MPC235
Low-speed USB micro-controller

General Description

The MPC235 is a 65C02 MCU with an embedded 8k bytes ROM, a 256 bytes RAM, a watch-dog timer, a USB and PS/2 combo interfaces, can be implemented via the USB bus line, D+ and D- pins, by the user’s program. The USB features fully meets the low-speed USB Specification version 1.1. It will be very suitable for the low-cost keyboard, joystick, I-toy, and some products like the hand-held devices, which need to download/ upload data through the PC system.

Features

- 8-bit 65C02 micro-controller
- The MPC235 is mask version of the MPC2F35
- Operation voltage: 4.35V to 5.5V
- Memory:
  - 8K Bytes ROM
  - 256 Bytes RAM
- 34 programmable GPIO:
  - 4 LED direct sink pins shared with Port0 (LED0/1/2/3)
  - 2 external interrupt pins (INT0, INT1)
  - Port3 provides the pin interrupt
  - 26 bi-directional I/O pins for Port1/ 2/3/4
- One 8-bit programmable timer
- Built-in power-on reset
- One watchdog timer
- Low-speed USB Specification version 1.1 compliance
  - Supports 4 endpoints, where EP0 is control endpoint, and EP1/2/3 are data endpoints
  - Integrated USB transceiver, and Built-in pull-up resistor support
- Provides remote-wake-up/host-resume from suspend mode.
- On chip 5V to 3.3V regulator support
- Built-in 6MHz oscillator
- External oscillator detector
- Support suspend/normal mode for power management
- Packages:
  - 28-SSOP: MPC235L
  - 40-PDIP: MPC235E2

This document contains information on a new product under development by MEGAWIN. MEGAWIN reserves the right to change or discontinue this product without notice.

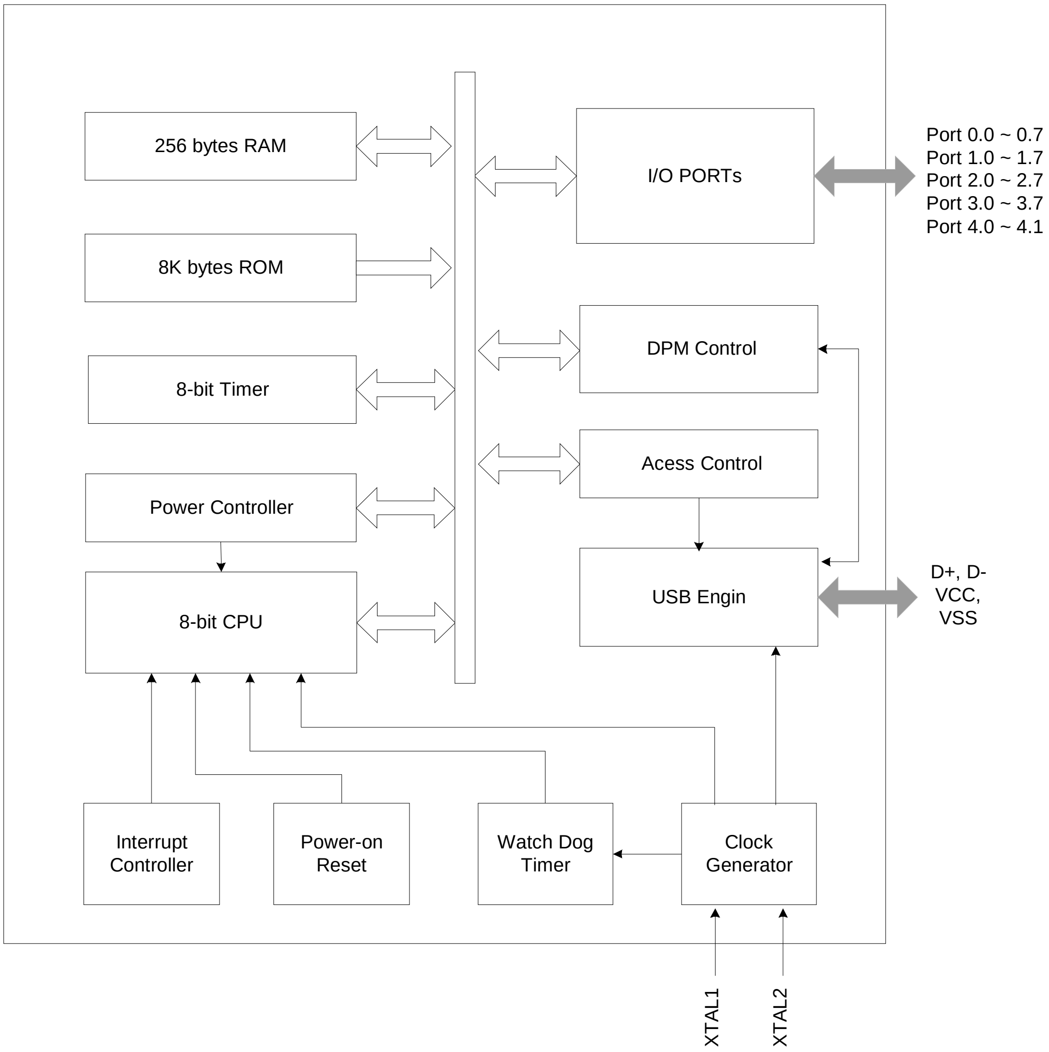
MEGAWIN

Pad Description

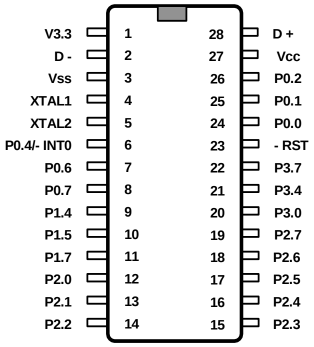
PIN Name	I/O	Description
P0.0~0.3	I/O	Bi-directional I/O (sink LED directly)*
P0.4/INT0	I/O	Bi-directional I/O with external interrupt 1
P0.5/INT1	I/O	Bi-directional I/O with external interrupt 2
P0.6~0.7	I/O	Bi-directional I/O
P1.0~1.7	I/O	Bi-directional I/O
P2.0~2.7	I/O	Bi-directional I/O
P3.0~3.7	I/O	Bi-directional I/O
P4.0~4.1	I/O	Bi-directional I/O
RESB	I	Reset pin
XTAL1	I	6MHz crystal or resonator in
XTAL2	I	6MHz crystal or resonator out
D+	I/O	USB data + with PS/2 compatible I/O
D-	I/O	USB data - with PS/2 compatible I/O
V_{CC}	I	Voltage supply
V_{SS}	I	Ground
V3.3	O	3.3V regulated output, a 0.1uF to 1uF capacitor should be added on this pin

* P0.0~0.3 can be used to sink LED directly (without any external resistor), i.e. the sink ability of these three pins are weaker than other I/O pins.

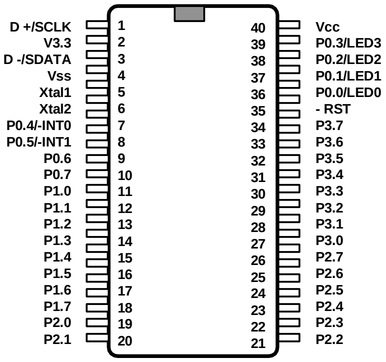
Block Diagram



Packages



MPC235L



MPC235E2

Function Description

Registers

	A
	Y
	X
	P
PCH	PCL
1	S

Accumulator

The accumulator is a general-purpose 8-bit register, which stores the results of most arithmetic and logic operations. In addition, the accumulator usually contains one of two data words used in these operations.

Index Register (X, Y)

There are two 8-bit index registers (X and Y), which may be used to count program steps or to provide an index value to be used in generating an effective address. When executing an instruction, which specifies indexed addressing, the CPU fetches the OP Code and the base address, and modifies the address by adding the index register to it prior to performing the desired operation. Pre- or post-index of index address is possible.

Processor Status Register (P)

The 8-bit processor status register contains seven status flags. Some of the flags are controlled by the program, others may be controlled both the program and the CPU.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N	V	1	B	D	I	Z	C

N: Signed flag, 1 = negative, 0 = positive

V: Overflow flag, 1 = true, 0 = false

B: BRK interrupt command, 1 = BRK, 0 = IRQB

D: Decimal mode, 1 = true, 0 = false

I: IRQB disable flag, 1 = disable, 0 = enable

Z: Zero flag, 1 = true, 0 = false

C: Carry flag, 1 = true, 0 = false

Program Counter (PC)

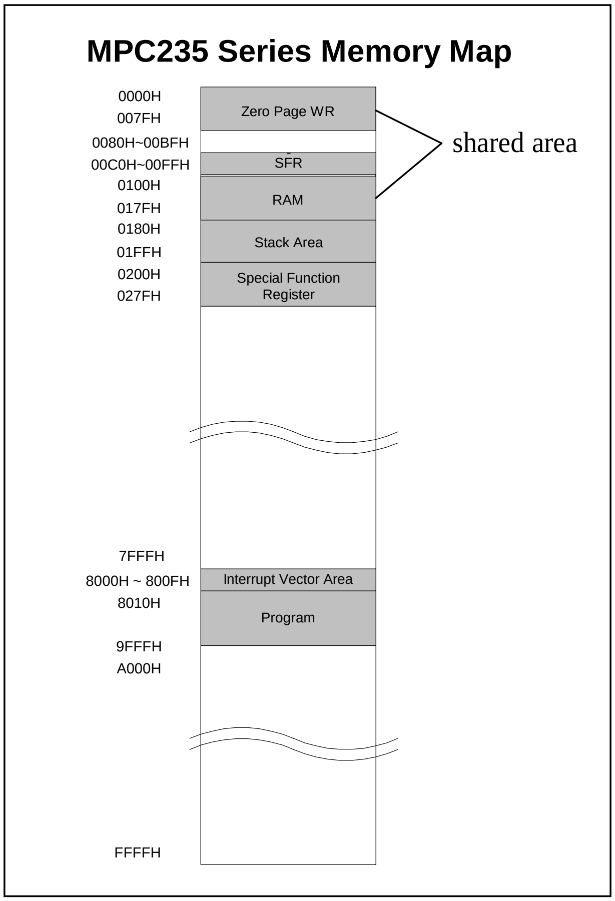
The 16-bit program counter register provides the addresses, which step the micro-controller through sequential program instructions. Each time the micro-controller fetch an instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order 8 bits of the address bus and the higher byte of the program counter (PCH) is placed on the high-order 8 bits. The counter is incremented each time an instruction or data is fetched from program memory.

