General Description 3
Features 4
Pin Description 5
Block Diagram 6
Packages 7
Function Description 8
Memory Map 9
Special Function Register (SFR) 10
Interrupt Vectors 11
    Interrupt Registers 12
        IRQ enable flag 12
        IRQ status flag 12
        IRQ clear flag 12
Watchdog Timer (WDT) 13
System Control Registers 14
    Power saving control 14
    Release halt mode enable flag 14
FCPU selector 15
Timer 16
    Timer 0 16
USB 17
    USB Block Diagram 17
    USB register access control 17
    USB SFR Category - Descriptions Summary 18
    USB SFR Description 19
        DCON: Device Control Register 19
        FADDR: USB Function Address Register 19
        FPCON: Function Power Control Register 20
        FIE: Function Interrupt Enable Register 20
        FIFLG: Function Interrupt Flag Register 21
        IEN1: USB Interrupt Enable Register 21
        EPINDEX: Endpoint Index Register 22
        EPCON: Endpoint Control Register (Endpoint-Indexed) 22
        RXCNT: Receive FIFO Byte-Count Register (Endpoint-Indexed) 23
        RXCON: Receive FIFO Control Register (Endpoint-Indexed) 23
        RXDAT: Receive FIFO Data Register (Endpoint-Indexed) 24
        RXSTAT: Endpoint Receive Status Register (Endpoint-Indexed) 24
        TXCNT: Transmit FIFO Byte-Count Register (Endpoint-Indexed) 25
        TXCON: Transmit FIFO Control Register (Endpoint-Indexed) 25
        TXDAT: Transmit FIFO Data Register (Endpoint-Indexed) 26
        TXSTAT: Endpoint Transmit Status Register (Endpoint-Indexed) 26
I/O Ports 27
    Port 0 27
    Port 1 27
    Port 2 28
    Port 3 28
    Port 4 28
    DPM control 29
Programming Notice 30
Application Circuit 31
    Normal: 31

USB keyboard: 31
Pad Assignment 32
MPC235 32
Absolute Maximum Rating 34
DC Characteristics 35
AC Characteristics 36
Instruction Set Summary 37
Addressing Mode Table 37
Instruction Set Table 38
Instruction Set Summary 39
Symbol Description 41
Arithmetic Operations 42
Logic Operations 44
Data Transfer 46
Boolean Variable Manipulation 47
Program and Machine Control 48
Package Dimensions 49
40-pin DIP 49
28-SSOP 49
Version History 50

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 meet 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

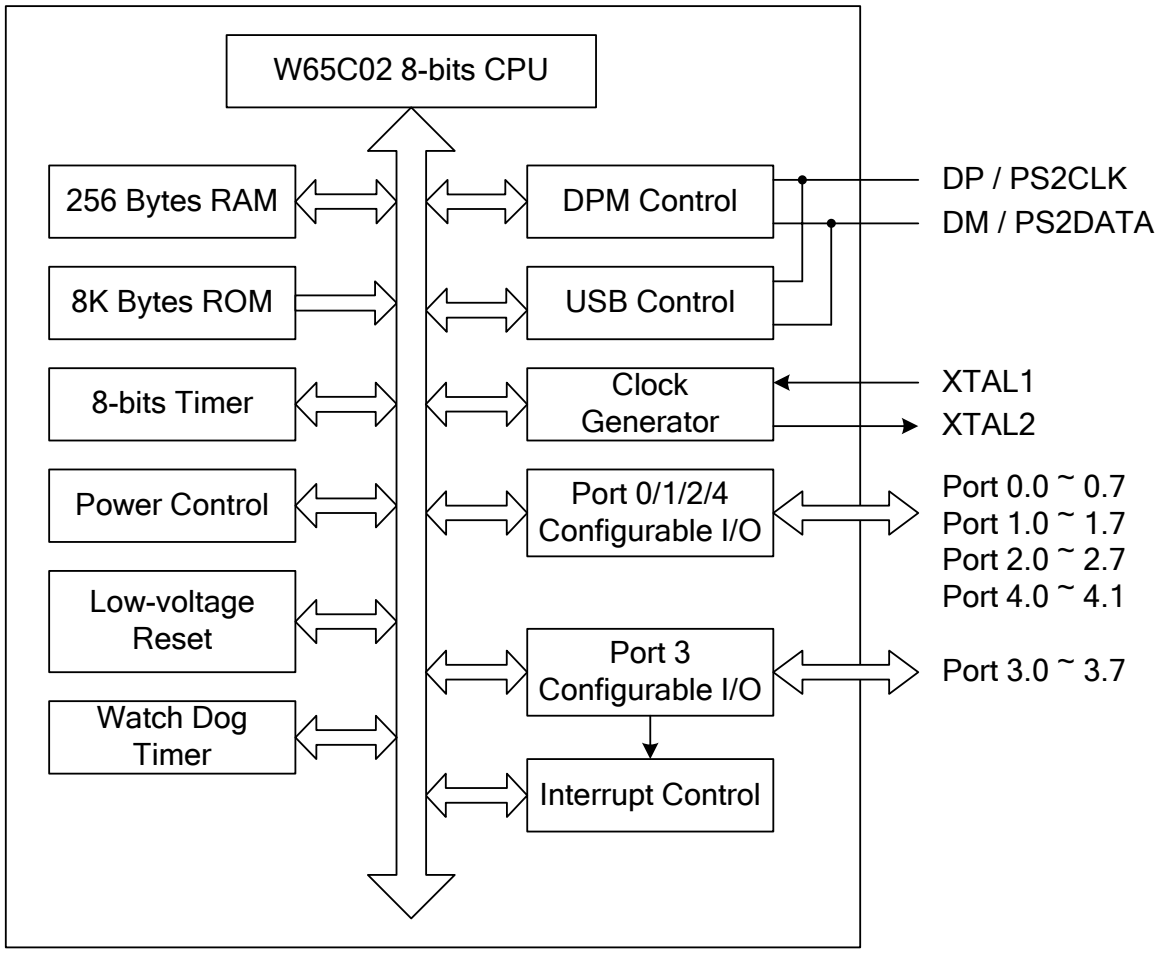
- The MPC235 is mask version of the MPC2F35
- 8-bit 65C02 micro-controller with 6MHz external crystal or ceramic resonator
- Memory:
  - 8K Bytes ROM
  - 256 Bytes RAM
- 34 programmable GPIO:
  - 4 LED direct sink pins shared with Port0 (LED0/1/2/3)
  - 2 external interrupt pins (-INT0, -INT1)
  - Port3 provides the pin interrupt
  - 34 bi-directional I/O pins for Port0/1/ 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 USB built-in pull-up resistor.
  - Provides remote-wake-up/host-resume from suspend mode.
  - Built-in 5V to 3.3V regulator for USB.
- Built-in low-voltage detector (reset)
- USB and PS/2 combo interfaces
- Support suspend/normal mode for power management
- Operating voltage: 4.35V to 5.5V
- Operating temperature: 0°C to 70°C
- Packages:
  - 28-SSOP: MPC235L
  - 40-PDIP: MPC235E2

Pin Description

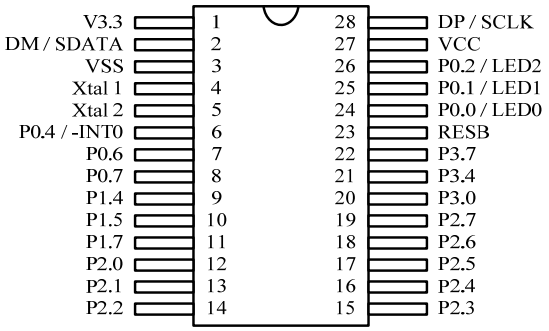
PIN Name	I/O	Description
P0.0~0.3 *1	I/O	Bi-directional I/O (sink LED directly)
P0.4/-INT0	I/O	Bi-directional I/O with external interrupt 1
P0.5/-INT1	I/O	Bi-directional I/O with external interrupt 2
P0.6~0.7	I/O	Bi-directional I/O
P1.0~1.7	I/O	Bi-directional I/O
P2.0~2.7	I/O	Bi-directional I/O
P3.0~3.7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

*1 P0.0~0.3 can be used to sink LED directly (without any external resistor).
*2 P4.0~P4.1 only available on Die form.

