GENERAL DESCRIPTION

The MPC2F35 is a 65C02 MCU with an embedded 8k bytes flash 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.

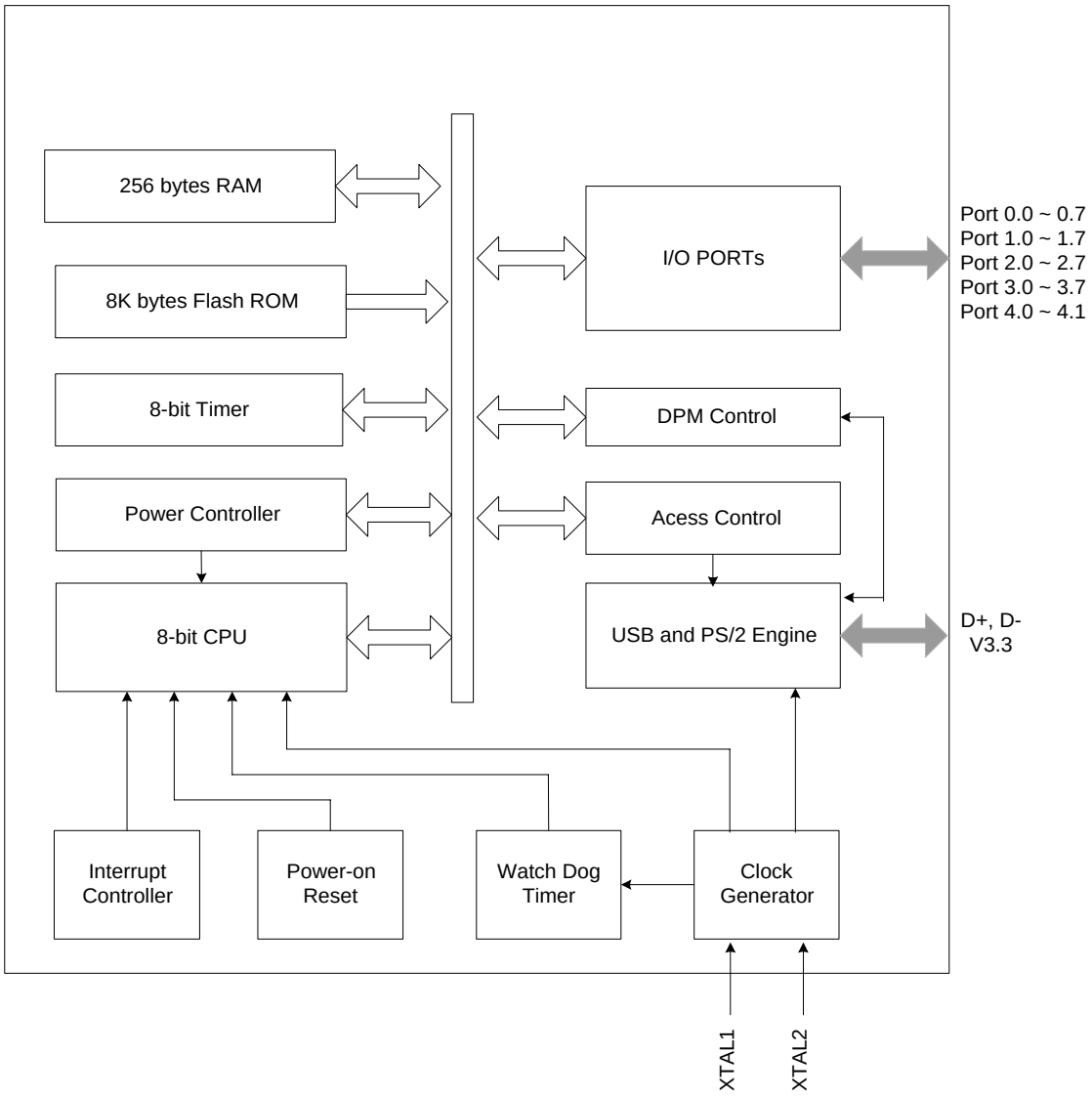
FEATURES

- 8-bit 65C02 micro-controller with 6 MHz external crystal or ceramic resonator
- Operation voltage: 4.35V to 5.5V
- Memory:
  - 8K Bytes Flash 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 3.3V regulated output for USB pull-up resistor
  - Provides remote wake-up
- Built-in low-voltage detector
- USB and PS/2 combo interfaces
- Support two power-saving modes: stop and halt mode
- Packages:
  - 28-SSOP: MPC2F35L
  - 40-PDIP: MPC2F35E2

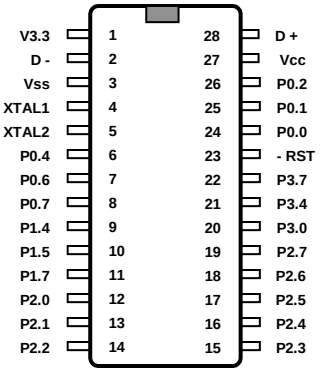
PAD DESCRIPTION

PIN Name	I/O	Description
P0.0~0.3	I/O	Bi-directional I/O, and 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
- RST	I	Reset pin, low active
XTAL1	I	6MHz crystal or resonator in
XTAL2	I	6MHz crystal or resonator out
D+/SCLK	I/O	USB data + with PS/2 compatible I/O
D-/SDATA	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 capacitor should be added on this pin

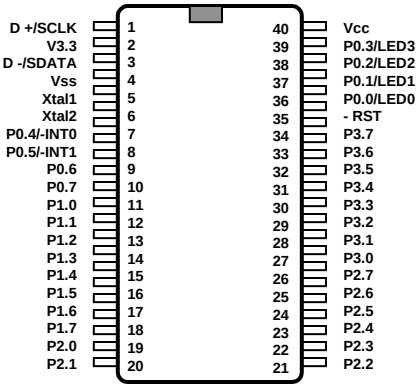
BLOCK DIAGRAM



PACKAGES



MPC2F35L



MPC2F35E2

FUNCTION DESCRIPTION

Registers

	A
	Y
	X
	P
PCH	PCL
1	S

Accumulator

The accumulator is a 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 MCU fetches the OP Code and the base address, and modifies the address by adding the index register to the base address before 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 flags are controlled by the program, and others may be controlled by both the program and the MCU.

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)