Stack Pointer (S)

The stack pointer is an 8-bit register, which is used to control the addressing of the variable-length stack. The stack pointer is automatically incremented and decremented under control of the micro-controller to perform stack manipulations under direction of either the program or interrupts (/NMI or /IRQ). The stack allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer is initialized by the user’s firmware.

Memory Map

There are 256 bytes SRAM in MPC235. They are working RAM (0000H to 007FH) and stacks (0180H to 01FFH). Locations 0100h to 017FH and the locations 0000h to 007FH share the same memory block. The address 00C0H to 00FFH and 0200H to 027FH are special function registers area. The 8k bytes ROM are addressed from 8000H to 9FFFH. The address mapping of MPC235 series is shown as below.



Special Function Register (SFR)

The address 00C0H to 00FFH and 0200H to 027FH are reserved for special function registers (SFR).

The SFR is used to control or store the status of I/O, timers, system clock and other peripheral.

SFR (special function register): 00C0H ~ 00FFH (page 0 area)

Address	Content	Default	Address	Content	Default
00C0		X	00D0	P0_BUF	00
00C1	IRQ_EN	X	00D1	P1_BUF	00
00C2	IRQ_ST	00	00D2	P2_BUF	00
00C3	IRQ_CLR	00	00D3	P3_BUF	00
00C4		X	00D4	P4_BUF	00
00C5	TM0	00	00D5		X
00C6	TM0_CTL	00	00D6		X
00C7		X	00D7		X
00C8		X	00D8	P0	X
00C9		X	00D9	P1	X
00CA		X	00DA	P2	X
00CB		X	00DB	P3	X
00CC		X	00DC	P4	X
00CD		X	00DD		X
00CE		X	00DE	WDT_ST	00
00CF		X	00DF	WDT_CLR	X

Address	Content	Default	Address	Content	Default
00E0	USB_CTL	00	00F0		X
00E1	USB_ADDR	00	00F1		X
00E2	USB_DI / USB_DO	00	00F2		X
00E3		X	00F3		X
00E4		X	00F4		X
00E5		X	00F5		X
00E6		X	00F6		X
00E7		X	00F7		X
00E8	DPM_CTL	00	00F8		X
00E9	DPMO	00	00F9		X
00EA	DPMI	X	00FA		X
00EB		X	00FB		X
00EC		X	00FC		X
00ED		X	00FD		X
00EE		X	00FE		X
00EF		X	00FF		X

SFR (special function register): 0200H ~ 027FH

Address	Content	Default	Address	Content	Default
0200	PWR_CTL	00	0210		X
0201	FCPU_SR	00	0211		X
0202	RLH_EN	00	0212		X
0203		X	0213		X
0204		X	0214		X
0205	P0_MFR	00	0215		X
0206		X	0216		X
0207		X	0217		X
0208	FLASH_INFO	X	0218		X
0209	PRGM_STS	X	0219		X
020A		X	021A		X
020B		X	021B		X
020C		X	021C		X
020D		X	021D		X
020E		X	021E		X
020F		X	021F		X

Address	Content	Default	Address	Content	Default
0220		X	0230		X
0221		X	0231		X
0222		X	0232		X
0223		X	0233		X
0224		X	0234		X
0225		X	0235		X
0226		X	0236		X
0227		X	0237		X
0228		X	0238		X
0229		X	0239		X
022A		X	023A		X
022B		X	023B		X
022C		X	023C		X
022D		X	023D		X
022E		X	023E		X
022F		X	023F		X

Special Function Register, Continued

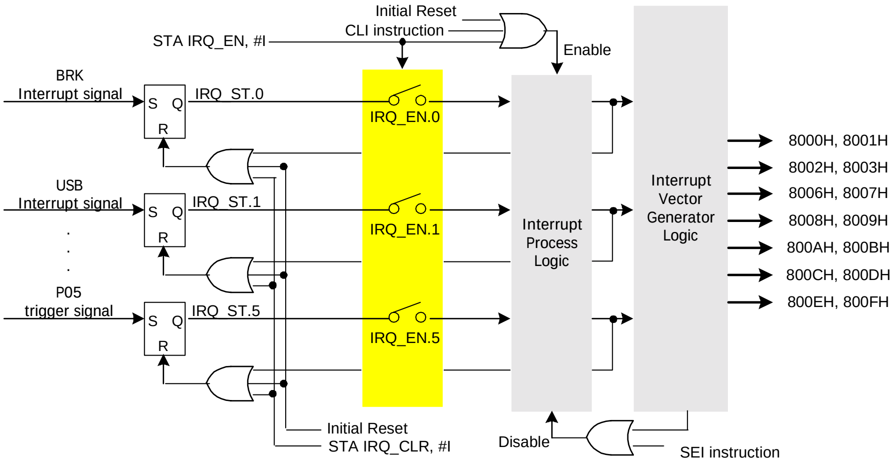
Address	Content	Default	Address	Content	Default
0240	P0_CR	00	0250	P4_CR	00
0241	P0_MR	00	0251	P4_MR	00
0242		X	0252		X
0243		X	0253		X
0244	P1_CR	00	0254		X
0245	P1_MR	00	0255		X
0246		X	0256		X
0247		X	0257		X
0248	P2_CR	00	0258		X
0249	P2_MR	00	0259		X
024A		X	025A		X
024B		X	025B		X
024C	P3_CR	00	025C		X
024D	P3_MR	00	025D		X
024E		X	025E		X
024F		X	025F		X

Address	Content	Default	Address	Content	Default
0260		X	0270		X
0261		X	0271		X
0262		X	0272		X
0263		X	0273		X
0264		X	0274		X
0265		X	0275		X
0266		X	0276		X
0267		X	0277		X
0268		X	0278		X
0269		X	0279		X
026A		X	027A		X
026B		X	027B		X
026C		X	027C		X
026D		X	027D		X
026E		X	027E		X
026F		X	027F		X

Interrupt Vectors

Vector Address	Item	Priority	Properties	Memo
8000H, 8001H	BRK	2	Int.	Software BRK interrupt vector
8002H, 8003H	RESET	1	Ext.	Initial reset
8006H, 8007H	USB	3	Int.	USB interrupt
8008H, 8009H	TM0	4	Int.	Timer 0 overflow interrupt
800AH, 800BH	P3	5	Ext.	Port P3 interrupt vector
800CH, 800DH	P04	6	Ext.	Port P0.4 interrupt vector
800EH, 800FH	P05	7	Ext.	Port P0.5 interrupt vector

There are seven interrupt sources provided in MPC235. The flag IRQ_EN and IRQ_ST are used to control the interrupts. When flag IRQ_ST is set to ‘1’ by hardware and the corresponding bits of flag IRQ_EN has been set by firmware, an interrupt is generated. When an interrupt occurs, all of the interrupts are inhibited until the CLI or STA IRQ_EN, #I instruction is invoked. Executing the SEI instruction can also disable the interrupts.



Interrupt Registers

IRQ enable flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C1H	IRQ_EN	-	-	P05	P04	P3	TM0	USB	-	√	√

Program can enable or disable the ability of triggering IRQ through this register.

0: Disable (default “0” at initialization)

1: Enable

USB: USB finished Rx or Tx data

TM0: Timer0 underflow

P3: Falling edge occurs at port 3 input mode

P04, P05: Falling edge occurs at P0.4/P0.5 input mode

IRQ status flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C2H	IRQ_ST	-	-	P05	P04	P3	TM0	USB	-	√	-

When IRQ occurs, program can read this register to know which source triggering IRQ.

IRQ clear flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C3H	IRQ_CLR	-	-	P05	P04	P3	TM0	USB	-	-	√

Program can clear the interrupt event by writing ‘1’ into the corresponding bit.

Watchdog Timer (WDT)

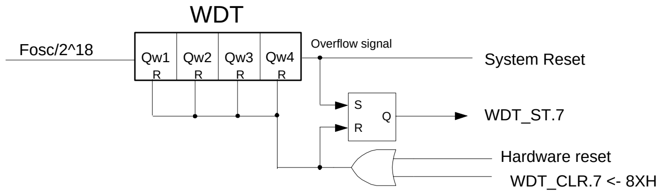
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00DEH	WDT_ST	RSTS	-	-	-	Bit 3	Bit 2	Bit 1	Bit 0	√	-
00DFH	WDT_CLR	CLR	-	-	-	-	-	-	-	-	√

Bit3~Bit0: Contents of WDT

RSTS: WDT reset status, set by hardware when WDT overflows, clear by firmware or hardware reset

CLR: RSTS clear and WDT reset control bit, program can clear RSTS and reset WDT by writing “1” into this bit

The watchdog timer (WDT) is organized as a 4-bit counter designed to prevent the program from unknown errors. If the WDT overflows, the WDT reset function will be performed. RSTS (Bit 7 of WDT_ST) is set by hardware when the WDT overflows and is cleared by hardware reset or writing 1 to bit 7 of WDT_CLR. The interval of WDT to cause reset is about 0.7s at 6MHz external oscillator. Programming one into the bit 7 of WDT_CLR register can reset the contents of the WDT. In normal operation, the application program must reset WDT before it overflows. A WDT overflow indicates that operation is not under control and the chip will be reset. The organization of the watchdog timer is shown as below



System Control Registers

Power saving control

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
0200H	PWR_CTL	LVDT	-	-	-	-	-	CKC	HALT	-	√

LVDT: low-voltage detector disable bit

0 (default): enable low-voltage detector

1: disable low-voltage detector

When the low-voltage detector is enabled, and it senses the power voltage is lower than 3.6V, the chip would be reset automatically.