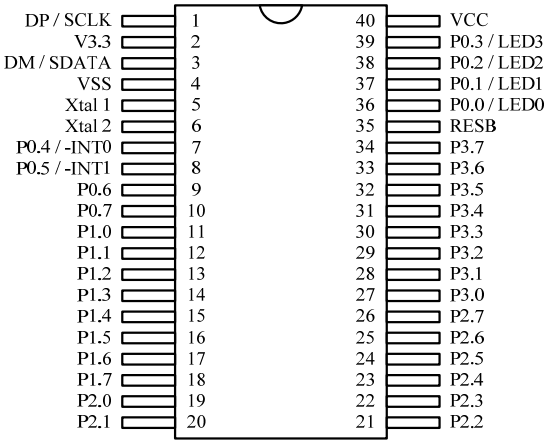
Block Diagram



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)

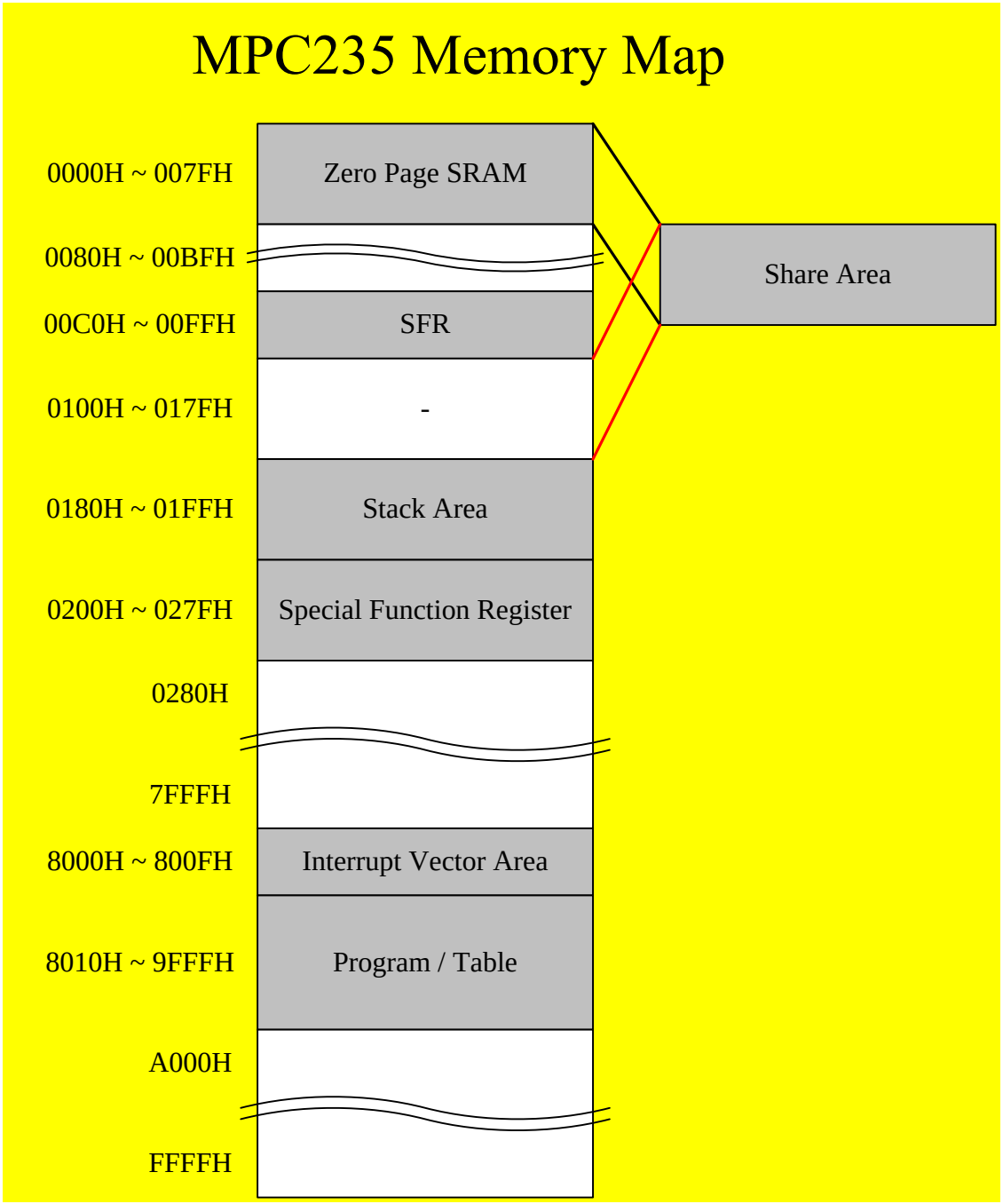
The 16-bit program counter register provides the addresses, which step the micro-controller through sequential program instructions. Each time the micro-controller fetch an instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order 8 bits of the address bus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

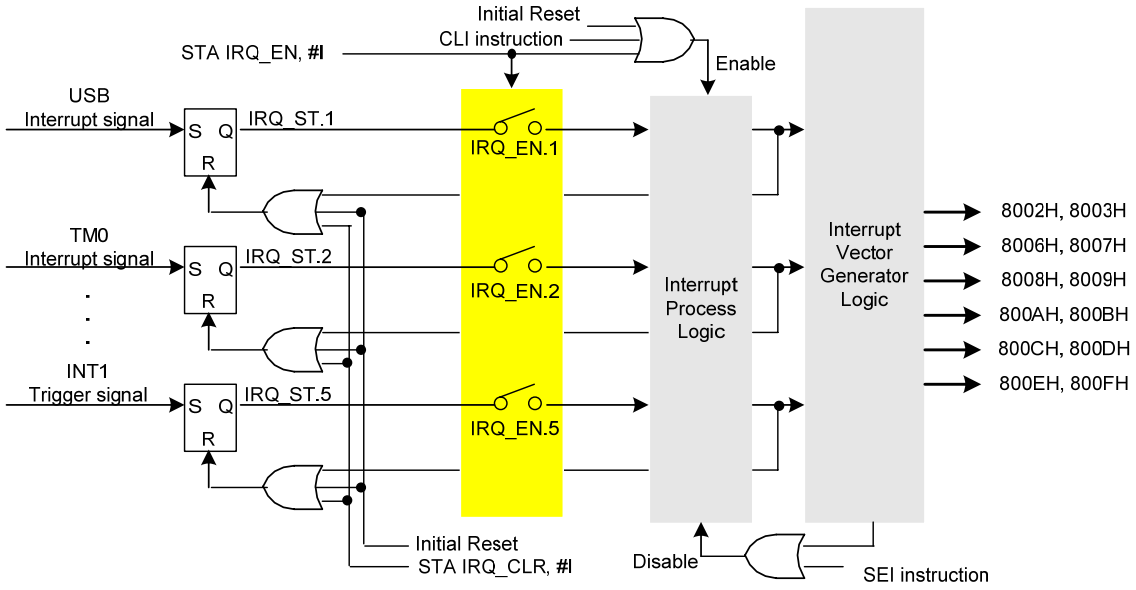
The address 00C0H to 00FFH and 0200H to 027FH are reserved for special function registers (SFR). The SFR is used to control or store the status of I/O, timers, system clock and other peripheral.