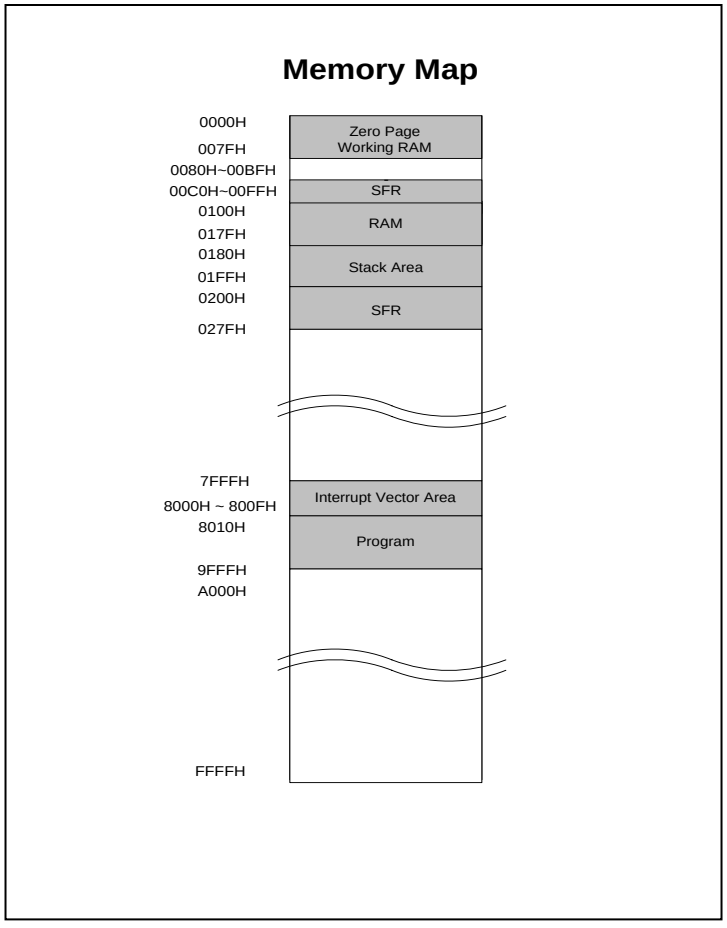
The 16-bit program counter register provides the addresses, which steps MPC2F35 through the sequential program instructions. Each time this MCU fetches an instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order 8 bits of the address bus, and the higher byte of the program counter (PCH) is placed on the high-order 8 bits. The counter is incremented each time when 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 the control of the MCU to perform the stack manipulations. 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 is a zero page working RAM (0000H ~ 007FH), a stack area (0180H ~ 01FFH) and two special function register areas (SFR, 00C0H ~ 00FFH and 0200H ~ 027FH) in MPC2F35. The locations 0100H to 017FH and the locations 0000H to 007FH share the same memory block, so MPC2F35 has a 256 bytes on-chip SRAM (zero page working RAM and stack area) and an 8k bytes on-chip flash ROM, which are addressed from 8000H to 9FFFH. The address mapping of MPC2F35 is shown as below.



Special Function Register (SFR)

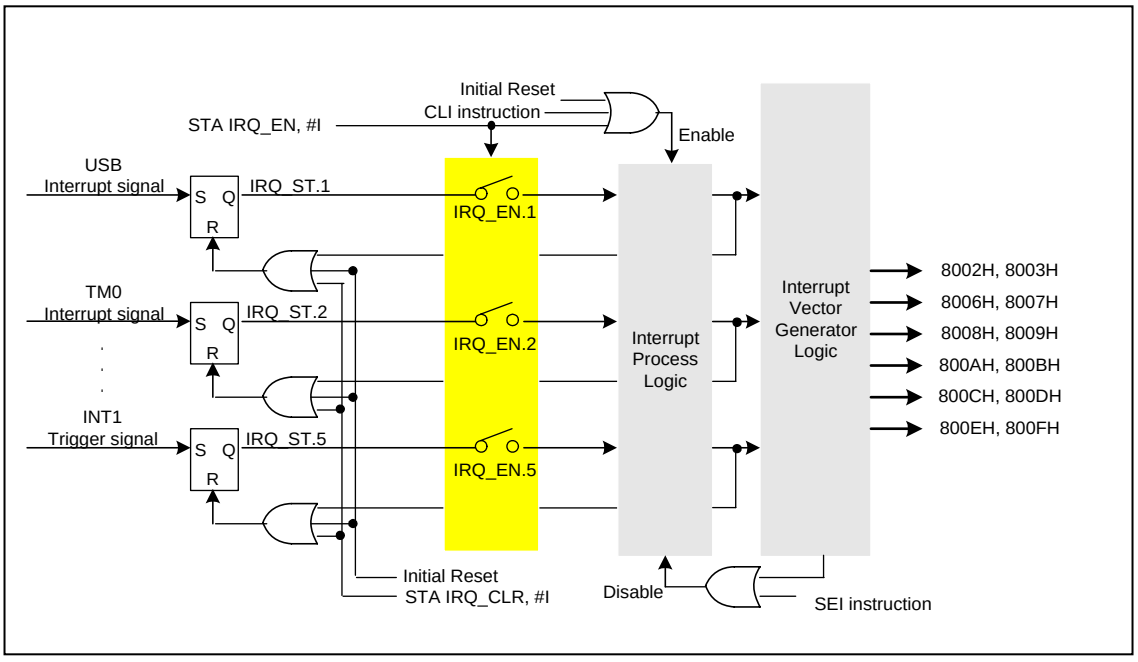
The address 00C0H to 00FFH and 0200H to 027FH are reserved for the special function registers (SFR). MPC2F35 has 36 SFRs to be used to control or store the status of I/O, timers, system clock and other peripherals.