CKC: Oscillator control bit. 1: disable OSC, 0: enable OSC

HALT: FCPU off-line control bit. 1: FCPU off-line, 0: FCPU on-line

Program can switch the normal operation mode to the power-saving mode for saving power consumption through this register. There are two power saving mode in this system.

Stop mode: *(PWR_CTL.CKC = 1)*

System clock stops oscillating. The uC can be awakened from stop mode by 3 ways: port 3 interrupt, hardware reset, power-on reset, USB host wake up call.

Halt mode: *(PWR_CTL.HALT = 1)*

The FCPU clock in off-line status. The oscillator oscillates or not depends on the content of PWR_CTL.CKC. The uC can be awakened from halt mode by 3-ways: interrupts (USB, Timer 0, Port 3, Port0.4, Port0.5) assigned in the RLH_EN register, hardware reset, or power-on reset.

FCPU selector

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
0201H	FCPU_SR	-	-	-	-	-	-	-	CKS	-	√

CKS: F_{CPU} clock source select. 0: $F_{OSC/2}$, 1:F_{OSC}

Release halt mode enable flag

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
0202H	RLH_EN	-	-	P05	P04	P3	TM0	USB	-	-	√

Program can select interrupt sources to release halt mode through this register.

0: Disable (default)

1: Enable

Release halt status flag is the IRQ_ST register.

Timer

Timer 0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00C5H	TM0	T7	T6	T5	T4	T3	T2	T1	T0	√	√
00C6H	TM0_CTL	-	STC	RL/S	-	-	TKI2	TKI1	TKI0	√	√

STC: Timer clock disable/enable. 0: disable timer clock, 1: enable timer clock

RL/S: Auto-reload disable/enable. 0: enable auto-reload, 1: disable auto-reload

TKI2	TKI1	TKI0	Selected TM0 input clock source
0	0	0	FOSC / 8
0	0	1	FOSC / 16
0	1	0	FOSC / 32
0	1	1	FOSC / 64
1	0	0	FOSC / 128
1	0	1	FOSC / 256
1	1	0	FOSC / 512
1	1	1	FOSC / 1024

Timer 0 is a dual function 8-bit counter. When timer 0 is under user’s firmware control, it will pre-load value to counter at the rising edge of TM0_CTL.STC and its underflow frequency of timer 0 can be calculated with the following equation:

$F_{TM0\_UV} = F_{TM0CK} / (TM0+1)$ where F_{TM0CK} is selected by TM0_CTL.TKI2/11/10

For example: if $F_{TM0CK} = F_{OSC} / 32$ (187.5 KHz)

TM1	F_{TM0_UV} Frequency
00H	invalid
01H	93.75 KHz
02H	62.5 KHz
…	…
FFH	732.42 Hz

Writing data to the TM0 would write data to the reload buffer of the timer 0. Reading data from the TM0 would read the run-time value from the counter.

USB

USB register access control

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00E0H	USB_CTL	REGC	-	-	-	-	-	UWT	URD	√	√
00E1H	USB_ADDR	-	-	UA5	UA4	UA3	UA2	UA1	UA0	√	√
00E2H	USB_DI	UDI7	UDI6	UDI5	UDI4	UDI3	UDI2	UDI1	UDI0	-	√
00E2H	USB_DO	UDO7	UDO6	UDO5	UDO4	UDO3	UDO2	UDO1	UDO0	√	-

USB_CTL:

REGC: 3.3V regulator control. 0: disable, 1: enable

URD: USB register read, writing 1 into this bit to read USB register (addressed by USB_ADDR)

UWT: USB register write, writing 1 into this bit to write USB register (addressed by USB_ADDR)

USB_ADDR: USB register address to be accessed

USB_DI: Data to be written into USB register (addressed by USB_ADDR)

USB_DO: Data read out from USB register (addressed by USB_ADDR)

The USB engine is an independent unit, which is Low-speed USB 1.1 version compliant, with transceiver and 3.3V regulator built-in. The 3.3V regulator can be controlled by user program through the USB_CTL.REGC control bit. The USB engine contains registers of its own, as attached in next page. User can access the USB registers through the access control registers provided here. The sequence to access USB register should be:

Write sequence:

1. Write the address of USB register to be accessed into USB_ADDR
2. Write 1 into USB_CTL.UWT
3. Write data into USB_DI
4. Write 0 into USB_CTL.UWT

Read sequence:

1. Write the address of USB register to be accessed into USB_ADDR
2. Write 1 into USB_CTL.URD
3. Read data from USB_DO
4. Write 0 into USB_CTL.URD

Whenever USB engine finished a transaction, it will generate an interrupt to acknowledge CPU. User can get information about the transaction through the above sequence. When USB engine received a reset instruction from host, it will reset itself and generate an interrupt. When USB engine received a wake-up instruction from host (the device is in suspend mode), it will generate a signal to enable oscillator. If host and the device are both in suspend mode, a falling edge on P3 can wake the device and thus remote wake up the host through USB engine.

USB SFR Category - Descriptions Summary

Mnemonic	USB Device SFRs	Address	Description							
DCON	Device Control Register	01H	TESTEN	–	–	–	–	EP3DIR	PUREN	CONPUEN
FADDR	Function Address Register	08H	–	A6	A5	A4	A3	A2	A1	A0
FPCON	Function Power Control Register	12H	–	–	FRWU	–	URST	–	FRSM	FSUS
Mnemonic	**USB Interrupt System SFRs**	**Address**	**Description**							
FIE	USB Function Interrupt Enable Register	18H	–	–	–	FRXIE3	FTXIE2	FTXIE1	FRXIE0	FTXIE0
FIFLG	USB Function Interrupt Flag Register	1AH	–	–	–	FRXD3	FTXD2	FTXD1	FRXD0	FTXD0
IEN1	USB Interrupt Enable Register	10H	–		–	–	EFSR	–	EF	–
Mnemonic	**USB Endpoint SFRs**	**Address**	**Description**							
EPINDEX	Endpoint Index Register	31H	–	–	–	–	–	–	EPINX1	EPINX0
EPCON*	Endpoint Control Register	21H	RXSTL	TXSTL	–	–	–	RXEPEN	–	TXEPEN
RXCNT*	Receive FIFO Byte-Count Register	26H	–	–	–	–	RXBC3	RXBC2	RXBC1	RXBC0
RXCON*	Receive FIFO Control Register	24H	RXCLR	–	–	RXFFRC	–	–	–	–
RXDAT*	Receive FIFO Data Register	23H	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
RXSTAT*	Endpoint Receive Status Register	22H	RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	–	–	–
TXCNT*	Transmit FIFO Byte-Count Register	36H	–	–	–	–	TXBC3	TXBC2	TXBC1	TXBC0
TXCON*	Transmit FIFO Control Register	34H	TXCLR	–	–	–	–	–	–	–
TXDAT*	Transmit FIFO Data Register	33H	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
TXSTAT*	Endpoint Transmit Status Register	32H	TXSEQ	–	–	–	TXSOVW	–	–	–

USB SFR Description

DCON: Device Control Register

Read/Write Address: 01H

Default: 0XXX_X001 System Reset

Bit Number	Bit Mnemonic	Function
7	TESTEN	TEST mode Enable: Use for test only. In normal operation, this bit should be cleared to “0”.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	-	Reserved: Write zero to this bit.
2	EP3DIR	Endpoint 3 Direction: When this bit is set by FW to enable endpoint 3 to be a TX endpoint. Endpoint 3 behaves as a RX endpoint when this bit is cleared to ‘0’.
1	PUREN	D- Pull-Up Resistor Enable: When this bit is set to ‘1’, enable internal D- pull-up resistor. After setting this bit, the device will act a connection to USB host.
0	CONPUEN	Device USP Connection Pull-Up Enable: This bit is used by FW to control whether device is connected to upper host/hub via driving bus SE0. Set ‘1’ to release bus to expose the D- pull-up resistor. Clear ‘0’ to force bus SE0 to inhibit the D- pull-up resistor. Default is set to ‘1’ after reset. FW can clear to ‘0’ to disable connection to upper host/hub.

FADDR: USB Function Address Register

Read/Write Address: 08H

Default: X000_0000 System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6:0	A [6:0]	Function Address: This register holds the address for the USB function. During bus enumeration, it is written with a unique value assigned by the host.

FPCON: Function Power Control Register

Read/Write Address: 12H

Default: XX0X_xX00 System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6	-	Reserved: Write zero to this bit.
5	FRWU	Function Remote Wake-up Trigger: This bit is used by the function to initiate a remote wake-up on the USB bus when uC is wake-up by the external trigger.
4	-	Reserved: Write zero to this bit.
3	URST	USB Reset Flag: Set by hardware when the function detects the USB bus reset. If this bit is set, and then the chip will generate the interrupt. Should be cleared by firmware when serving the USB reset interrupt.
2	-	Reserved: Write zero to this bit.
1	FRSM	Function Resume Flag: Set by hardware when the function detects the resume state on the USB bus. If this bit is set, and then the chip will generate the interrupt. Cleared by firmware when servicing the function resume interrupt.
0	FSUS	Function Suspend Flag: Set by hardware when the function detects the suspend state on the USB bus. If this bit is set, and then the chip will generate the interrupt. During the function suspend ISR, firmware should clear this bit before enter the suspend mode.