Symbol	Address	Description	Initial Value
IRQ_EN	00C1	Interrupt Request Enable	X
IRQ_ST	00C2	Interrupt Request Status Flag	00
IRQ_CLR	00C3	Interrupt Request Clear	00
TM0	00C5	Timer 0	00
TM0_CTL	00C6	Timer 0 Control	00
P0_BUF	00D0	Port 0 Output Buffer	00
P1_BUF	00D1	Port 1 Output Buffer	00
P2_BUF	00D2	Port 2 Output Buffer	00
P3_BUF	00D3	Port 3 Output Buffer	00
P4_BUF	00D4	Port 4 Output Buffer	00
P0	00D8	Port 0 Pad Value	X
P1	00D9	Port 1 Pad Value	X
P2	00DA	Port 2 Pad Value	X
P3	00DB	Port 3 Pad Value	X
P4	00DC	Port 4 Pad Value	X
WDT_ST	00DE	Watch Dog Timer Status Flag	00
WDT_CLR	00DF	Watch Dog Timer Status Clear	X
USB_CTL	00E0	USB Control	00
USB_ADDR	00E1	USB Register Address	00
USB_DI / USB_DO	00E2	USB Register Data Buffer	00
DPM_CTL	00E8	USB Bus Mode Control	00
DPMO	00E9	USB Bus Output for the PS/2 Mode	00
DPMI	00EA	USB Bus Output for the PS/2 Mode	X
PWR_CTL	0200	Power-Saving Control	00
FCPU_SR	0201	FCPU Selector	00
RLH_EN	0202	Release Halt Mode Enable	00
P0_CR	0240	Port 0 Control Register	00
P0_MR	0241	Port 0 Mode Register	00
P1_CR	0244	Port 1 Control Register	00
P1_MR	0245	Port 1 Mode Register	00
P2_CR	0248	Port 2 Control Register	00
P2_MR	0249	Port 2 Mode Register	00
P3_CR	024C	Port 3 Control Register	00
P3_MR	024D	Port 3 Mode Register	00
P4_CR	0250	Port 4 Control Register	00
P4_MR	0251	Port 4 Mode Register	00

Interrupt Vectors

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

There are six 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_CLR, #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 event

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)

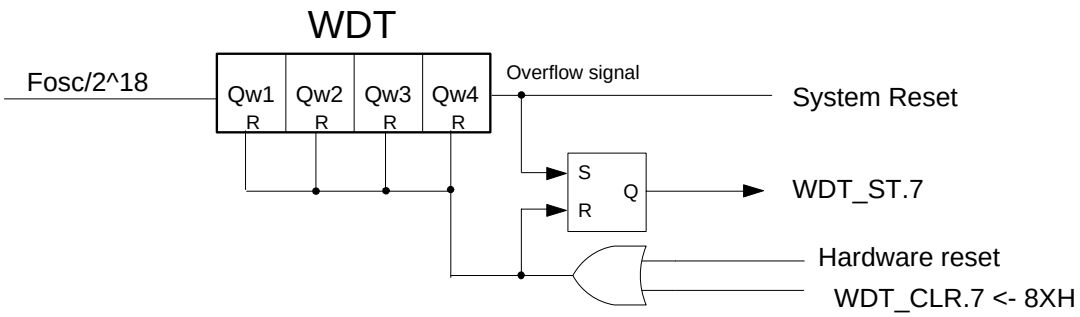
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W
00DEH	WDT_ST	RSTS	-	-	-	WDT 3	WDT 2	WDT 1	WDT 0	√	-
00DFH	WDT_CLR	CLR	-	-	-	-	-	-	-	-	√

WDT〔3:0〕: 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	0	-	-	-	-	-	CKC	HALT	-	√

Write "0" to Bit 7

When the low-voltage detector is enabled, and it senses the power voltage is lower than V_{LVR}, 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 5 ways: port 3 interrupt, hardware reset, power-on reset, USB Host Reset and USB Host Resume.

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.

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.

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: FCPU clock source select. 0: FOSC/2, 1: FOSC

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:

FTM0_UV = FTM0CK / (TM0+1)
where FTM0CK is selected by TM0_CTL.TKI2/11/10

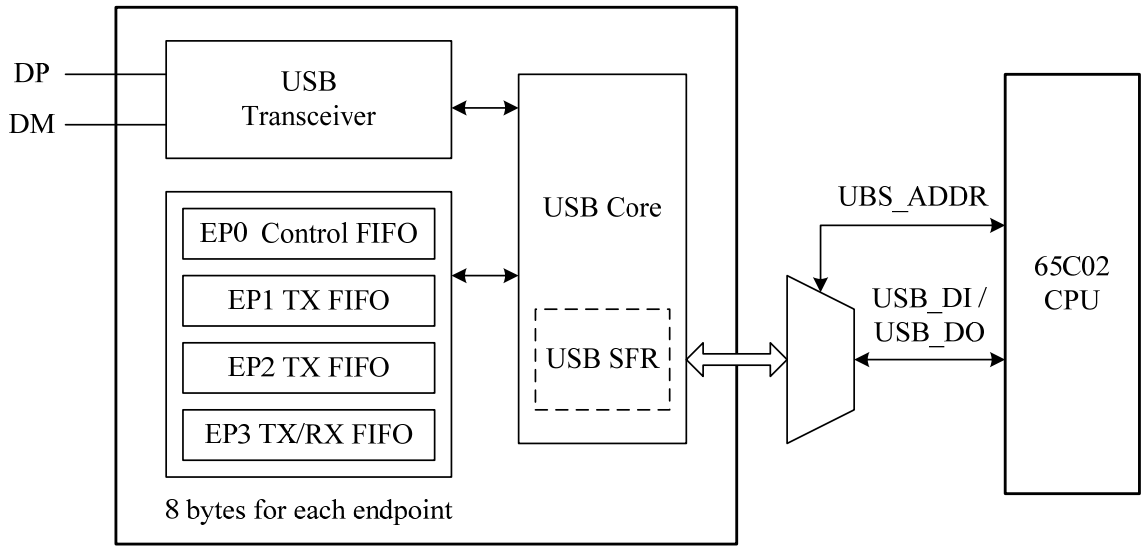
For example: if FTM0CK= FOSC / 32 (187.5 KHz)

TM1	FTM0_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 Block Diagram



EP0: Only for Control Transfer
EP1、EP2: Only Support Interrupt IN
EP3: Support Interrupt IN and Interrupt OUT

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:

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

Read sequence:

Write the address of USB register to be accessed into USB_ADDR
Write 1 into USB_CTL.URD
Read data from USB_DO
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 event from host, it will reset itself and generate an interrupt. When USB engine received a resume event 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 firmware com trigger a remote wakeup event from usb engine to host.

USB SFR Category - Descriptions Summary

Mnemonic	USB Device SFRs	Address	Description							
DCON	Device Control Register	01H	–	–	–	–	–	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_0X00 System 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	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.

Note: MPC2F35 do not have EP3DIR, Endpoint 3 is dedicated RX FIFO.

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_0X00 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 (write zero) 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 (write zero) 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 (write zero) 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.
3:0	RXBC[3:0]	Store receives byte count. Maximum is 8 bytes. Receive Byte Count

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 (write one) 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 RXSOVW 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 by framware to write “0”.
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 (write zero) after 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.

Note: When RXSETUP is set by hardware, device will return **NAK** when receive a IN or OUT Token from USB Host.

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.
3:0	TXBC[3:0]	Store transmits byte count. Maximum is 8 bytes. Transmit Byte Count

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.

Note: Before write “1” to TXCLR, firmware must write “1” to RXFFRC.

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 27kohm 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_EN.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	√	√

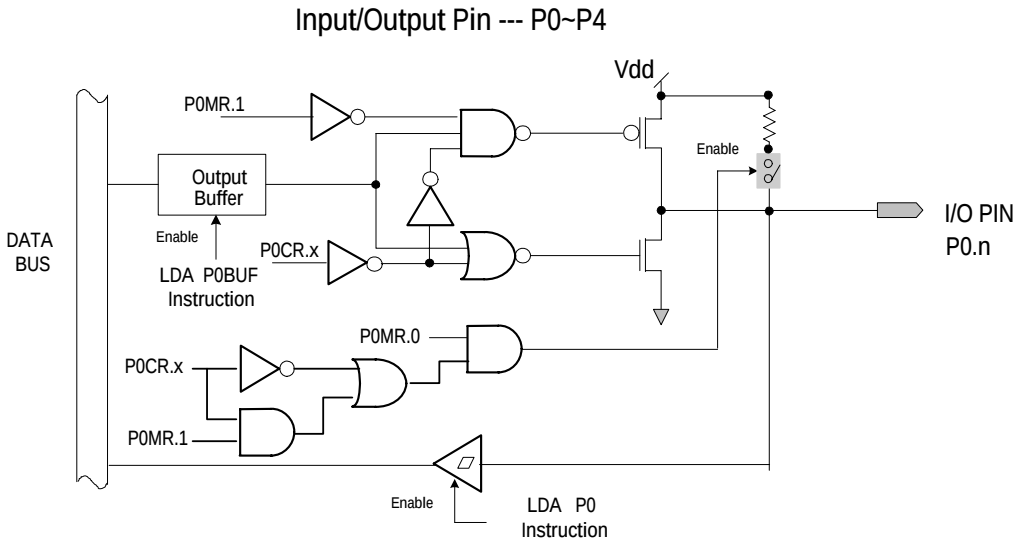
Port 4 is a 2-bit I/O port; refer to port 0 for more information.