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	Watchdog timer status flag	00
WDT_CLR	00DF	Watchdog clear	X
USB_CTL	00E0	USB bus 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 value for the PS/2 mode	X
PWR_CTL	0200	Power-saving control	00
FCPU_SR	0201	FCPU selector	00
RLH_EN	0202	Release the 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	Description
8002H, 8003H	RESET	1	External.	Initial reset
8006H, 8007H	USB	2	Internal	USB interrupt
8008H, 8009H	TM0	3	Internal	Timer 0 overflow interrupt
800AH, 800BH	P3	4	External.	Port 3 interrupt vector
800CH, 800DH	INT0	5	External.	INT0 external interrupt vector
800EH, 800FH	INT1	6	External.	INT1 external interrupt vector

There are six interrupt sources provided in MPC2F35. The flag IRQ_EN and IRQ_ST are used to control the interrupts. When flag IRQ_ST is set to '1' by the hardware and the corresponding bits of flag IRQ_EN has been set by firmware, an interrupt is generated. When an interrupt occurs, all 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	-	-	INT1	INT0	P3	TM0	USB	-	√	√

Program can enable (setting to "1") or disable (clearing to "0") the ability of triggering IRQ through this register.

- USB: USB finishes Rx or Tx data
- TM0: Timer0 underflow
- P3: Falling edge trigger signal occurs at port 3 input mode
- INT0, INT1: Falling edge trigger signal occurs at P0.4 and 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	-	-	INT1	INT0	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	-	-	INT1	INT0	P3	TM0	USB	-	-	√

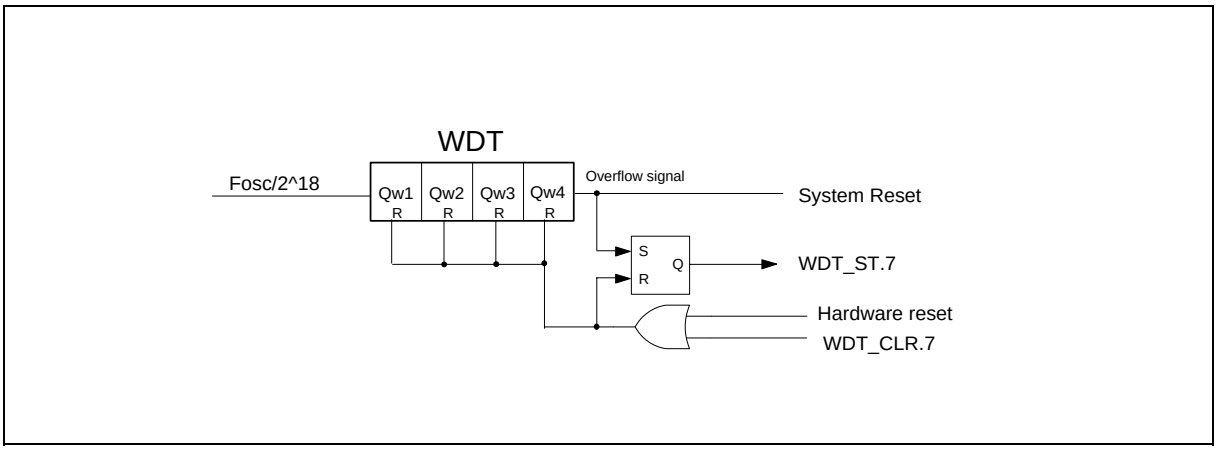
Program can clear the interrupt event by writing '1' into the corresponding bit.