FIE: Function Interrupt Enable Register

Read/Write Address: 18H

Default: XXX0_0000 System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	FRXIE3	Function Transmit/Receive Interrupt Enable 3: Enables the transmit/receive done interrupt for function endpoint 3 (FRXD3/FTXD3).
3	FTXIE2	Function Transmit Interrupt Enable 2: Enables the transmit done interrupt for function endpoint 2 (FTXD2).
2	FTXIE1	Function Transmit Interrupt Enable 1: Enables the transmit done interrupt for function endpoint 1 (FTXD1).
1	FRXIE0	Function Receive Interrupt Enable 0: Enables the receive done interrupt for function endpoint 0 (FRXD0).
0	FTXIE0	Function Transmit Interrupt Enable 0: Enables the transmit done interrupt for function endpoint 0 (FTXD0).

FIFLG: Function Interrupt Flag Register

Read/Write	Address: 1AH
Default: XXX0_0000	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	FRXD3	Function Transmit/Receive Done Flag 3: For endpoint 3, uC can read/write-clear on this bit. This bit is cleared when firmware writes ‘1’ to it.
3	FTXD2	Function Transmit Done Flag 2: For endpoint 2, uC can read/write-clear on this bit. This bit is cleared when firmware writes ‘1’ to it.
2	FTXD1	Function Transmit Done Flag 1: For endpoint 1, uC can read/write-clear on this bit. This bit is cleared when firmware writes ‘1’ to it.
1	FRXD0	Function Receive Done Flag 0: For endpoint 0, uC can read/write-clear on this bit. This bit is cleared when firmware writes ‘1’ to it.
0	FTXD0	Function Transmit Done Flag 0: For endpoint 0, uC can read/write-clear on this bit. This bit is cleared when firmware writes ‘1’ to it.

IEN1: USB Interrupt Enable Register

Read/Write Address: 10H

Default: XXXX_0X0X System Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write “one” to this bit.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	EFSR	Enable Function Suspend/Resume: Function suspend/resume/usb_reset interrupt enable bit.
2	-	Reserved: Write zero to this bit.
1	EF	Enable Function: Transmit/receive done interrupt enable bit for USB function endpoints.
0	-	Reserved: Write zero to this bit.

EPINDEX: Endpoint Index Register

Read/Write Address: 31H

Default: XXXX_XX00 System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	-	Reserved: Write zero to this bit.
2	-	Reserved: Write zero to this bit.
1:0	EPINX1:0	Endpoint Index Bit 1:0: EPINDEX [7:0] = XXXX XX00: Function Endpoint 0 = XXXX XX01: Function Endpoint 1 = XXXX XX10: Function Endpoint 2 = XXXX XX11: Function Endpoint 3

EPCON: Endpoint Control Register (Endpoint-Indexed)

Read/Write Address: 21H

Default: 00XX_X0X0 System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	RXSTL	Stall Receive Endpoint: Set this bit to stall the receive endpoint.
6	TXSTL	Stall Transmit Endpoint: Set this bit to stall the transmit endpoint.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	-	Reserved: Write “one” to this bit.
2	RXEPEN	Receive Endpoint Enable: Set this bit to enable the receive endpoint. When disabled, the endpoint does not respond to a valid OUT or SETUP token.
1	-	Reserved: Write “one” to this bit.
0	TXEPEN	Transmit Endpoint Enable: This bit is used to enable the transmit endpoint. When disabled, the endpoint does not respond to a valid IN token.

RXCNT: Receive FIFO Byte-Count Register (Endpoint-Indexed)

Read Only Address: 26H

Default: XXXX_0000 System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	RXBC3	Receive Byte Count Bit 3: Store receives byte count. Maximum is 8 bytes.
2	RXBC2	Receive Byte Count Bit 2: Store receives byte count. Maximum is 8 bytes.
1	RXBC1	Receive Byte Count Bit 1: Store receives byte count. Maximum is 8 bytes.
0	RXBC0	Receive Byte Count Bit 0: Store receives byte count. Maximum is 8 bytes.

RXCON: Receive FIFO Control Register (Endpoint-Indexed)

Read/Write	Address: 24H
Default: 0XX0_XXXX	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	RXCLR	Receive FIFO Clear: Set this bit to flush the entire receive FIFO. All FIFO statuses are reverted to their reset states. Hardware clears this bit when the flush operation is completed.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	RXFFRC	FIFO Read Complete: Set this bit to release the receive FIFO when data set read is complete. Hardware clears this bit after the FIFO release operation has been finished.
3	-	Reserved: Write zero to this bit.
2	-	Reserved: Write zero to this bit.
1	-	Reserved: Write zero to this bit.
0	-	Reserved: Write zero to this bit.

RXDAT: Receive FIFO Data Register (Endpoint-Indexed)

Read Only	Address: 23H
Default: XXXX_XXXX	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7:0	RD [7:0]	Receive FIFO data specified by EPINDEX is stored and read from this register.

RXSTAT: Endpoint Receive Status Register (Endpoint-Indexed)

Read/Write	Address: 22H
Default: 0000_0XXX	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	RXSEQ	Receive Endpoint Sequence Bit (read, conditional write): The bit will be toggled on completion of an ACK handshake in response to an OUT token. This bit can be written by firmware if the RXOVW bit is set when written along with the new RXSEQ value.
6	RXSETUP	Received Setup Transaction: This bit is set by hardware when a valid SETUP transaction has been received. Clear this bit upon detection of a SETUP transaction or the firmware ready to handle the data/status stage of control transfer.
5	STOVW	Start Overwrite Flag (read-only): Set by hardware upon receipt of a SETUP token for any control endpoint to indicate that the receive FIFO is being overwritten with new SETUP data. This bit is used only for control endpoints.
4	EDOVW	End Overwrite Flag: This flag is set by hardware during the handshake phase of a SETUP stage. This bit is cleared by firmware to read FIFO data. This bit is only used for control endpoints.
3	RXSOVW	Receive Data Sequence Overwrite Bit: Write ‘1’ to this bit to allow the value of the RXSEQ bit to be overwritten. Writing a ‘0’ to this bit has no effect on RXSEQ. This bit always returns ‘0’ when read.
2	-	Reserved: Write zero to this bit.
1	-	Reserved: Write zero to this bit.
0	-	Reserved: Write zero to this bit.

TXCNT: Transmit FIFO Byte-Count Register (Endpoint-Indexed)

Write Only Address: 36H

Default: XXXX_XXXX System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	-	Reserved: Write zero to this bit.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	TXBC3	Transmit Byte Count Bit 3: Store transmits byte count. Maximum is 8 bytes.
2	TXBC2	Transmit Byte Count Bit 2: Store transmits byte count. Maximum is 8 bytes.
1	TXBC1	Transmit Byte Count Bit 1: Store transmits byte count. Maximum is 8 bytes.
0	TXBC0	Transmit Byte Count Bit 0: Store transmits byte count. Maximum is 8 bytes.

TXCON: Transmit FIFO Control Register (Endpoint-Indexed)

Read/Write	Address: 34H
Default: 0XXX_XXXX	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	TXCLR	Transmit FIFO Clear: Set this bit to flush the entire transmit FIFO. All FIFO statuses are reverted to their reset states. Hardware clears this bit when the flush operation is completed.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	-	Reserved: Write zero to this bit.
2	-	Reserved: Write zero to this bit.
1	-	Reserved: Write zero to this bit.
0	-	Reserved: Write zero to this bit.

TXDAT: Transmit FIFO Data Register (Endpoint-Indexed)

Write Only	Address: 33H
Default: XXXX_XXXX	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7:0	TD [7:0]	Data to be transmitted in the FIFO specified by EPINDEX is written to this register.

TXSTAT: Endpoint Transmit Status Register (Endpoint-Indexed)

Read/Write	Address: 32H
Default: 0XXX_0XXX	System Reset or USB Reset

Bit Number	Bit Mnemonic	Function
7	TXSEQ	Transmit Endpoint Sequence Bit (read, conditional write): The bit will be transmitted in the next PID and toggled on a valid ACK handshake of an IN transaction. This bit can be written by firmware if the TXSOVW bit is set when written along with the new TXSEQ value.
6	-	Reserved: Write zero to this bit.
5	-	Reserved: Write zero to this bit.
4	-	Reserved: Write zero to this bit.
3	TXSOVW	Transmit Data Sequence Overwrite Bit: Write a "1" to this bit to allow the value of the TXSEQ bit to be overwritten. Writing a "0" to this bit has no effect on TXSEQ. This bit always returns "0" when read.
2	-	Reserved: Write zero to this bit.
1	-	Reserved: Write zero to this bit.
0	-	Reserved: Write zero to this bit.

I/O Ports

Port 0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D0H	P0_BUF	BP07	BP06	BP05	BP04	BP03	BP02	BP01	BP00	√	√
00D8H	P0	P07	P06	P05	P04	P03	P02	P01	P00	√	-
0240H	P0_CR	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	√	√
0241H	P0_MR	-	-	MP05	MP04	-	-	MP01	MP00	√	√

Port 0 is an 8-bit I/O port; each pin can be programmed as input or output individually.

P0_BUF: Port 0 output buffer. When P0.n is configured as an output pin, it outputs the content of P0_BUF.n.

P0: Port 0 pad value.

P0_CR: p0.0~p0.7 is input or output. 0: input, 1: output

P0_MR: p0.0~p0.7, pull-high, CMOS/NMOS

P0_MR.0: P0.0 ~ P0.3 Pull-high control, 0: disable, 1:enable

P0_MR.1: P0.0 ~ P0.3 CMOS/NMOS selector, 0: CMOS, 1:NMOS

P0_MR.4: P0.4 ~ P0.7 Pull-high control, 0: disable, 1: enable

P0_MR.5: P0.4 ~ P0.7 CMOS/NMOS selector, 0: CMOS, 1:NMOS

At initial reset, the port P0 is all in input mode. Each pin of port P0 can be specified as input or output mode independently by the P0_CR registers. When P0 is used as output port, CMOS or NMOS open drain output type can be selected by the P0_MR register. Port P0 has 17kohm internal pull-high resistors that can be enabled/disabled by specifying the P0_MR.0 and P0_MR.4 respectively. The pull-high resistor is automatically disabled only when port is configured as CMOS output. Schmitt trigger circuit is added in the input path of P0.0~P0.3. User should be careful on setting pin as input with no pull high resistor since this setting has potential to cause leakage.