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



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

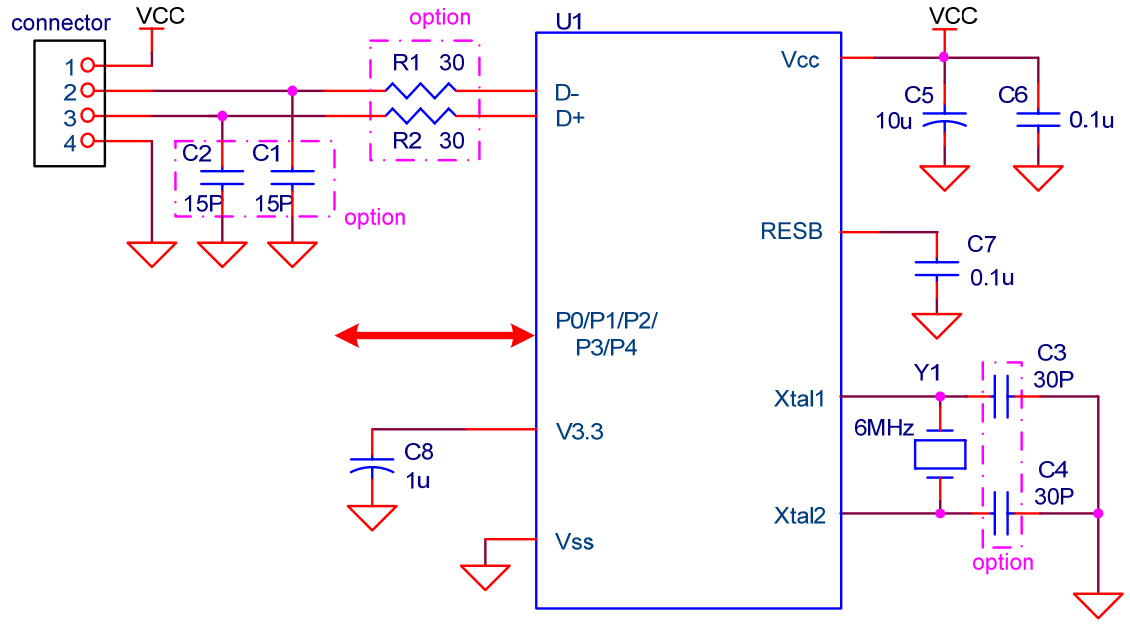
The status after different reset condition is listed below:

	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

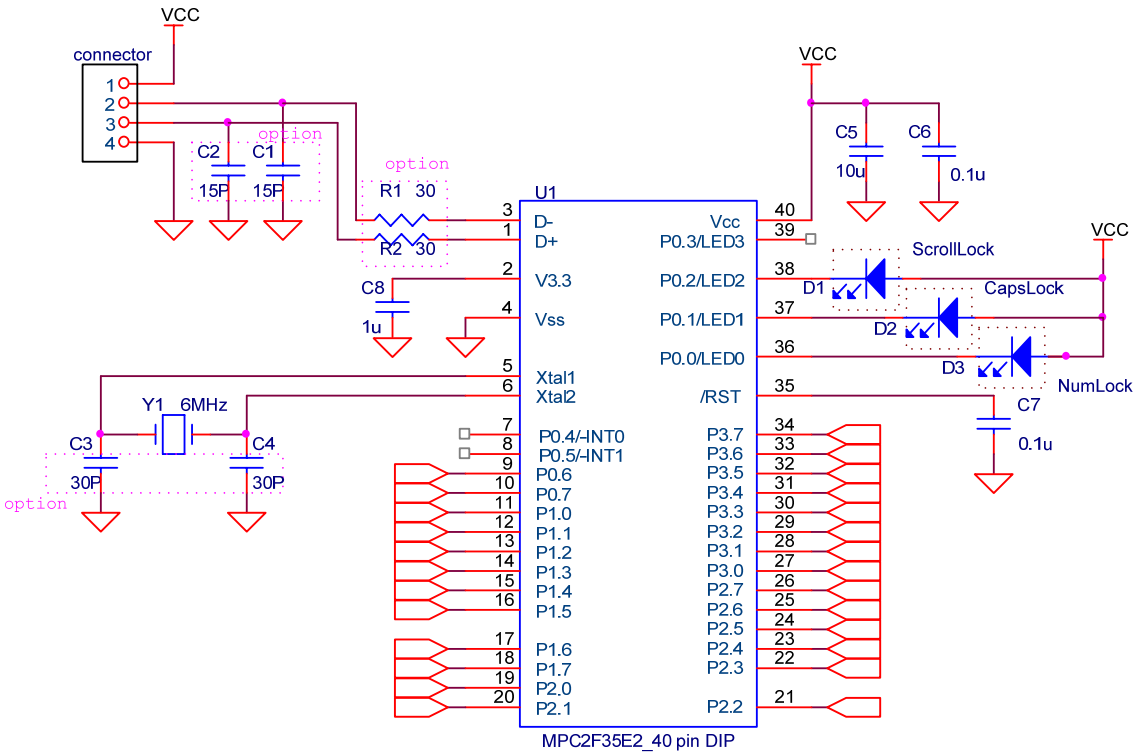
Application Circuit

Normal:



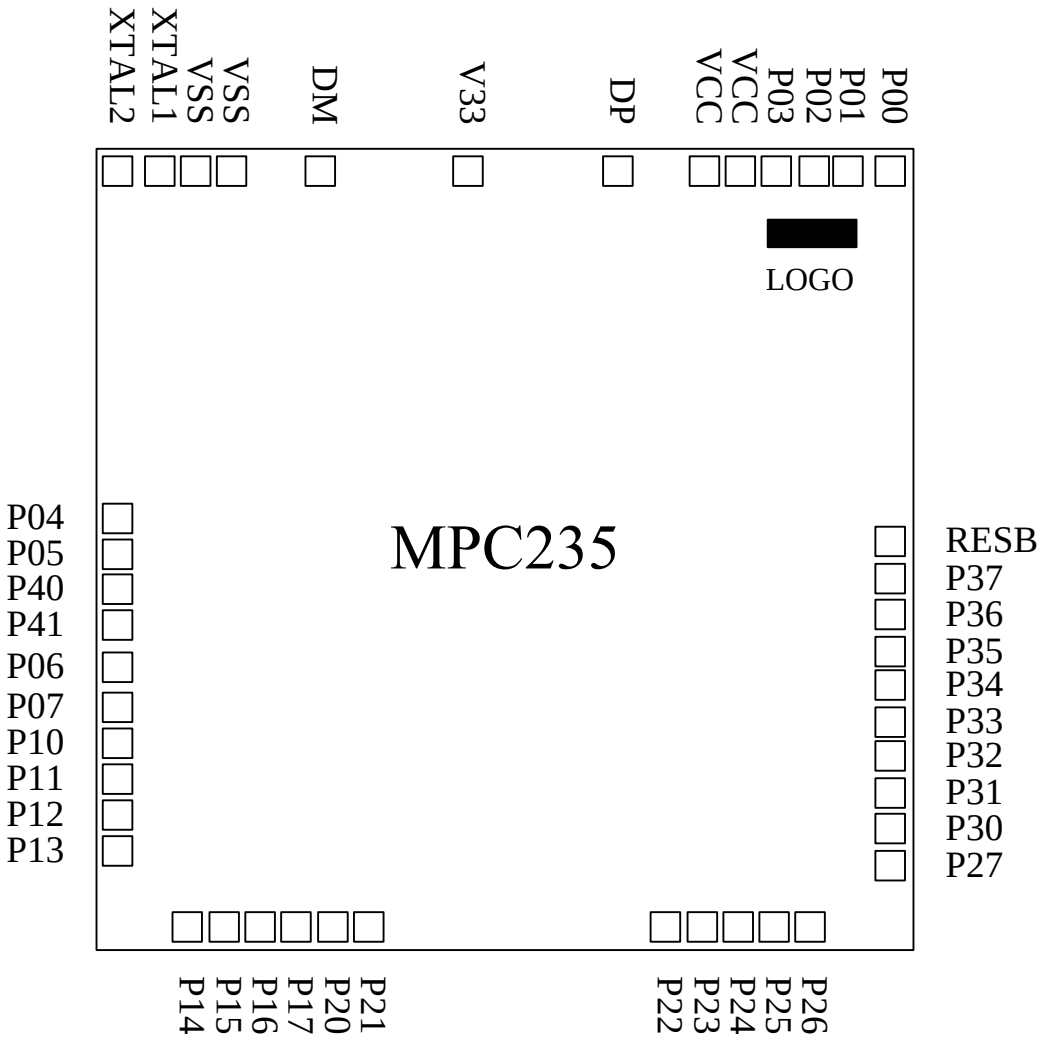
Note: The capacitor between RESB-pin and ground must be below 0.1uF.