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	-	-	-	-	-	-	-	-	√

- Bit 3 ~ Bit 0: Contents of WDT
- RSTS: WDT reset status, set by the hardware when WDT overflows, and clear by the firmware or the hardware reset
- CLR: RSTS clear and WDT reset control bit, the program can clear the RSTS bit and reset WDT by writing "1" into the CLR bit

The watchdog timer (WDT) is organized as a 4-bit counter, which is 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. It also can be cleared by hardware reset or writing 1 to bit 7 of WDT_CLR. The interval of WDT to cause reset is around 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. 1: Disable, 0: Enable (default)
- CKC: Oscillator control bit. 1: Disable OSC, 0: Enable OSC (default)
- HALT: FCPU off-line control bit. 1: FCPU off-line, 0: FCPU on-line (default)

When the low-voltage detector is enabled, and if it senses the power voltage is lower than 3.3V, then MPC2F35 will be reset automatically.

Programmer can switch the normal operation mode to the power-saving mode for reducing power consumption through this register. There are two power saving modes in MPC2F35.

Stop mode: (PWR_CTL.CKC = 1)

System clock stops the built-in oscillator if setting the CKC bit in the PWR_CTL SFR. MPC2F35 can be awakened from the stop mode by 4 ways: the port 3 interrupt, the hardware reset, the power-on reset and the USB wake-up.

Halt mode: (PWR_CTL.HALT = 1)

Setting the HALT bit to let the clock source of MPC2F35 to be in the off-line status, but the oscillator works or not will be depended on the content of the CKC bit in the PWR_CTL SFR.

MPC2F35 can be awakened from the halt mode by 3 ways: the interrupts (USB, Timer 0, Port3, INT0 and INT1) can be assigned by the RLH_EN register, the hardware reset, or the 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	-	√

At the 6M Hz external crystal, the internal clock source of MPC2F35 is 3M Hz by the default value.

- CKS: F_{CPU} clock source select register. 0: FOSC/2 (default), 1:FOSC

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	-	-	INT1	INT0	P3	TM0	USB	-	-	√

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

0: Disable (default), 1: Enable

After setting RLH_EN register, once there is one interrupt to release the halt mode, the programmer can check the corresponding bit of the IRQ_ST register to know which interrupt source to execute this release process.

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 (default), 1: Enable timer clock
- RL/S: Auto-reload disable/enable. 0: Enable auto-reload (default), 1: Disable auto-reload

TKI2	TKI1	TKI0	Timer 0 clock source (F_{TM0CK})
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

When timer 0 is used, it starts to pre-load value to this down-counter by setting the STC bit in the TM0_CTL SFR and its underflow frequency (F_{TM0_UV}) of timer 0 can be calculated by the following equation:

$F_{TM0\_UV} = F_{TM0CK} / (TM0+1)$, where F_{TM0CK} is selected by TKI2, TKI1 and TKI0

For example: if F_{OSC}=6M Hz and TKI2=0, TKI1=1, TKI0=0, then $F_{TM0CK} = F_{OSC} / 32$

TM0	F_{TM0_UV} Frequency
00H	Invalid
01H	93.75 KHz
02H	62.5 KHz
…	…
FFH	732.42 Hz

General Purpose 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: Values on the pin of Port 0 while reading from Port 0.
- P0_CR: Configure P0.0 ~ P0.7 to be input or output individually. 0: Input (default), 1: Output
- P0_MR: Configure the output mode of P0.0 ~ P0.7 with a 17k ohm pull-high resistor, CMOS or NMOS open drain
  - MP00 (P0_MR.0): P0.0 ~ P0.3 with the pull-high control bit, 0: Disable (default), 1:Enable
  - MP01 (P0_MR.1): P0.0 ~ P0.3 with the CMOS or NMOS selector, 0: CMOS (default), 1: NMOS
  - MP04 (P0_MR.4): P0.4 ~ P0.7 with the pull-high control bit, 0: Disable (default), 1: Enable
  - MP05 (P0_MR.5): P0.4 ~ P0.7 with the CMOS or NMOS selector, 0: CMOS (default), 1: NMOS