When P0.4 and P0.5 are set as input pins, they are interrupt sources. A falling edge at these two pin will set corresponding IRQ_ST bits to 1, and their interrupt subroutines will be executed if corresponding IRQ_EN bits are set.

Port 1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D1H	P1_BUF	BP17	BP16	BP15	BP14	BP13	BP12	BP11	BP10	√	√
00D9H	P1	P17	P16	P15	P14	P13	P12	P11	P10	√	-
0244H	P1_CR	CP17	CP16	CP15	CP14	CP13	CP12	CP11	CP10	√	√
0245H	P1_MR	-	-	MP15	MP14	-	-	MP11	MP10	√	√

Port 1 is an 8-bit I/O port; refer to port 0 for more information.

P1_BUF: Port 1 output buffer. When P1.n is configured as an output pin, it outputs the content of P1_BUF.n.

P1: Port 1 pad value.

P1_CR: P1.0 ~ P1.7 is input or output. 0: input, 1: output

P1_MR: P1.0 ~ P1.7, pull-high and CMOS/NMOS

Port 2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D2H	P2_BUF	BP27	BP26	BP25	BP24	BP23	BP22	BP21	BP20	√	√
00DAH	P2	P27	P26	P25	P24	P23	P22	P21	P20	√	-
0248H	P2_CR	CP27	CP26	CP25	CP24	CP23	CP22	CP21	CP20	√	√
0249H	P2_MR	-	-	MP25	MP24	-	-	MP21	MP20	√	√

Port 2 is an 8-bit I/O port; refer to port 0 for more information.

P2_BUF: Port 2 output buffer. When P2.n is configured as an output pin, it outputs the content of P2_BUF.n.

P2: Port 2 pad value.

P2_CR: P2.0 ~ P2.7 is input or output. 0: input, 1: output

P2_MR: P2.0 ~ P2.7, pull-high and CMOS/NMOS

Port 3

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D3H	P3_BUF	BP37	BP36	BP35	BP34	BP33	BP32	BP31	BP30	√	√
00D9H	P3	P37	P36	P35	P34	P33	P32	P31	P30	√	-
0244H	P3_CR	CP37	CP36	CP35	CP34	CP33	CP32	CP31	CP30	√	√
0245H	P3_MR	-	-	MP35	MP34	-	-	MP31	MP30	√	√

Port 3 is an 8-bit I/O port; refer to port 0 for more information.

P3_BUF: Port 3 output buffer. When P3.n is configured as an output pin, it outputs the content of P3_BUF.n.

P3: Port 3 pad value.

P3_CR: P3.0 ~ P3.7 is input or output. 0: input, 1: output

P3_MR: P3.0 ~ P3.7, pull-high and CMOS/NMOS

When P3 port is used as input mode, it is an interrupt source, a falling edge at any pin will set the IRQ_ST.P3. The same event can release suspend mode (enable oscillator), and release halt mode (resume Fcpu) if RLH_EN.P3 is set. An interrupt subroutine will be executed if IRQ_E.P3 is set.

Port 4

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00D1H	P4_BUF	BP47	BP46	BP45	BP44	BP43	BP42	BP41	BP40	√	√
00DCH	P4	-	-	-	-	-	-	P41	P40	√	-
0250H	P4_CR	-	-	-	-	-	-	CP41	CP40	√	√
0251H	P4_MR	-	-	-	-	-	-	MP41	MP40	√	√

Port 4 is a 2-bit I/O port; refer to port 0 for more information.

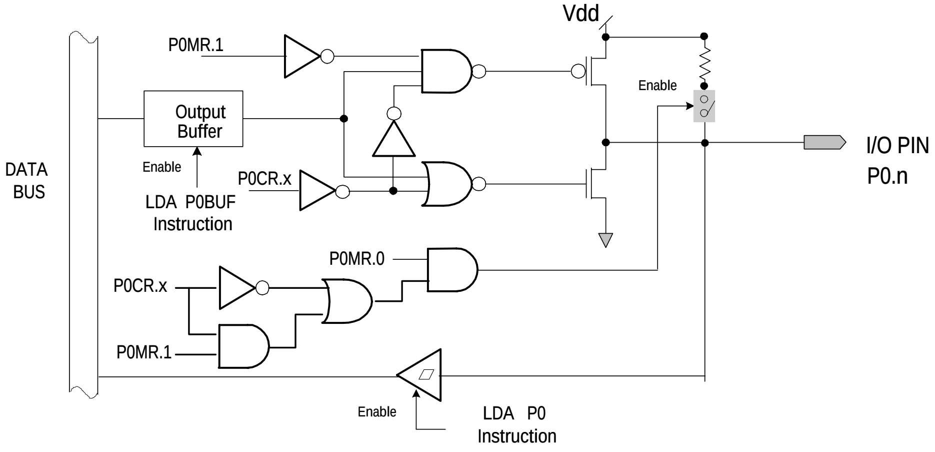
P4_BUF: Port 4 output buffer. When P4.n is configured as an output pin, it outputs the content of P4_BUF.n.

P4: Port 4 pad value.

P4_CR: P4.0 ~ P4.1 is input or output. 0: input, 1: output

P4_MR: P4.0 ~ P4.1, pull-high and CMOS/NMOS

Input/Output Pin --- P0~P4



DPM control

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00E8H	DPM_CTL	-	-	-	-	-	-	C1	C0	√	√
00E9H	DPMO	-	-	-	-	-	-	DPO	DMO	√	√
00EAH	DPMI	-	-	-	-	-	-	DPI	DMI	√	-

DPM_CTL:

C1, C0: D+/D- control selector.

0x: controlled by USB Engine

10: controlled by CPU

DPMO:

DPO/DMO: D+/D- pin output (at {DPM_CTL.C1, DPM_CTL.C0}=10). 0: output low, 1: pull high (input)

DPMI:

DPI/DMI: D+/D- pin value (Read only)

MPC235 provides a way to control D+ and D- pins by user's firmware. The control focuses on PS/2 interface and in system program operations. The DPM.DPI and DPM.DMI record the D+ and D- pin value respectively.

For PS/2 interface, firmware can judge the D+ and D- pins' connection be USB or PS/2 protocol by reading the value of DPI and DMI. The DPM_CTL.C1 and DPM_CTL.C0 set the controller of D+ and D- pins. If they are set to 10, the D+ and D- pins are under CPU's control, thus the USB function is unavailable. DPM.DPO/DMO sets the value of D+/D- pin when CPU controls the D+/D- pin; Writing 0 to DPO/DMO let the D+/D- pin to output low, writing 1 causes the pin to be pulled high (input mode). This I/O control would be enough to perform PS/2 operation.

Programming Notice

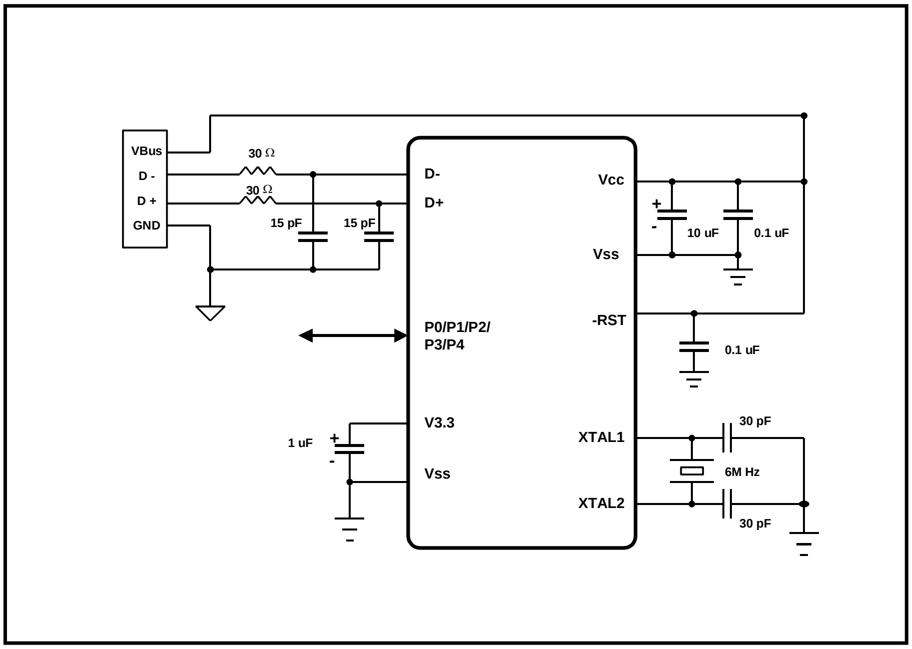
The status after different reset condition is listed below:

	Power on reset	**CPU RESB pin reset**	**WDT reset**
SRAM Data	Unknown	Unchanged	Unchanged
CPU Register	Unknown	Unknown	Unknown
Special Function Register	Default value	Default value (*)	Default value (*)