USB keyboard:



Pad Assignment

MPC235



The substrate should be connected to Vss

MPC235 Pad Assignment

Pad	Name	X	Y	Pad	Name	X	Y
1	P00	1177.4	1167.9	23	P13	-1178.8	-931.5
2	P01	1047.4	1167.9	24	P14	-961.4	-1170.8
3	P02	937.4	1167.9	25	P15	-851.4	-1170.8
4	P03	827.4	1167.9	26	P16	-741.2	-1170.8
5	VDD	717.4	1167.9	27	P17	-631.2	-1170.8
6	VDD	607.4	1167.9	28	P20	-521	-1170.8
7	DP	344	1167.9	29	P21	-411	-1170.8
8	V33	-110.7	1167.9	30	P22	489.45	-1170.8
9	DM	-565.4	1167.9	31	P23	599.45	-1170.8
10	VSS	-828.8	1167.9	32	P24	709.65	-1170.8
11	VSS	-938.8	1167.9	33	P25	819.65	-1170.8
12	XTAL1	-1048.8	1167.9	34	P26	929.85	-1170.8
13	XTAL2	-1178.8	1167.9	35	P27	1177.4	-970.65
14	P04	-1178.8	96.3	36	P30	1177.4	-860.65
15	P05	-1178.8	-13.7	37	P31	1177.4	-750.45
16	P40	-1178.8	-123.9	38	P32	1177.4	-640.45
17	P41	-1178.8	-233.9	39	P33	1177.4	-530.25
18	P06	-1178.8	-362.95	40	P34	1177.4	-420.25
19	P07	-1178.8	-491.1	41	P35	1177.4	-310.05
20	P10	-1178.8	-601.3	42	P36	1177.4	-200.05
21	P11	-1178.8	-711.3	43	P37	1177.4	-89.85
22	P12	-1178.8	-821.5	44	RESB	1177.4	22.75

Absolute Maximum Rating

Parameter	Rating	Unit
Ambient temperature under bias	-55 ~ +125	°C
Storage temperature	-65 ~ + 150	°C
Voltage on any Port I/O Pin or RST with respect to Ground	-0.5 ~ VCC + 0.5	V
Voltage on VCC with respect to Ground	-0.5 ~ +6.0	V
Maximum total current through VCC and Ground	100	mA
Maximum output current sunk by any Port pin	25	mA

*Note: stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

(VDD-VSS = 5.0 V, FOSC = 6MHz, Ta = 25 C; unless otherwise specified)

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Op. Voltage	VDD	-	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	A
Input High Voltage	V_{IH}	-	2	-	VDD	V
Input Low Voltage	V_{IL}	-	0	-	0.8	V
Port 0, 1, 2, 3 drive current	I_{OH}	VOH = 4.5V, VDD = 5.0V	-	2.5	-	mA
Port 0.4~0.7, 1, 2, 3 sink current	I_{OL1}	VOL = 0.4V, VDD = 5.0V	-	4.0	-	mA
Port 0.0~0.3 sink current	I_{OL2}	VOL = 3.2V, VDD = 5.0V	6	8	-	mA
Internal Pull-high Resistor	R_{PH}	VIL = 0V	-	27K	-	
Low Voltage Reset	V_{LVR}			3.6		V

AC Characteristics

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
CPU Op. Frequency	FCPU	VDD = 5.0V	0.5	3	-	MHz
POR duration	TPOR	FOSC = 6 MHz	10	-	-	mS

Instruction Set Summary

Addressing Mode Table

	Address Mode	Instruction Times in Memory Cycle	Memory Utilization in Number of Program Sequence Bytes
1	Absolute a	4(3)	3
2	Absolute Indexed Indirect (a,x)	5	3
3	Absolute Indexed with X a,x	4(1,3)	3
4	Absolute Indexed with Y a,y	4(1)	3
5	Absolute Indirect (a)	4(3)	3
6	Accumulator A	2	1
7	Immediate #	2	2
8	Implied i	2	1
9	Program Counter Relative r	2(2)	2
10	Stack s	3-7	1-4
11	Zero Page zp	3(3)	2
12	Zero Page Indexed Indirect (zp,x)	6	2
13	Zero Page Indexed with X zp,x	4(3)	2
14	Zero Page Indexed with Y zp,y	4	2
15	Zero Page Indirect (zp)	5	2
16	Zero Page Indirect Indexed with Y (zp),y	5	2

Notes: (indicated in parenthesis)
1. Page boundary, add 1 cycle if page boundary is crossed when forming address
2. Branch taken, add 1 cycle if branch is taken
3. Read-Modify-Write, add 2 cycles

Instruction Set Table

ADC	Add memory to accumulator with Carry	LDY	Load the Y register with memory
AND	"AND" memory with accumulator	LSR	Logical Shift one bit Right memory or accumulator
ASL	Arithmetic Shift one bit Left, memory or accumulator	NOP	No Operation
BBR	Branch on Bit Reset	ORA	"OR" memory with Accumulator
BBS	Branch of Bit Set	PHA	Push Accumulator on stack
BCC	Branch on Carry Clear (Pc=0)	PHP	Push Processor status on stack
BCS	Branch on Carry Set (Pc=1)	PHX	Push X register on stack
BEQ	Branch if Equal (Pz=1)	PHY	Push Y register on stack
BIT	Bit Test	PLA	Pull Accumulator from stack
BMI	Branch if result Minus (Pn=1)	PLP	Pull Processor status from stack
BNE	Branch if Not Equal (Pz=0)	PLX	Pull X register from stack
BPL	Branch if result Plus (Pn=0)	PLY	Pull Y register from stack
BRA	Branch Always	RMB	Reset Memory Bit
BVC	Branch on overflow Clear (Pv=0)	ROL	Rotate one bit Left memory or accumulator
BVS	Branch on overflow Set (Pv=1)	ROR	Rotate one bit Right memory or accumulator
CLC	Clear Cary flag	RTI	Return from Interrupt
CLD	Clear Decimal mode	RTS	Return from Subroutine
CLI	Clear Interrupt disable bit	SBC	Subtract memory from accumulator with borrow (Carry bit)
CLV	Clear overflow flag	SED	Set Decimal mode
CMP	Compare memory and accumulator	SEI	Set Interrupt disable status
CPX	Compare memory and X register	SMB	Set Memory Bit
CPY	Compare memory and Y register	STA	Store Accumulator in memory
DEC	Decrement memory or accumulate by one	STX	Store the X register in memory
DEX	Decrement X by one	STY	Store the Y register in memory
DEY	Decrement Y by one	STZ	Store Zero in memory
EOR	"Exclusive OR" memory with accumulate	TAX	Transfer the Accumulator to the X register
INC	Increment memory or accumulate by one	TAY	Transfer the Accumulator to the Y register
INX	Increment X register by one	TRB	Test and Reset memory Bit
INY	Increment Y register by one	TSB	Test and Set memory Bit
JMP	Jump to new location	TSX	Transfer the Stack pointer to the X register
JSR	Jump to new location Saving Return (Jump to Subroutine)	TXA	Transfer the X register to the Accumulator
LDA	Load Accumulator with memory	TXS	Transfer the X register to the Stack pointer register
LDX	Load the X register with memory	TYA	Transfer Y register to the Accumulator

