normal mode, FIFO interface completes memory access only when USB is idle	0: disable 1: enable
WRS	RW	Write operation kick start bit Write '1' to kick start write operation. This bit reset to '0' when the write operation is finished.	0: inactive 1: kick-start write
RDS	RW	Read operation kick start bit Write '1' to kick start read operation. This bit reset to '0' when the read operation is finished.	0: inactive 1: kick-start read

FIFODATA: FIFO Data Register
TYPE: R/W
ADDRESS: 0x9F

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	FIFODATA							
Default	0x00							

FIFOPTR: FIFO Pointer Register
TYPE: R/W
ADDRESS: 0x9D

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	FIFOPTR							
Default	0x00							

7.10.1 Operation Flow

Read Operation Flow

1. Enable FIFO interface by setting EN in FIFOCON
2. Select operation mode (ONLY in FIFOCON)
3. Set up desired address to FIFOPTR
4. Kick start the read operation by writing ‘1’ to RDS in FIFOCON
5. Poll RDS until it is cleared
6. Data is ready to ready from FIFODATA

RX Operation Flow

1. Enable FIFO interface by setting EN in FIFOCON
2. Select operation mode (ONLY in FIFOCON)
3. Set up desired address to FIFOPTR
4. Put the write-in data to FIFODATA
5. Kick start the write operation by writing ‘1’ to WRS in FIFOCON
6. The write operation is completed when WRS is cleared

Caution:

(1) Never do arithmetic, logic and bit operation on FIFODATA. FIFODATA is designed for MOV operation only

7.11 Watchdog Timer

The WDT is clocked by internal free-running RC oscillator, which WDT operates properly when there is no clock from crystal oscillator. When device resets, the WDT is disabled and user should enable the WDT if it is needed.

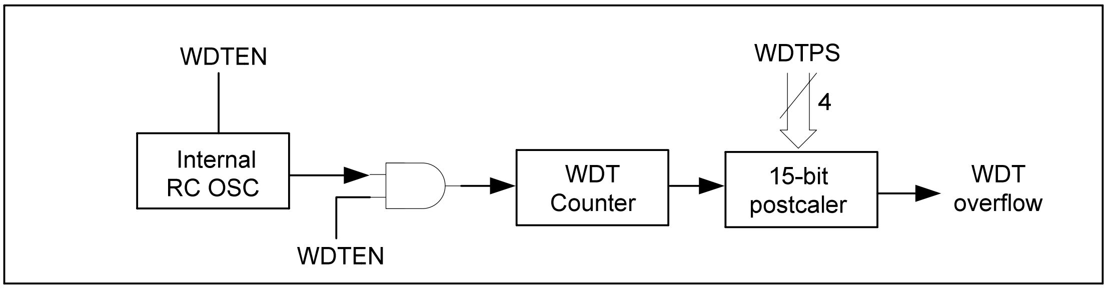


Figure 7-29: Block diagram of watchdog timer

WDTCON: Watchdog Timer Control Register
TYPE: R / W
ADDRESS: 0xA1

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	WDTTO	WDTPD	WDTPND	WDTEN	WDTPS[3]	WDTPS[2]	WDTPS[1]	WDTPS[0]
Default	0	0	0	0	0	0	0	0

Name	Mode	Description	Setting
WDTTO	RW	Watchdog time timeout flag	0: inactive 1: time out
WDTPD	R	Watchdog timer power down indicator Read out ‘0’ before power down operation; Read out ‘1’ after power down operation. This flag is cleared if software reset the watchdog timer	0: inactive 1: active
WDTPND	RW	Watchdog timer interrupt pending flag Write ‘0’ to this flag clears the watchdog timer	0: inactive 1: time out
WDTEN	RW	Watchdog timer enable bit	0: disable 1: enable
WDTPS	RW	Watchdog timer postscaler setting	Refer to Table 7-8

7.11.1 Timer Function

The time interval of the watchdog timeout can be calculated as:

$$T_{timeout} = \frac{256 \times 2^{WDTPS}}{f_{RC}}, \quad f_{RC} = 16\text{KHz (typical)}$$

In the default configuration (WDTPS is 0, thus postscaler divide ratio set to 1:1), the 8-bit counter overflows in 16 milliseconds.