*: some SFR are unchanged

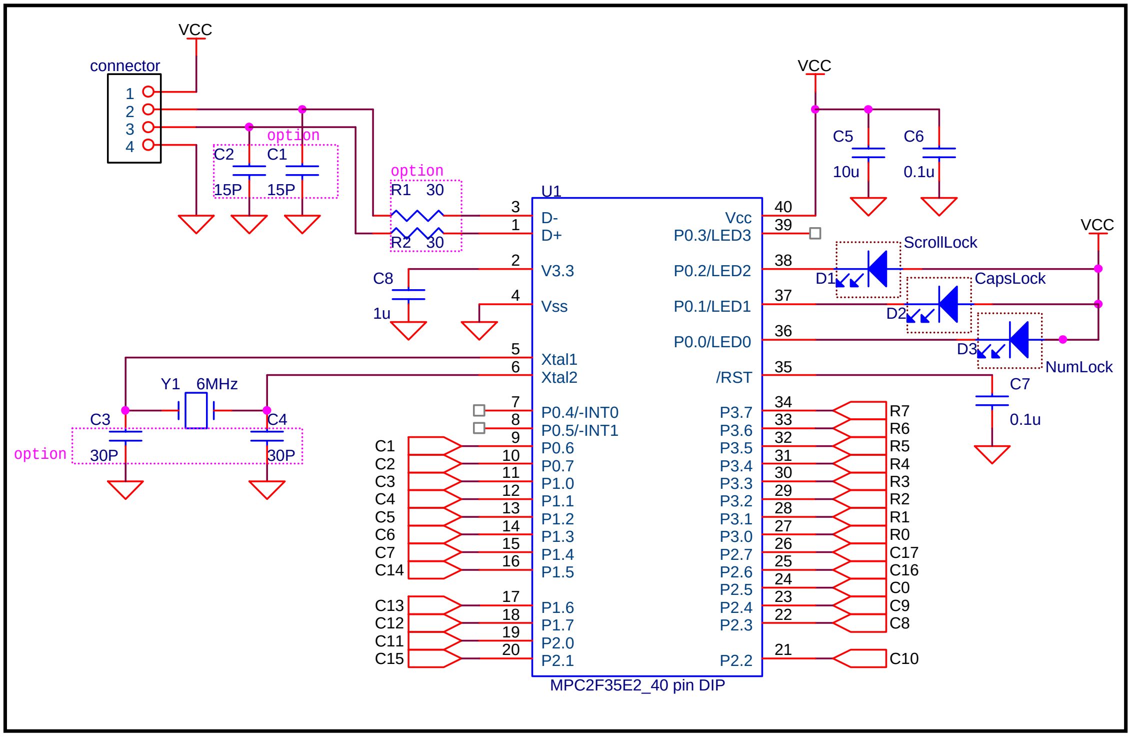
Application Circuit

1. Normal:

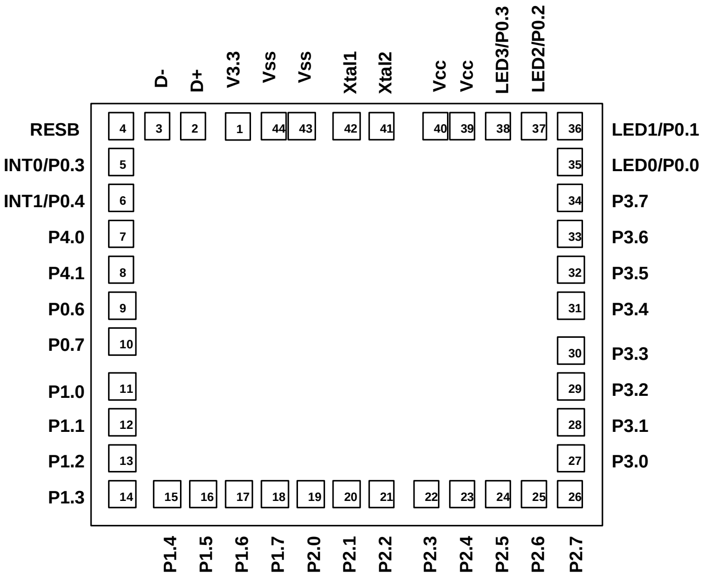


Note: The capacitor between RESB-pin and ground must be below 0.1uF.

2. USB keyboard:



Pad Assignment



The substrate should be connected to Vss

Timing

Absolute Maximum Rating

PARAMETER	RATING	UNIT
Supply Voltage to Ground Potential	-0.5 to +6.0	V
Maximum current per pin excluding V_{DD} and V_{SS}	25	mA
Maximum current out of GND	100	mA
Maximum current out of VCC	100	mA
Ambient Operating Temperature	0 to +70	°C
Storage Temperature	-55 to +150	°C

Note: Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

DC Characteristics

(V_{DD}-V_{SS} = 5.0 V, F_{OSC} = 6MHz, Ta = 25° C; unless otherwise specified)

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Op. Voltage	V_{DD}	-	4.35	-	5.5	V
Op. Current	I_{OP}	No load (Ext.-V) In normal operation	-	?	20	mA
Suspend Current	I_{SUS}	No load (Ext.-V)	-	300	450	μA
Input High Voltage	V_{IH}	-	2	-	V_{DD}	V
Input Low Voltage	V_{IL}	-	0	-	0.8	V
Port 0, 1, 2, 3 drive current	I_{OH}	V_{OH} = 4.5V, V_{DD} = 5.0V	-	2.5	-	mA
Port 0.4~0.7, 1, 2, 3 sink current	I_{OL1}	V_{OL} = 0.4V, V_{DD} = 5.0V	-	4.0	-	mA
Port 0.0~0.3 sink current	I_{OL2}	V_{OL} = 3.2V, V_{DD} = 5.0V	6	8	-	mA
Internal Pull-high Resistor	R_{PH}	V_{IL} = 0V	-	27K	-	Ω

AC Characteristics

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
CPU Op. Frequency	F_{CPU}	V_{DD} = 5.0V	0.5	3	-	MHz
POR duration	T_{POR}	F_{OSC} = 6 MHz	10	?	?	mS

Instruction Set Summary

Symbol Description

ACC:	Accumulator
(ACC):	Contents of Accumulator
ACC.n:	Accumulator bit n
X:	Index Register X
Y:	Index Register Y
SP:	Stack Pointer Register
PC:	Program Counter
#data:	Constant parameter
C:	Carry Flag
Z:	Zero Flag
I:	Interrupt Disable Status
B:	Break Status
D:	Decimal Mode Status
V:	Overflow Flag
S:	Sign Flag
$addr_{16}$:	Absolute Address
$addr_{8}$:	Zero Page/Relative Address
$\underline{\text{addr+(index)}}$:	Combined Address
$\underline{\text{addr }}_{\rightarrow 16}$:	Address Extend to Absolute Address (Get two $addr_{8}$ contents continuously)
label:	Address Variable
~:	1’s compliment
∩:	AND
∪:	OR
⊕ :	Exclusive OR
←:	Transfer direction, result

Instruction Set Summary (212 instructions)

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
ADC	addr8	(ACC) ← (ACC) + (addr8) + (C)	C, Z, V,S	2	3
	#data	(ACC) ← (ACC) + #data + (C)	C, Z, V,S	2	2
	(addr8)	(ACC) ← (ACC) + [(addr8)] + (C)	C, Z, V,S	2	5
	addr8, X	(ACC) ← (ACC) + [addr8 + (X)] + (C)	C, Z, V,S	2	4
	(addr8, X)	(ACC) ← (ACC) + {[addr8 + (X) →16]} + (C)	C, Z, V,S	2	6
	(addr8), Y	(ACC) ← (ACC) + [(addr8→16) + (Y)] + (C)	C, Z, V,S	2	5*
	addr16	(ACC) ← (ACC) + (addr16) + (C)	C, Z, V,S	3	4
	addr16, X	(ACC) ← (ACC) + [addr16 + (X)] + (C)	C, Z, V,S	3	4*
	addr16, Y	(ACC) ← (ACC) + [addr16 + (Y)] + (C)	C, Z, V,S	3	4*
SBC	addr8	(ACC) ← (ACC) – (addr8) – (~C)	C, Z, V,S	2	3
	#data	(ACC) ← (ACC) – #data – (~C)	C, Z, V,S	2	2
	(addr8)	(ACC) ← (ACC) – [(addr8)] – (~C)	C, Z, V,S	2	5
	addr8, X	(ACC) ← (ACC) – [addr8 + (X)] – (~C)	C, Z, V,S	2	4
	(addr8, X)	(ACC) ← (ACC) – {[addr8 + (X) →16]} – (~C)	C, Z, V,S	2	6
	(addr8), Y	(ACC) ← (ACC) – [(addr8→16) + (Y)] – (~C)	C, Z, V,S	2	5*
	addr16	(ACC) ← (ACC) – (addr16) – (~C)	C, Z, V,S	3	4
	addr16, X	(ACC) ← (ACC) – [addr16 + (X)] – (~C)	C, Z, V,S	3	4*
	addr16, Y	(ACC) ← (ACC) – [addr16 + (Y)] – (~C)	C, Z, V,S	3	4*
AND	addr8	(ACC) ← (ACC) ∩ (addr8)	Z, S	2	3
	#data	(ACC) ← (ACC) ∩ #data	Z, S	2	2
	(addr8)	(ACC) ← (ACC) ∩ [(addr8)]	Z, S	2	5
	addr8, X	(ACC) ← (ACC) ∩ [addr8 + (X)]	Z, S	2	4
	(addr8, X)	(ACC) ← (ACC) ∩ {[addr8 + (X) →16]}	Z, S	2	6
	(addr8), Y	(ACC) ← (ACC) ∩ [(addr8→16) + (Y)]	Z, S	2	5*
	addr16	(ACC) ← (ACC) ∩ (addr16)	Z, S	3	4
	addr16, X	(ACC) ← (ACC) ∩ [addr16 + (X)]	Z, S	3	4*
	addr16, Y	(ACC) ← (ACC) ∩ [addr16 + (Y)]	Z, S	3	4*

Note: * Add one clock period of page boundary is crossed.