Instruction Set Summary

MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSD
0		ORA (zp,x) 6,2			TSB zp 5,2	ORA zp 3,2	ASL zp 5,2	RMB0 zp 5,2	PHP s 3,1	ORA # 2,2	ASL A 2,1		TSB a 6,3	ORA a 4,3	ASL a 6,3	BBR0 r 5,3	0
1	BPL r 2,2	ORA (zp),y 5,2	ORA (zp) 5,2		TRB zp 5,2	ORA zp,x 4,2	ASL zp,x 6,2	RMB1 zp 5,2	CLC i 2,1	ORA a,y 4,3	INC A 2,1		TRB a 6,3	ORA a,x 4,3	ASL a,x 6,3	BBR1 r 5,3	1
2	JSR a 6,3	AND (zp,x) 6,2			BIT zp 3,2	AND zp 3,2	ROL zp 5,2	RMB2 zp 5,2	PLP s 4,1	AND # 2,2	ROL A 2,1		BIT a 4,3	AND a 4,3	ROL a 6,3	BBR2 r 5,3	2
3	BMI r 2,2	AND (zp),y 5,2	AND (zp) 5,2		BIT zp,x 4,2	AND zp,x 4,2	ROL zp,x 6,2	RMB3 zp 5,2	SEC i 2,1	AND a,y 4,3	DEC A 2,1		BIT a,x 4,3	AND a,x 4,3	ROL a,x 6,3	BBR3 r 5,3	3
4	RTI s 6,1	EOR (zp,x) 6,2				EOR zp 3,2	LSR zp 5,2	RMB4 zp 5,2	PHA s 3,1	EOR # 2,2	LSR A 2,1		JMP a 3,3	EOR a 4,3	LSR a 6,3	BBR4 r 5,3	4
5	BVC r 2,2	EOR (zp),y 5,2	EOR (zp) 5,2			EOR zp,x 4,2	LSR zp,x 6,2	RMB5 zp 5,2	CLI i 2,1	EOR a,y 4,3	PHY s 3,1			EOR a,x 4,3	LSR a,x 6,3	BBR5 r 5,3	5
6	RTS s 6,1	ADC (zp,x) 6,2			STZ zp 3,2	ADC zp 3,2	ROR zp 5,2	RMB6 zp 5,2	PLA s 4,1	ADC # 2,2	ROR A 2,1		JMP (a) 6,3	ADC a 4,3	ROR a 6,3	BBR6 r 5,3	6
7	BVS r 2,2	ADC (zp),y 5,2	ADC (zp) 5,2		STZ zp,x 4,2	ADC zp,x 4,2	ROR zp,x 6,2	RMB7 zp 5,2	SEI i 2,1	ADC a,y 4,3	PLY s 4,1		JMP (a,x) 6,3	ADC a,x 4,3	ROR a,x 6,3	BBR7 r 5,3	7
8	BRA r 3,2	STA (zp,x) 6,2			STY zp 3,2	STA zp 3,2	STX zp 3,2	SMB0zp 5,2	DEY i 2,1	BIT # 2,2	TXA i 2,1		STY a 4,3	STA a 4,3	STX a 4,3	BBS0 r 5,3	8
9	BCC r 2,2	STA (zp),y 6,2	STA (zp) 5,2		STY zp,x 4,2	STA zp,x 4,2	STX zp,y 4,2	SMB1 zp 5,2	TYA i 2,1	STA a,y 4,3	TXS i 2,1		STZ a 4,3	STA a,x 4,3	STZ a,x 5,3	BBS1 r 5,3	9
A	LDY # 2,2	LDA (zp,x) 6,2	LDX # 2,2		LDY zp 3,2	LDA zp 3,2	LDX zp 3,2	SMB2 zp 5,2	TAY i 2,1	LDA # 2,2	TAX i 2,1		LDY a 4,3	LDA a 4,3	LDX a 4,3	BBS2 r 5,3	A
B	BCS r 2,2	LDA (zp),y 5,2	LDA (zp) 5,2		LDY zp,x 4,2	LDA zp,x 4,2	LDX zp,y 4,2	SMB3 zp 5,2	CLV i 2,1	LDA a,y 4,3	TSX i 2,1		LDY a,x 4,3	LDA a,x 4,3	LDX a,y 4,3	BBS3 r 5,3	B
C	CPY # 2,2	CMP (zp,x) 6,2			CPY zp 3,2	CMP zp 3,2	DEC zp 5,2	SMB4 zp 5,2	INY i 2,1	CMP # 2,2	DEX i 2,1		CPY a 4,3	CMP a 4,3	DEC a 6,3	BBS4 r 5,3	C
D	BNE r 2,2	CMP (zp),y 5,2	CMP (zp) 5,2			CMP zp,x 4,2	DEC zp,x 6,2	SMB5 zp 5,2	CLD i 2,1	CMP a,y 4,3	PHX s 3,1			CMP a,x 4,3	DEC a,x 6,3	BBS5 r 5,3	D
E	CPX # 2,2	SBC (zp,x) 6,2			CPX zp 3,2	SBC zp 3,2	INC zp 5,2	SMB6 zp 5,2	INX i 2,1	SBC # 2,2	NOP i 2,1		CPX a 4,3	SBC a 4,3	INC a 6,3	BBS6 r 5,3	E
F	BEQ r 2,2	SBC (zp),y 5,2	SBC (zp) 5,2			SBC zp,x 4,2	INC zp,x 6,2	SMB7 zp 5,2	SED i 2,1	SBC a,y 4,3	PLX s 4,1			SBC a,x 4,3	INC a,x 6,3	BBS7 r 5,3	F
MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSD

A: Accumulator
X: Index Register X
Y: Index Register Y
zp: Address8 or zero page
a: Adress16
s: Stack
r: Relative
i : Implied

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
addr16:	Absolute Address
addr8:	Zero Page/Relative Address
addr+(index):	Combined Address
addr $\rightarrow$16:	Address Extend to Absolute Address (Get two addr8 contents continuously)
label:	Address Variable
~:	1's compliment
$\cap$:	AND
$\cup$:	OR
$\oplus$:	Exclusive OR
$\leftarrow$:	Transfer direction, result

Arithmetic Operations

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
ADC	#data	(ACC) ← (ACC) + #data + (C)	C, Z, V, S	2	2(4)
	addr8	(ACC) ← (ACC) + (addr8) + (C)	C, Z, V, S	2	3
	(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(1)(4)
	addr16	(ACC) ← (ACC) + (addr16) + (C)	C, Z, V, S	3	4(4)
	addr16, X	(ACC) ← (ACC) + [addr16 + (X)] + (C)	C, Z, V, S	3	4(1)(4)
	addr16, Y	(ACC) ← (ACC) + [addr16 + (Y)] + (C)	C, Z, V, S	3	4(1)(4)
SBC	#data	(ACC) ← (ACC) – #data – (~C)	C, Z, V, S	2	2(4)
	addr8	(ACC) ← (ACC) – (addr8) – (~C)	C, Z, V, S	2	3
	(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(1)(4)
	addr16	(ACC) ← (ACC) – (addr16) – (~C)	C, Z, V, S	3	4(4)
	addr16, X	(ACC) ← (ACC) – [addr16 + (X)] – (~C)	C, Z, V, S	3	4(1)(4)
	addr16, Y	(ACC) ← (ACC) – [addr16 + (Y)] – (~C)	C, Z, V, S	3	4(1)(4)
INC	A	(ACC) ← (ACC) + 1	C, Z	1	2
	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(1)
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
	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(1)
DEX		(X) ← (X) – 1	Z, S	1	2
DEY		(Y) ← (Y) – 1	Z, S	1	2
CMP	#data	(ACC) – #data	C, Z, S	2	2(4)
	addr8	(ACC) – (addr8)	C, Z, S	2	3
	(addr8)	(ACC) – [(addr8)]	C, Z, S	2	5
	addr8, X	(ACC) – [addr8 + (X)]	C, Z, S	2	4
	(addr8, X)	(ACC) – {[addr8 + (X) →16]}	C, Z, S	2	6
	(addr8), Y	(ACC) – [(addr8→16) + (Y)]	C, Z, S	2	5(1)(4)
	addr16	(ACC) – (addr16)	C, Z, S	3	4(4)
	addr16, X	(ACC) – [addr16 + (X)]	C, Z, S	3	4(1)(4)
	addr16, Y	(ACC) – [addr16 + (Y)]	C, Z, S	3	4(1)(4)
CPX	#data	(X) – #data	C, Z, S	2	2(4)
	addr8	(X) – (addr8)	C, Z, S	2	3
	addr16	(X) – (addr16)	C, Z, S	3	4(4)
CPY	#data	(Y) – #data	C, Z, S	2	2(4)
	addr8	(Y) – (addr8)	C, Z, S	2	3
	addr16	(Y) – (addr16)	C, Z, S	3	4(4)