Table 7-8: Postscaler settings

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

7.11.2 Clear Watchdog Timer

In normal operation, watchdog timer has to be cleared periodically to prevent watchdog timeout. To do that, user should write ‘0’ to WDTPND before every watchdog timeout.

7.11.3 Wakeup, Reset and Interrupt

To cope with different system recovery strategies, user can define the severity of failure alarm by setting watchdog overflow as interrupt source or reset source. The setting is in register FSET. User should either enable watchdog interrupt or enable watchdog reset. The value in FSET does not change to default after watchdog reset to avoid unintended change.

Watchdog interrupt has the highest priority among all other interrupts. The system can get into recovery subroutine even during (1) serving other interrupts; (2) IDLE mode; and (3) HALT mode. User is recommended to use watchdog interrupt if intended to keep the peripherals operating during recovery.

Watchdog reset forces the system to fallback to default status in any situation, giving the most immediate response to failure conditions. After watchdog reset, WDTPD will be asserted. User should detect WDTPD in reset routine to do suitable system recovery tasks.

In system that does not require failure detection or recovery, watchdog timer can be used as generic timer with interrupt capability. It is especially useful when out of timer resources. However, the clock of watchdog timer is generated from internal RC oscillator, the period of the watchdog clock is varied among chips.

FSET: **Special Function Setting Register**
TYPE: **R / W**
ADDRESS: **0xBF**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic							EWDT	WDTRST
Default							1	0

Name	Mode	Description	Setting
EWDT	RW	Watchdog timer interrupt enable bit	0: disable 1: enable
WDTRST	RW	Watchdog timer reset enable bit	0: disable 1: enable

8 Code Protection

Code protection is always a severe but unsolvable problem in class 8051/8052. The keep the code safe, CFP5102 deploys the follow practice:

1. When CFP5102 is forced to use external memory by pull-down pin EA, the OTP is disabled automatically, which means no access to OTP is available.
2. There is a control bit (PROT in CCON) restricts the instruction MOVC to access OTP, when the bit is set, instruction MOVC cannot read out correct data from OTP. This avoids hackers to crack the source code from manipulating the code in external memory. This bit can be cleared only in system resets. It is recommended to enable this feature in the boot-loader stored in OTP.
3. The program sent to CFP5102 in in-system programming is protected by a 24-bit random key. This avoids hackers to capturing the program by cracking the programmer.

CCON: **CPU Control Register**
TYPE: **R/W**
ADDRESS: **0xAB**

Position	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Mnemonic	PROT							
Default	0							

Name	Mode	Description	Setting
PROT	RW	OTP code protection enable bit. When set to ‘1’, MOVC to OTP memory is disabled. This bit can clear by reset sources only	0: disable 1: enable

9 Electrical Characteristics

9.1.1 Absolute Maximum Rating

Ambient temperature under bias	0°C to +125°C
Storage temperature	-65°C - 150°C
Voltage on DVDD with respect to VSS	0V to +3.6V
Voltage on VDDIO with respect to VSS	0V to +5.5V

9.1.2 DC Characteristics

POR DC Voltage Parameters

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

DC Voltage Parameters

Symbol	Characteristics	Min	Typ	Max	Units	Conditions
V_{DD}	Supply Voltage	3.0	3.3	3.6	V	Without LDO Input from pin DVDD
		3.0	3.3	3.6	V	Using LDO Output from LDO
V_{DDIO}	Supply Voltage to IO	3.0	3.3	3.6	V	Without LDO
		4.5	5.0	5.5	V	Using LDO
V_{IL}	Input Low Voltage I/O ports	V_{SS}	-	0.8	V	V_{DDIO} = 3.3V
		V_{SS}	-	0.3xV_{DD}	V	V_{DDIO} = 5.0V
V_{IH}	Input High Voltage I/O ports	2.0	-	V_{DD}	V	V_{DDIO} = 3.3V
		0.7xV_{DD}	-	V_{DD}	V	V_{DDIO} = 5.0V
V_{OL}	Output Low Voltage I/O ports	-	-	0.65	V	V_{DDIO} = 3.3V, I_{OL}= 8mA
		-	-	0.60	V	V_{DDIO} = 5.0V, I_{OL}= 8mA
V_{OH}	Output High Voltage I/O ports	2.80	-	-	V	V_{DDIO} = 3.3V, I_{OH}= 8mA
		4.65	-	-	V	V_{DDIO} = 5.0V, I_{OH}= 8mA
V_{SYS}	Hysteresis of I/O Schmitt Trigger	-	200	-	mV	V_{DDIO} = 3.3V
		-	400	-	mV	V_{DDIO} = 5.0V