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
ORA	addr8	(ACC) ← (ACC) ∪ (addr8)	Z, S	2	3
	#data	(ACC) ← (ACC) ∪ #data	Z, S	2	2
	(addr8)	(ACC) ← (ACC) ∪ [(addr8)]	Z, S	2	5
	addr8, X	(ACC) ← (ACC) ∪ [addr8 + (X)]	Z, S	2	4
	(addr8, X)	(ACC) ← (ACC) ∪ {[addr8 + (X) →16]}	Z, S	2	6
	(addr8), Y	(ACC) ← (ACC) ∪ [(addr8→16) + (Y)]	Z, S	2	5*
	addr16	(ACC) ← (ACC) ∪ (addr16)	Z, S	3	4
	addr16, X	(ACC) ← (ACC) ∪ [addr16 + (X)]	Z, S	3	4*
	addr16, Y	(ACC) ← (ACC) ∪ [addr16 + (Y)]	Z, S	3	4*
EOR	addr8	(ACC) ← (ACC) ⊕ (addr8)	Z, S	2	3
	#data	(ACC) ← (ACC) ⊕ #data	Z, S	2	2
	(addr8)	(ACC) ← (ACC) ⊕ [(addr8)]	Z, S	2	5
	addr8, X	(ACC) ← (ACC) ⊕ [addr8 + (X)]	Z, S	2	4
	(addr8, X)	(ACC) ← (ACC) ⊕ {[addr8 + (X) →16]}	Z, S	2	6
	(addr8), Y	(ACC) ← (ACC) ⊕ [(addr8→16) + (Y)]	Z, S	2	5*
	addr16	(ACC) ← (ACC) ⊕ (addr16)	Z, S	3	4
	addr16, X	(ACC) ← (ACC) ⊕ [addr16 + (X)]	Z, S	3	4*
	addr16, Y	(ACC) ← (ACC) ⊕ [addr16 + (Y)]	Z, S	3	4*
CMP	addr8	(ACC) – (addr8) – (~C)	C, Z, S	2	3
	#data	(ACC) – #data – (~C)	C, Z, S	2	2
	(addr8)	(ACC) – [(addr8)] – (~C)	C, Z, S	2	5
	addr8, X	(ACC) – [addr8 + (X)] – (~C)	C, Z, S	2	3
	(addr8, X)	(ACC) – {[addr8 + (X) →16]} – (~C)	C, Z, S	2	6
	(addr8), Y	(ACC) – [(addr8→16) + (Y)] – (~C)	C, Z, S	2	5*
	addr16	(ACC) – (addr16) – (~C)	C, Z, S	3	4
	addr16, X	(ACC) – [addr16 + (X)] – (~C)	C, Z, S	3	4*
	addr16, Y	(ACC) – [addr16 + (Y)] – (~C)	C, Z, S	3	4*
CPX	#data	(X) – #data	C, Z, S	2	2
	addr8	(X) – (addr8)	C, Z, S	2	3
	addr16	(X) – (addr16)	C, Z, S	3	4
CPY	#data	(Y) – #data	C, Z, S	2	2
	addr8	(Y) – (addr8)	C, Z, S	2	3
	addr16	(Y) – (addr16)	C, Z, S	3	4

Note: * Add one clock period of page boundary is crossed.

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
CLC		(C) ← 0	C	1	2
CLI		(I) ← 0	I	1	2
CLD		(D) ← 0	D	1	2
CLV		(V) ← 0	V	1	2
RMB0*	addr8	(addr8.0) ← 0	Z	2	5
…					
RMB7	addr8	(addr8.7) ← 0	Z	2	5
SEC		(C) ← 1	C	1	2
SEI		(I) ← 1	I	1	2
SED		(D) ← 1	D	1	2
SMB0*	addr8	(addr8.0) ← 1	Z	2	5
…					
SMB7	addr8	(addr8.7) ← 1	Z	2	5
INC	A	(ACC) ← (ACC) + 1	C, Z	1	2
INC	addr8	(addr8) ← (addr8) + 1	Z, S	2	5
	addr8, X	[addr8 + (X)] ← [addr8 + (X)] + 1	Z, S	2	6
	addr16	(addr16) ← (addr16) + 1	Z, S	3	6
	addr16, X	[addr16 + (X)] ← [addr16 + (X)] + 1	Z, S	3	6*
INX		(X) ← (X) + 1	Z, S	1	2
INY		(Y) ← (Y) + 1	Z, S	1	2
DEC	A	(ACC) ← (ACC) – 1	C, Z	1	2
DEC	addr8	(addr8) ← (addr8) – 1	Z, S	2	5
	addr8, X	[addr8 + (X)] ← [addr8 + (X)] – 1	Z, S	2	6
	addr16	(addr16) ← (addr16) – 1	Z, S	3	6
	addr16, X	[addr16 + (X)] ← [addr16 + (X)] – 1	Z, S	3	6*
DEX		(X) ← (X) – 1	Z, S	1	2
DEY		(Y) ← (Y) – 1	Z, S	1	2

Note: * Add one clock period of page boundary is crossed.

* If the assembler does not support this instruction, please use DB to implement it. The OP code of RMB0 ~ RMB7 is 07 ~ 77, and the SMB0 ~ SMB7 is 87 ~ F7.

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
ROL	A	(C) ← (ACC.7), (ACC.(n+1)) ← (ACC. n), (ACC.0) ← (C)	C, Z, S	1	2
ROL	addr8	(C) ← (addr8.7), (addr8.(n+1)) ← (addr8.n), (addr8.0) ← (C)	C, Z, S	2	5
	addr8, X	(C) ← [addr8 + (X).7], [addr8 + (X).(n+1)] ← [addr8 + (X).n], [addr8 + (X).0] ← (C)	C, Z, S	2	6
	addr16	(C) ← (addr16.7), (addr16.(n+1))← (addr16.n), (addr16.0) ← (C)	C, Z, S	3	6
	addr16, X	(C) ← [addr16 + (X).7], [addr16 + (X).(n+1)] ← [addr16 + (X).n], [addr16 + (X).0] ← (C)	C, Z, S	3	6
ROR	A	(ACC.7) ← (C), (ACC. n) ← (ACC.(n+1)), (C) ← (ACC.0)	C, Z, S	1	2
ROR	addr8	(addr8.7) ← (C), (addr8. n) ← (addr8.(n+1)), (C) ← (addr8.0)	C, Z, S	2	5
	addr8, X	[addr8 + (X).7] ← (C), [addr8 + (X).n] ← [addr8 + (X).(n+1)], (C) ← [addr8 + (X).0]	C, Z, S	2	6
	addr16	(addr16.7) ← (C), (addr16. n) ← (addr16.(n+1)), (C) ← (addr16.0)	C, Z, S	3	6
	addr16, X	[addr16 + (X).7] ← (C), [addr16 + (X).n] ← [addr16 + (X).(n+1)], (C) ← [addr16 + (X).0]	C, Z, S	3	6
ASL	A	(C) ← (ACC.7), (ACC.(n+1)) ← (ACC. n), (ACC.0) ← 0	C, Z, S	1	2
ASL	addr8	(C) ← (addr8.7), (addr8.(n+1)) ← (addr8. n), (addr8.0) ← 0	C, Z, S	2	5
	addr8, X	(C) ← [addr8 + (X).7], [addr8 + (X).(n+1)] ← [addr8 + (X).n], [addr8 + (X).0] ← 0	C, Z, S	2	6
	addr16	(C) ← (ACC.7), (ACC.(n+1)) ← (ACC. n), (ACC.0) ← 0	C, Z, S	3	6
	addr16, X	(C) ← [addr16 + (X).7], [addr16 + (X).(n+1)] ← [addr16 + (X).n], [addr16 + (X).0] ← 0	C, Z, S	3	6
LSR	A	(ACC.7) ← 0, (ACC. n) ← (ACC.(n+1)), (C) ← (ACC.0)	C, Z, S	1	2
LSR	addr8	(addr8.7) ← 0, (addr8. n) ← (addr8.(n+1)), (C) ← (addr8.0)	C, Z, S	2	5
	addr8, X	[addr8 + (X).7] ← 0, [addr8 + (X).n] ← [addr8 + (X).(n+1)], (C) ← [addr8 + (X).0]	C, Z, S	2	6
	addr16	(addr16.7) ← 0, (addr16. n) ← (addr16.(n+1)), (C) ← (addr16.0)	C, Z, S	3	6
	addr16, X	[addr16 + (X).7] ← 0, [addr16 + (X).n] ← [addr16 + (X).(n+1)], (C) ← [addr16 + (X).0]	C, Z, S	3	6

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
LDA	#data	(ACC) ← #data	Z, S	2	2
LDA	addr$_8$	(ACC) ← (addr$_8$)	Z, S	2	3
	(addr$_8$)	(ACC) ← [(addr$_8$)]	Z, S	2	5
	addr$_8$, X	(ACC) ← [addr$_8$ + (X)]	Z, S	2	4
	(addr$_8$, X)	(ACC) ← {[$\underline{\text{addr}_8 + (X)}$ $\rightarrow_{16}$]}	Z, S	2	6
	(addr$_8$), Y	(ACC) ← [($\underline{\text{addr}_8}$$\rightarrow_{16}$) + (Y)]	Z, S	2	6*
	addr$_{16}$	(ACC) ← (addr$_{16}$)	Z, S	3	4
	addr$_{16}$, X	(ACC) ← [addr$_{16}$ + (X)]	Z, S	3	4*
	addr$_{16}$, Y	(ACC) ← [addr$_{16}$ + (Y)]	Z, S	3	4*
LDX	#data	(X) ← #data	Z, S	2	2
	addr$_8$	(X) ← (addr$_8$)	Z, S	2	3
	addr$_8$, Y	(X) ← [addr$_8$ + (Y)]	Z, S	2	4
	addr$_{16}$	(X) ← (addr$_{16}$)	Z, S	3	4
	addr$_{16}$, Y	(X) ← [addr$_{16}$ + (Y)]	Z, S	3	4*
LDY	#data	(Y) ← #data	Z, S	2	2
	addr$_8$	(Y) ← (addr$_8$)	Z, S	2	3
	addr$_8$, X	(Y) ← [addr$_8$ + (X)]	Z, S	2	4
	addr$_{16}$	(Y) ← (addr$_{16}$)	Z, S	3	4
	addr$_{16}$, X	(Y) ← [addr$_{16}$ + (X)]	Z, S	3	4*

Note: * Add one clock period of page boundary is crossed.