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

Logic Operations

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
AND	#data	(ACC) ← (ACC) ∩ #data	Z, S	2	2(4)
	addr8	(ACC) ← (ACC) ∩ (addr8)	Z, S	2	3
	(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(1)(4)
	addr16	(ACC) ← (ACC) ∩ (addr16)	Z, S	3	4(4)
	addr16, X	(ACC) ← (ACC) ∩ [addr16 + (X)]	Z, S	3	4(1)(4)
	addr16, Y	(ACC) ← (ACC) ∩ [addr16 + (Y)]	Z, S	3	4(1)(4)
ORA	#data	(ACC) ← (ACC) ∪ #data	Z, S	2	2(4)
	addr8	(ACC) ← (ACC) ∪ (addr8)	Z, S	2	3
	(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(1)(4)
	addr16	(ACC) ← (ACC) ∪ (addr16)	Z, S	3	4(4)
	addr16, X	(ACC) ← (ACC) ∪ [addr16 + (X)]	Z, S	3	4(1)(4)
	addr16, Y	(ACC) ← (ACC) ∪ [addr16 + (Y)]	Z, S	3	4(1)(4)
EOR	#data	(ACC) ← (ACC) ⊕ #data	Z, S	2	2(4)
	addr8	(ACC) ← (ACC) ⊕ (addr8)	Z, S	2	3
	(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(1)(4)
	addr16	(ACC) ← (ACC) ⊕ (addr16)	Z, S	3	4(4)
	addr16, X	(ACC) ← (ACC) ⊕ [addr16 + (X)]	Z, S	3	4(1)(4)
	addr16, Y	(ACC) ← (ACC) ⊕ [addr16 + (Y)]	Z, S	3	4(1)(4)
ROL	A	(C) ← (ACC.7), (ACC.(n+1)) ← (ACC. n), (ACC.0) ← (C)	C, Z, S	1	2
	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(1)
ROR	A	(ACC.7) ← (C), (ACC. n) ← (ACC.(n+1)), (C) ← (ACC.0)	C, Z, S	1	2
	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) ←	C, Z, S	3	6

		(addr16.(n+1)), (C) ← (addr16.0)			
	addr16, X	[addr16 + (X).7] ← (C), [addr16 + (X).n] ← [addr16 + (X).(n+1)], (C) ← [addr16 + (X).0]	C, Z, S	3	6(1)
ASL	A	(C) ← (ACC.7), (ACC.(n+1)) ← (ACC. n), (ACC.0) ← 0	C, Z, S	1	2
	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(1)
LSR	A	(ACC.7) ← 0, (ACC. n) ← (ACC.(n+1)), (C) ← (ACC.0)	C, Z, S	1	2
	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(1)
BIT	#data	(ACC) ∩ #data	Z	2	2(4)
	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(1)(4)
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.

Data Transfer

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
LDA	#data	(ACC) ← #data	Z, S	2	2(4)
	addr8	(ACC) ← (addr8)	Z, S	2	3
	(addr8)	(ACC) ← [(addr8)]	Z, S	2	5
	addr8, X	(ACC) ← [addr8 + (X)]	Z, S	2	4
	(addr8, X)	(ACC) ← {[addr8 + (X) →16]}	Z, S	2	6
	(addr8), Y	(ACC) ← [(addr8→16) + (Y)]	Z, S	2	6(1)(4)
	addr16	(ACC) ← (addr16)	Z, S	3	4(4)
	addr16, X	(ACC) ← [addr16 + (X)]	Z, S	3	4(1)(4)
	addr16, Y	(ACC) ← [addr16 + (Y)]	Z, S	3	4(1)(4)
LDX	#data	(X) ← #data	Z, S	2	2(4)
	addr8	(X) ← (addr8)	Z, S	2	3
	addr8, Y	(X) ← [addr8 + (Y)]	Z, S	2	4
	addr16	(X) ← (addr16)	Z, S	3	4(4)
	addr16, Y	(X) ← [addr16 + (Y)]	Z, S	3	4(1)(4)
LDY	#data	(Y) ← #data	Z, S	2	2(4)
	addr8	(Y) ← (addr8)	Z, S	2	3
	addr8, X	(Y) ← [addr8 + (X)]	Z, S	2	4
	addr16	(Y) ← (addr16)	Z, S	3	4(4)
	addr16, X	(Y) ← [addr16 + (X)]	Z, S	3	4(1)(4)
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(1)(4)
	addr16	(addr16) ← (ACC)	-	3	4(4)
	addr16, X	[addr16 + (X)] ← (ACC)	-	3	4(1)(4)
	addr16, Y	[addr16 + (Y)] ← (ACC)	-	3	4(1)(4)
STX	addr8	(addr8) ← (X)	-	2	3
	addr8, Y	[addr8 + (Y)] ← (X)	-	2	4
	addr16	(addr16) ← (X)	-	3	4(4)
STY	addr8	(addr8) ← (Y)	-	2	3
	addr8, X	[addr8 + (X)] ← (Y)	-	2	4
	addr16	(addr16) ← (Y)	-	3	4(4)
STZ	addr8	(addr8) ← 00H	-	2	3
	addr8, X	[addr8 + (X)] ← 00H	-	2	4
	addr16	(addr16) ← 00H	-	3	4(4)
	addr16, X	[addr16 + (X)] ← 00H	-	3	5(1)(4)
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
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

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

Boolean Variable Manipulation

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
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
RMB0	addr8	(addr8.0) ← 0	Z	2	5
…					
RMB7	addr8	(addr8.7) ← 0	Z	2	5

Note: 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.

Program and Machine Control

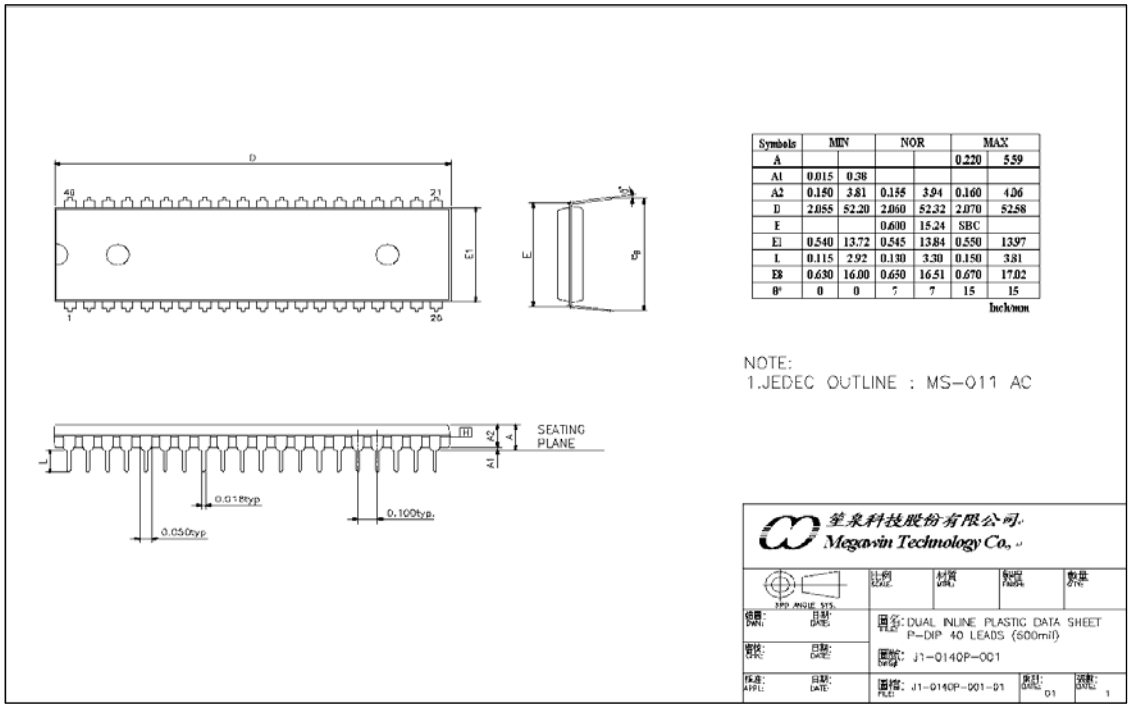
Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
JMP	addr16	(PC) ← label; the label may be address or variable.	-	3	3
	(addr16)	(PC) ← (label)	-	3	6
	(addr16, X)	(PC) ← {[label + (X) →16]}	-	3	6(1)
BRA	addr8	(PC) ← (PC)+addr8	-	2	3(2)
BEQ	addr8	(PC) ← (PC)+addr8 if Z == 1 (+/- relative)	-	2	2(2)(3)
BNE	addr8	(PC) ← (PC)+addr8 if Z == 0 (+/- relative)	-	2	2(2)(3)
BCS	addr8	(PC) ← (PC)+addr8 if C == 1 (+/- relative)	-	2	2(2)(3)
BCC	addr8	(PC) ← (PC)+addr8 if C == 0 (+/- relative)	-	2	2(2)(3)
BMI	addr8	(PC) ← (PC)+addr8 if (S == 1) (+/- relative)	-	2	2(2)(3)
BPL	addr8	(PC) ← (PC)+addr8 if (S == 0) (+/- relative)	-	2	2(2)(3)
BVS	addr8	(PC) ← (PC)+addr8 if (V == 1) (+/- relative)	-	2	2(2)(3)
BVC	addr8	(PC) ← (PC)+addr8 if (V == 0) (+/- relative)	-	2	2(2)(3)
BBR0	addr8	(PC) ← (PC)+addr8 if ACC.0 == 0 (+/- relative)	-	3	5(2)(3)
…					
BBR7	addr8	(PC) ← (PC)+addr8 if ACC.7 == 0 (+/- relative)	-	3	5(2)(3)
BBS0	addr8	(PC) ← (PC)+addr8 if ACC.0 == 1 (+/- relative)	-	3	5(2)(3)
…					
BBS7	addr8	(PC) ← (PC)+addr8 if ACC.7 == 1 (+/- relative)	-	3	5(2)(3)
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
NOP		No operation	-	1	2