DC Current Parameters

Symbol	Characteristics	Min	Typ	Max	Units	Conditions
I_{IDLE} (without LDO)	Idle mode current	-	5.80	-	mA	DPLL disable, OTP disable, all ports pull-up $F_{system\_clock}^{*}$ = 24MHZ 32,768 Hz oscillator off V_{DD} = 3.3V CKCON1 = 0x60
	Idle mode current (32,768 Hz oscillator in normal mode)	-	11.18	-	uA	DPLL disable, OTP disable, all ports pull-up $F_{system\_clock}^{*}$ = 32,768 Hz 24MHz oscillator off V_{DD} = 3.3V CKCON1 = 0x60
	Idle mode current (32,768 Hz oscillator in low power mode)	-	4.40	-	uA	
I_{HALT} (without LDO)	Halt mode current		1.23	-	mA	DPLL disable, OTP disable, all ports pull-up $F_{system\_clock}^{*}$ = 24MHZ 32,768 Hz oscillator off V_{DD} = 3.3V
	Halt mode current (32,768 Hz oscillator in normal mode)		9.20	-	uA	DPLL disable, OTP disable, all ports pull-up $F_{system\_clock}^{*}$ = 32,768 Hz 24MHz oscillator off V_{DD} = 3.3V
	Halt mode current (32,768 Hz oscillator in low power mode)		3.29	-	uA	
I_{PD}	Power Down mode current (without LDO)		0.43	-	uA	DPLL disable, OTP disable, all ports pull-up High-speed oscillator off 32,768 Hz oscillator off V_{DD} = 3.3V
I_{LDO}	LDO quiescent current		45.89	-	uA	When V_{DDIO} = 5.0V, DVDD output around 3.3V
I_{DD} (without LDO)	Normal mode current	-	18.06	-	mA	DPLL disable, all ports pull-up $F_{system\_clock}^{*}$ = 24MHZ 32,768 Hz oscillator off V_{DD} = 3.3V CKCON1 = 0x1E
		-	23.91	-	mA	DPLL enable, all ports pull-up $F_{system\_clock}^{*}$ = 24MHZ 32,768 Hz oscillator off V_{DD} = 3.3V CKCON1 = 0x1E
	Normal mode current (Disable OTP low power mode)	-	5.51	-	mA	DPLL disable, all ports pull-up $F_{system\_clock}^{*}$ = 4MHZ 32,768 Hz oscillator off VDD = 3.3V CKCON1 = 0x1E
	(Enable OTP low power mode)	-	4.49	-	mA	