At initial reset, Port 0 is all in the input mode. Each pin of Port 0 can be specified as the input or output mode independently by the P0_CR SFR. When Port 0 is used as the output port, CMOS or NMOS open drain output type can be selected by the P0_MR register. Port 0 has 17k ohm internal pull-high resistors that can be enabled/disabled by specifying the MP00 and MP04 in the P0_MR register respectively. The pull-high resistor is automatically disabled only when the port is configured as CMOS output. Schmitt trigger circuit is added in the input path of P0.0~P0.3. User should be carefully 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 INT0 and INT1 interrupt sources. A falling edge at the two pins will set the corresponding bits in the IRQ_ST register to 1, and the external interrupt subroutines will be executed if the corresponding bits in the IRQ_EN register are also 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. Its structure is the same with Port 0, and refers 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: Values on Port 1 pins while reading from Port 1.
- P1_CR: Configure P1.0 ~ P1.7 to be input or output individually. 0: input, 1: output
- P1_MR: Configure the output mode of P1.0 ~ P1.7 with a 17k ohm pull-high resistor, CMOS or NMOS open drain

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, its structure is the same with Port 0 and refers 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: Values on Port 2 pins while reading from Port 2.
- P2_CR: Configure P2.0 ~ P2.7 to be input or output individually. 0: input, 1: output
- P2_MR: Configure the output mode of P2.0 ~ P2.7 with a 17k ohm pull-high resistor, CMOS or NMOS open drain

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, its structure is the same with Port 0 and refers 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: Values on Port 3 pins while reading from Port 3.

- P3_CR: Configure P3.0 ~ P3.7 to be input or output individually. 0: input, 1: output
- P3_MR: Configure the output mode of P3.0 ~ P3.7 with a 17k ohm pull-high resistor, CMOS or NMOS open drain

When port 3 is used as the input mode, it provides the pin interrupt function while a falling edge occurs at any pin of the port 3 and will set the P3 bit of the IRQ_ST SFR. The same event can release the stop mode to enable the oscillator, and this interrupt also can release the halt mode if the P3 bit in the RLH_EN SFR is set. Finally, an interrupt subroutine will be executed if setting the P3 bit in the IRQ_E SFR.