Note: (1) Add 1 cycle for indexing across page boundaries, or write. This cycle contains invalid addresses.
Note: (2) Add 1 cycle if branch is taken.
Note: (3) Add 1 cycle if branch is taken across page boundaries.
Note: (4) Add 1 cycle for decimal mode.

Note: 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.

Package Dimensions

40-pin DIP



Symbols	MIN		NOR		MAX	
A					0.220	5.59
A1	0.015	0.38				
A2	0.150	3.81	0.155	3.94	0.160	4.06
D	2.055	52.20	2.060	52.32	2.070	52.58
E			0.600	15.24	SBC	
E1	0.540	13.72	0.545	13.84	0.550	13.97
L	0.115	2.92	0.130	3.30	0.150	3.81
eB	0.630	16.00	0.650	16.51	0.670	17.02
θ°	0	0	5	7	15	15

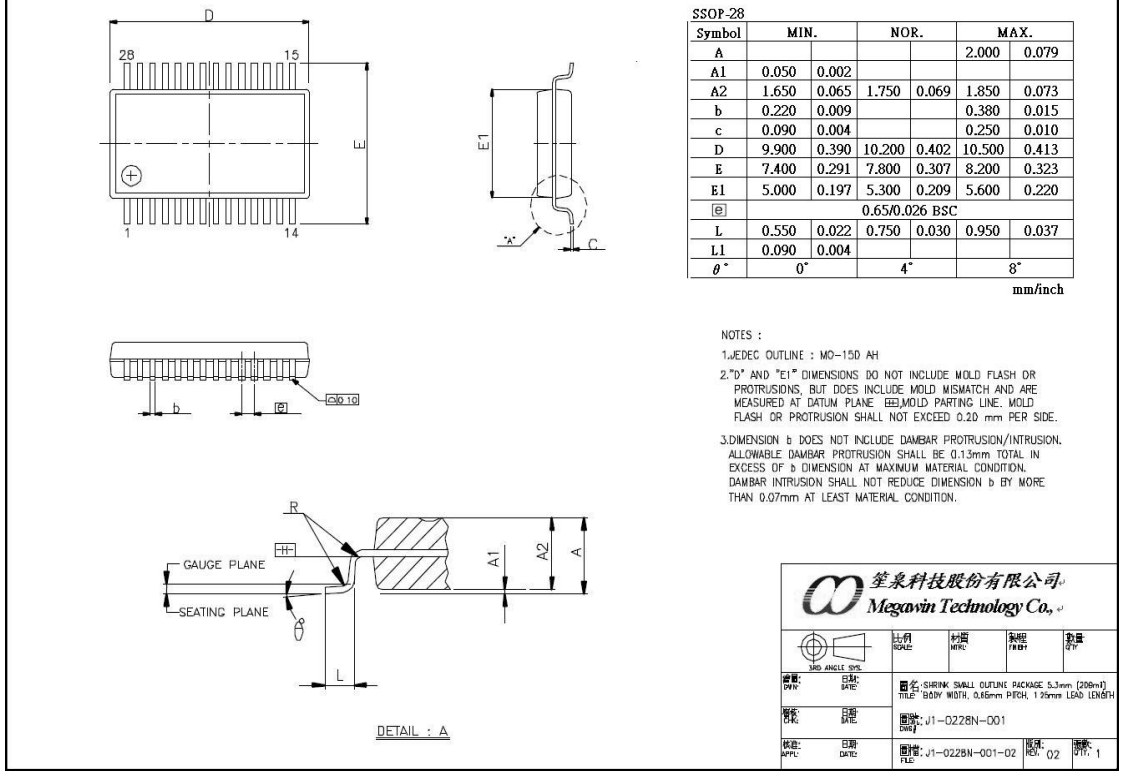
Inch/mm

NOTE:
1.JEDEC OUTLINE : MS-011 AC

3RD ANGLE SYS.	比例 SCALE:	材質 MTRL:	表面處理 FINISH	數量 QTY:
繪圖: DWN: 日期: DATE:	圖名: DUAL INLINE PLASTIC DATA SHEET P-DIP 40 LEADS (600mil) TITLE:			
審核: CHK: 日期: DATE:	圖號: J1-0140P-001 DWG#:			
核准: APPL: 日期: DATE:	檔名: J1-0140P-001-01 FILE:	版次: DATE: 01	張數: DATE: 1	

笙泉科技股份有限公司
Megawin Technology Co.,

28-SSOP



SSOP-28

Symbol	MIN.		NOR.		MAX.	
A					2.000	0.079
A1	0.050	0.002				
A2	1.650	0.065	1.750	0.069	1.850	0.073
b	0.220	0.009			0.380	0.015
c	0.090	0.004			0.250	0.010
D	9.900	0.390	10.200	0.402	10.500	0.413
E	7.400	0.291	7.800	0.307	8.200	0.323
E1	5.000	0.197	5.300	0.209	5.600	0.220
e	0.65/0.026 BSC					
L	0.550	0.022	0.750	0.030	0.950	0.037
L1	0.090	0.004				
θ°	0°		4°		8°	

mm/inch

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 ⊞ MOLD PARTING LINE. MOLD FLASH OR PROTRUSION SHALL NOT EXCEED 0.20 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.

3RD ANGLE SYS.	比例 SCALE:	材質 MTRL:	表面處理 FINISH	數量 QTY:
繪圖: DWN: 日期: DATE:	圖名: SHRINK SMALL OUTLINE PACKAGE 5.3mm (209mil) TITLE: BODY WIDTH, 0.65mm PITCH, 1.25mm LEAD LENGTH			
審核: CHK: 日期: DATE:	圖號: J1-0228N-001 DWG#:			
核准: APPL: 日期: DATE:	檔名: J1-0228N-001-02 FILE:	版次: REV. 02	張數: QTY. 1	

笙泉科技股份有限公司
Megawin Technology Co.,

Revision 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.
A3	2008/6	7	Added Memory Map diagram Modify 17k ohm to 27k ohm Revised Pad Define. Added 6502 instruction Modify Block Diagram Added USB Block Diagram Modify Power Saving Control SFR
A4	2008/12		Formatting