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
STA	addr8	(addr8) ← (ACC)	-	2	3
	(addr8)	[(addr8)] ← (ACC)	-	2	5
	addr8, X	[addr8 + (X)] ← (ACC)	-	2	4
	(addr8, X)	{[addr8 + (X) →16]} ← (ACC)	-	2	6
	(addr8), Y	[(addr8→16) + (Y)] ← (ACC)	-	2	6*
	addr16	(addr16) ← (ACC)	-	3	4
	addr16, X	[addr16 + (X)] ← (ACC)	-	3	4*
	addr16, Y	[addr16 + (Y)] ← (ACC)	-	3	4*
STX	addr8	(addr8) ← (X)	-	2	3
	addr8, Y	[addr8 + (Y)] ← (X)	-	2	4
	addr16	(addr16) ← (X)	-	3	4
STY	addr8	(addr8) ← (Y)	-	2	3
	addr8, X	[addr8 + (X)] ← (Y)	-	2	4
	addr16	(addr16) ← (Y)	-	3	4
STZ	addr8	(addr8) ← 00H	-	2	3
	addr8, X	[addr8 + (X)] ← 00H	-	2	4
	addr16	(addr16) ← 00H	-	3	4
	addr16, X	[addr16 + (X)] ← 00H	-	3	5
TAX		(X) ← (ACC)	Z, S	1	2
TXA		(ACC) ← (X)	Z, S	1	2
TAY		(Y) ← (ACC)	Z, S	1	2
TYA		(ACC) ← (Y)	Z, S	1	2
TSX		(X) ← (SP)	Z, S	1	2
TXS		(SP) ← (X)	-	1	2
TRB	addr8	(addr8) ← (~ACC) ∩ (addr8)	-	2	5
	addr16	(addr16) ← (~ACC) ∩ (addr16)	-	3	6
TSB	addr8	(addr8) ← (ACC) ∪ (addr8)	-	2	5
	addr16	(addr16) ← (ACC) ∪ (addr16)	-	3	6

Note: * Add one clock period of page boundary is crossed.

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
JMP	label	(PC) ← label; the label may be address or variable.	-	3	3
	(label)	(PC) ← (label)	-	3	6
	(label, X)	(PC) ← {[label + (X) →16]}	-	3	6
BRA	addr8	(PC) ← (PC)+addr8	-	2	3
BEQ	addr8 (relative)	(PC) ← (PC)+addr8 if Z == 1 (+/- relative)	-	2	2**
BNE	addr8	(PC) ← (PC)+addr8 if Z == 0 (+/- relative)	-	2	2**
BCS	addr8	(PC) ← (PC)+addr8 if C == 1 (+/- relative)	-	2	2**
BCC	addr8	(PC) ← (PC)+addr8 if C == 0 (+/- relative)	-	2	2**
BMI	addr8	(PC) ← (PC)+addr8 if (S == 1)	-	2	2**
BPL	addr8	(PC) ← (PC)+addr8 if (S == 0)	-	2	2**
BVS	addr8	(PC) ← (PC)+addr8 if (V == 1)	-	2	2**
BVC	addr8	(PC) ← (PC)+addr8 if (V == 0)	-	2	2**
BIT	addr8	(ACC) ∩ (addr8)	Z	2	3
	addr8, X	(ACC) ∩ [addr8 + (X)]	Z	2	4
	addr16	(ACC) ∩ (addr16)	Z	3	4
	addr16, X	(ACC) ∩ [addr16 + (X)]	Z	3	4
	#data	(ACC) ∩ #data	Z	2	2
BBR0*	addr8	(PC) ← (PC)+addr8 if ACC.0 == 0 (+/- relative)	-	3	5
…					
BBR7	addr8	(PC) ← (PC)+addr8 if ACC.7 == 0 (+/- relative)	-	3	5
BBS0*	addr8	(PC) ← (PC)+addr8 if ACC.0 == 1 (+/- relative)	-	3	5
…					
BBS7	addr8	(PC) ← (PC)+addr8 if ACC.7 == 1 (+/- relative)	-	3	5

Note: ** Add one clock period if branch occurs to location in same page. Add two clock periods if branch to another page occurs.

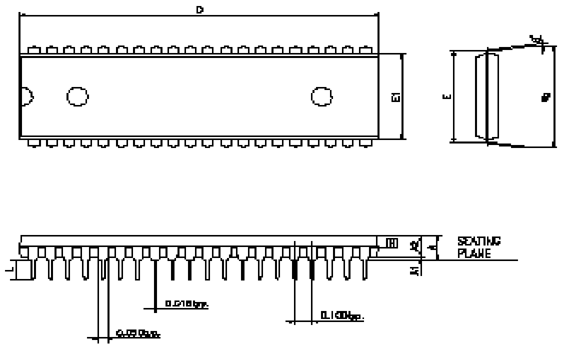
* If the assembler does not support this instruction, please use DB to implement it. The OP code of BBR0 ~ BBR7 is 0F ~ 7F, and the BBS0 ~ BBS7 is 8F ~ FF.

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
JSR	label	stack ← (PC), (PC) ← label	-	3	6
RTS		(PC) ← pop stack	-	1	6
RTI		(PC) ← pop stack, restore status register P	C, Z, I, D, V, S	1	6
PHA		[(SP)] ← (ACC), (SP) ← (SP) – 1	-	1	3
PHP		[(SP)] ← (P), (SP) ← (SP) – 1	-	1	3
PHX		[(SP)] ← (X), (SP) ← (SP) – 1	-	1	3
PHY		[(SP)] ← (Y), (SP) ← (SP) – 1	-	1	3
PLA		(ACC) ← [(SP+1)], (SP) ← (SP) + 1	Z, S	1	4
PLP		(P) ← [(SP+1)], (SP) ← (SP) + 1	C, Z, I, D, V, S	1	4
PLX		(X) ← [(SP+1)], (SP) ← (SP) + 1	Z, S	1	4
PLY		(Y) ← [(SP+1)], (SP) ← (SP) + 1	Z, S	1	4
STP		CPU stop, PHI2 ← 1	-	1	3
WAI		CPU waiting, RDY ← 0	-	1	3
BRK		(SP) ← (PCH), (SP-1) ← (PCL), (SP-2) ← (P), (SP) ← (SP-3), (PCH) ← #$FFFF, (PCL) ← #$FFFE, I ← 1, B ← 1 ; (for ICE step trace)	-	2	7
NOP		No operation	-	1	2

Note: ** Add one clock period if branch occurs to location in same page. Add two clock periods if branch to another page occurs.

Package Dimensions

40-pin DIP

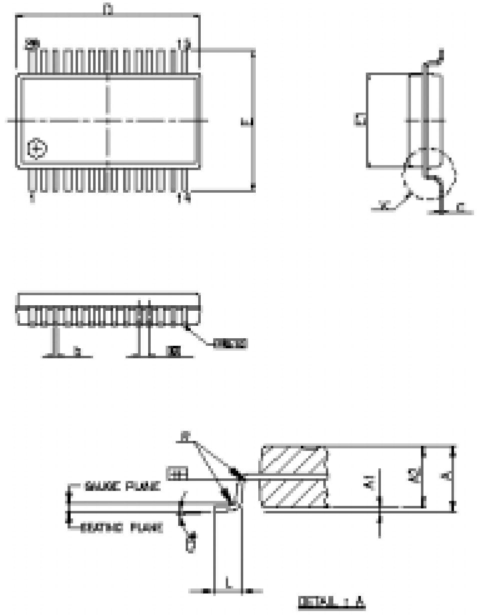


SYMBOLS	MIN.	NOR.	MAX.
A	–	–	0.220
A1	0.015	–	–
A2	0.150	0.155	0.160
D	2.055	2.060	2.070
E		0.600 BSC	
E1	0.540	0.545	0.550
L	0.115	0.130	0.150
eB	0.630	0.650	0.670
θ	0	7	15

UNIT : INCH

NOTE:
1.JEDEC OUTLINE:N/A

28-SSOP



SYMBOLS	MIN.	NOM.	MAX.
A	–	–	2.0
A1	0.05	–	–
A2	1.62	1.75	1.85
b	0.22	–	0.38
c	0.09	–	0.25
D	9.90	10.20	10.50
E	7.40	7.80	8.20
E1	5.00	5.30	5.60
e		0.65 BSC	
L	0.55	0.75	0.95
R	0.09	–	–
θ	0°	4°	8°

UNIT : MM

NOTES :

1.JEDEC OUTLINE : MO-150 AH

2."D" AND "E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS, BUT DOES INCLUDE MOLD MISMATCH AND ARE MEASURED AT DATUM PLANE. INMOLD PARTING LINE. MOLD FLASH OR PROTRUSION SHALL NOT EXCEED 0.15 mm PER SIDE.

3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION/INTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13mm TOTAL IN EXCESS OF b DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR INTRUSION SHALL NOT REDUCE DIMENSION b BY MORE THAN 0.07mm AT LEAST MATERIAL CONDITION.

Version History

Version	Date	Page	Description
0.1	2003/9		Initial document
1.0	2004/8	1,44	Revised the operating voltage from 4.35V to 5.5V
1.1	2005/1	41	Application circuit has been modified.
A2	2005/7	19~34	Revised USB SFR Description.