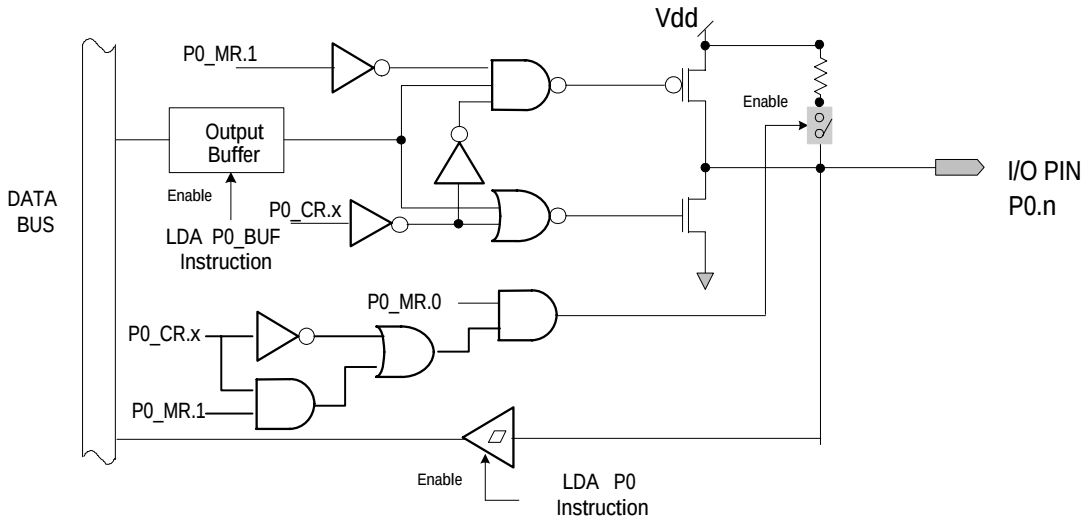
Port 4

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	√	√

MPC2F35 only provide two pins (P4.0 and P4.1) on the port 4, and these pins also are I/O pins. The structure is the same with Port 0, and please refers 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: Values on Port 4 pins while reading from Port 4.
- P4_CR: Configure P4.0 and P4.1 to be input or output individually. 0: input, 1: output
- P4_MR: Configure the output mode of P4.0 and P4.1 with a 17k ohm pull-high resistor, CMOS or NMOS open drain

Input/Output Pin --- P0~P4



USB INTERFACE

MPC2F35 provides the interface of PS/2 and USB combinative operation by programming the below registers, the user can be easily change the configuration between USB and PS/2 for meeting the environment of the host.

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: USB bus control register,
  - REGC: 3.3V regulator control. 0: Disable (default), 1: Enable
  - URD: USB register read control, writing 1 into this bit to read the USB register addressed by the USB_ADDR SFR
  - UWT: USB register write control, writing 1 into this bit to write the USB register addressed by the USB_ADDR SFR
- USB_ADDR: One USB register address to be accessed
- USB_DI: Data to be written into the USB register addressed by the USB_ADDR SFR
- USB_DO: Data to be read out from the USB register addressed by the USB_ADDR SFR

MPC2F35 is a Low-speed USB 1.1 version compliant with the USB transceiver and a built-in 3.3V regulator. The 3.3V regulator can be controlled by programming the REGC bit in the USB_CTL SFR. There are some USB registers in MPC2F35. The user can access these USB registers through the access the control registers, which is provided by MPC2F35. The sequence to access USB register should be as the following:

A. Write sequence:

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

B. Read sequence:

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

4. Write 0 into the URD bit

Whenever USB engine finished a transaction, it will generate an interrupt to acknowledge MPC2F35. The user can get information about the transaction through the above sequence. When USB engine received a reset instruction from the host, it will reset by itself and generate an interrupt. When USB engine received a wake-up instruction from the host while the device is being in the stop mode, it will generate a signal to enable the oscillator. If the host and the device are both in the stop mode, a falling edge on Port 3 can wake-up the device, and then remote wake up the host through USB engine.

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	√	-

- C1, C0: USB bus (D+ and D-) mode control selector.
    1. 0x: USB bus is at the USB operation (Default)
    2. 10: USB bus is at the PS/2 interface operation
- DPMO: PS/2 data output on USB data bus line (D+/D-), 0: output low, 1: pull-high
- DPMI: Value on the USB data bus line (Read only) while working at the PS/2 operation

MPC2F35's USB bus lines (D+ and D -) have two operating modes: USB low speed and PS/2 interface mode. User can program the C0 and C1 bit in the CDPM_CTL SFR to determine the operating mode of USB bus. The DPI and DMI bit in the DPMI SFR will record the content on the D+ and D- pin respectively.

The firmware can judge the USB bus line (D+ and D-) connection will be USB or PS/2 protocol by reading the value of the DPI and DMI bit in the DPMI SFR. For PS/2 interface application, the C1 and C0 in the DPM_CTL SFR have to set "10" first, thus the USB function will be unavailable. The user programs the value of USB bus (D+/D-) into the DPO and DMO bit in the DPMO SFR when MPC2F35 controls the D+/D- pin for the PS/2 operation. When DPO/DMO is programming as writing 0, it will make the D+/D- pin to output low. On the other hand, writing 1 will cause these pins to be pulled high. This I/O control operation would be easy to perform the PS/2 interface.