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

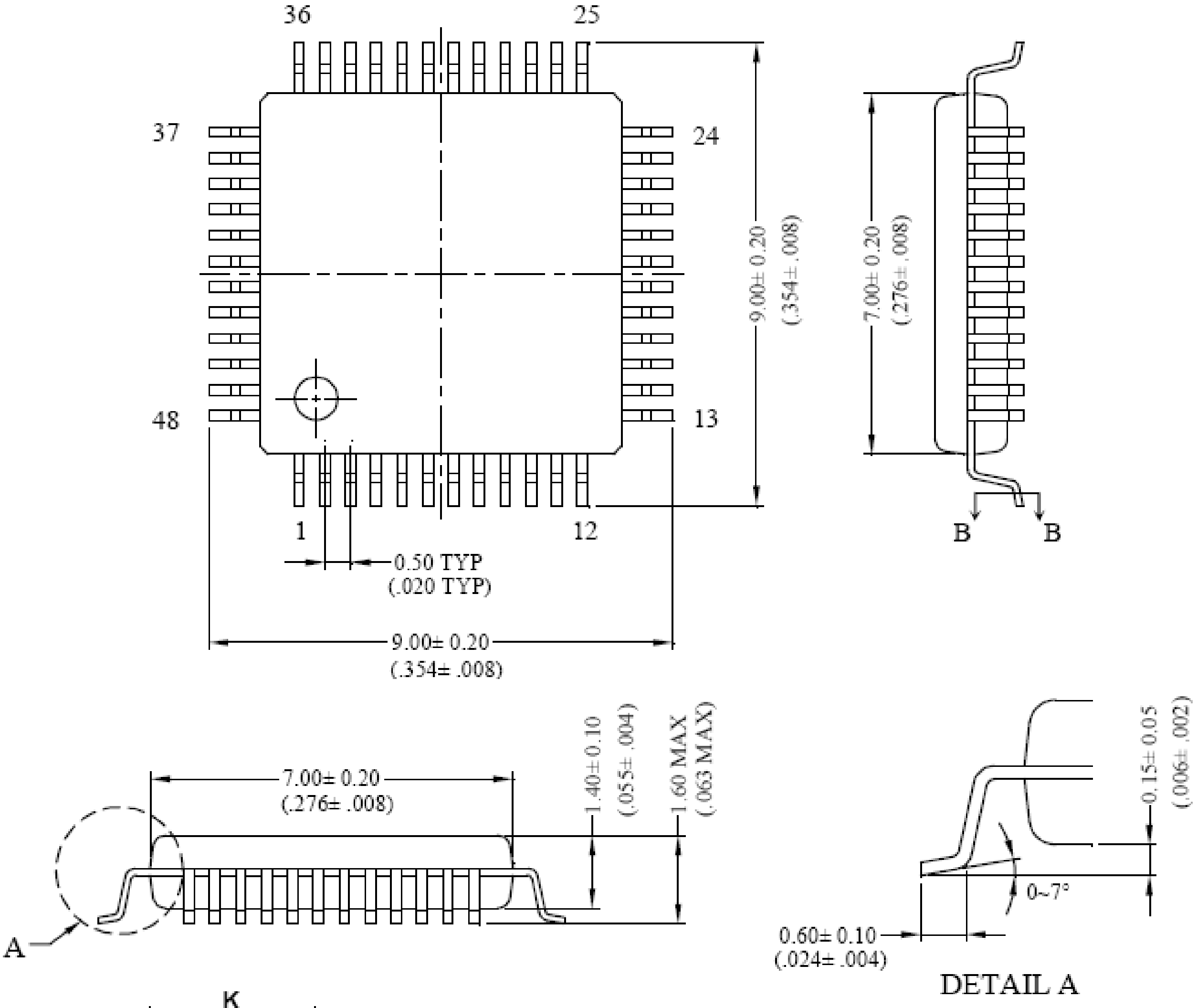
9.1.3 AC Parameters

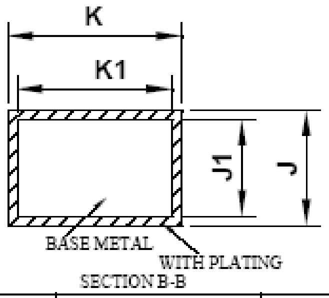
Symbol	Characteristics	Min	Typ	Max	Units	Conditions
F_{OSC}	External OSC1 Frequency	DC	-	24	MHz	
	Oscillator Frequency	1	-	24	MHz	
F_{CORE}	Digital Core Operating Frequency	-	-	24	MHZ	V_{DD} = 3.3V
T_{CY}	Instruction Cycle Time	41.7	-	DC	ns	V_{DD} = 3.3V
T_{MCLR}	MCLR Pulse Width	2	-	-	us	V_{DD} = 3.3V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

ADC Characteristics

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	C_{load} = 100pF R_{load} = 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

10 Package Outline





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)

11 Revision History

Date	Version	Author	Notice
2007.12.03	V0.1	Merlin	1. first version.