USB Special Function Registers (SFRs) Summary

There are 18 special function registers for the operation of Universal Serial Bus (USB). The detail definition is described as the following:

Mnemonic	USB Device SFRs	Address	Description							
DCON	Device Control Register	01H	TESTEN	–	–	–	–	–	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_XX00 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	-	Reserved: Write zero to this bit.
1	PUREN	Internal 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 cleared to ‘0’ after reset. FW should set ‘1’ to enable 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 Receive Interrupt Enable 3: Enables the receive done interrupt for function endpoint 3 (FRXD3).
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 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 receive byte count. Maximum is 8 bytes.
2	RXBC2	Receive Byte Count Bit 2: Store receive byte count. Maximum is 8 bytes.
1	RXBC1	Receive Byte Count Bit 1: Store receive byte count. Maximum is 8 bytes.
0	RXBC0	Receive Byte Count Bit 0: Store receive 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	Receive 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, it has no effect on RXSEQ. This bit always returns to ‘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 transmit byte count. Maximum is 8 bytes.
2	TXBC2	Transmit Byte Count Bit 2: Store transmit byte count. Maximum is 8 bytes.
1	TXBC1	Transmit Byte Count Bit 1: Store transmit byte count. Maximum is 8 bytes.
0	TXBC0	Transmit Byte Count Bit 0: Store transmit 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)

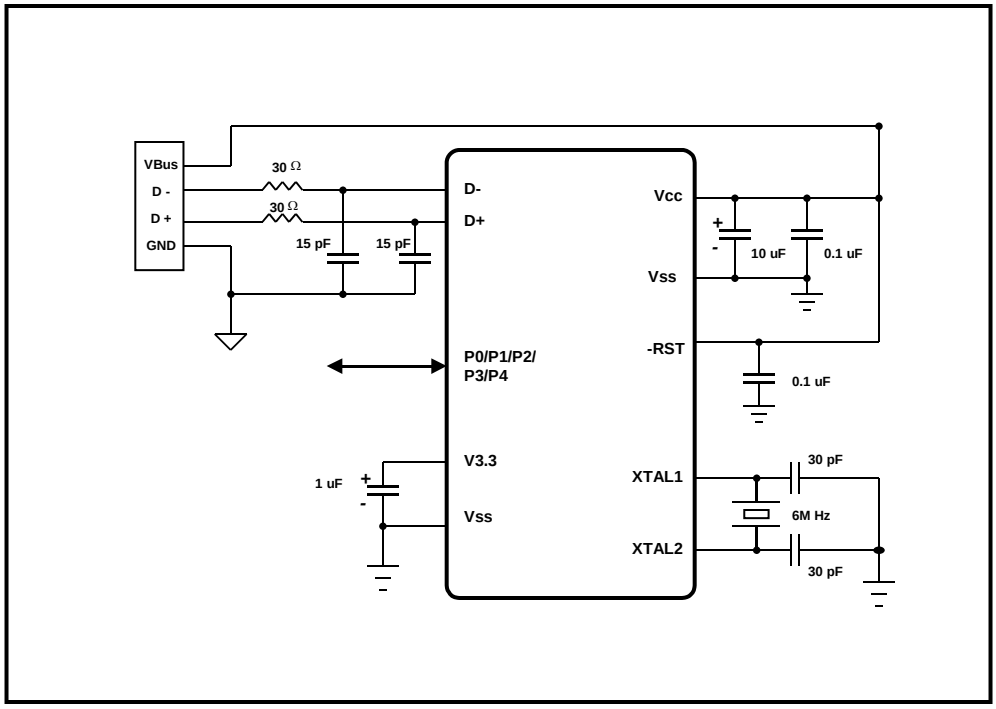
Read/Write Address: 32H

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 to "0" when read.
2	-	Reserved: Write zero to this bit.
1	-	Reserved: Write zero to this bit.
0	-	Reserved: Write zero to this bit.

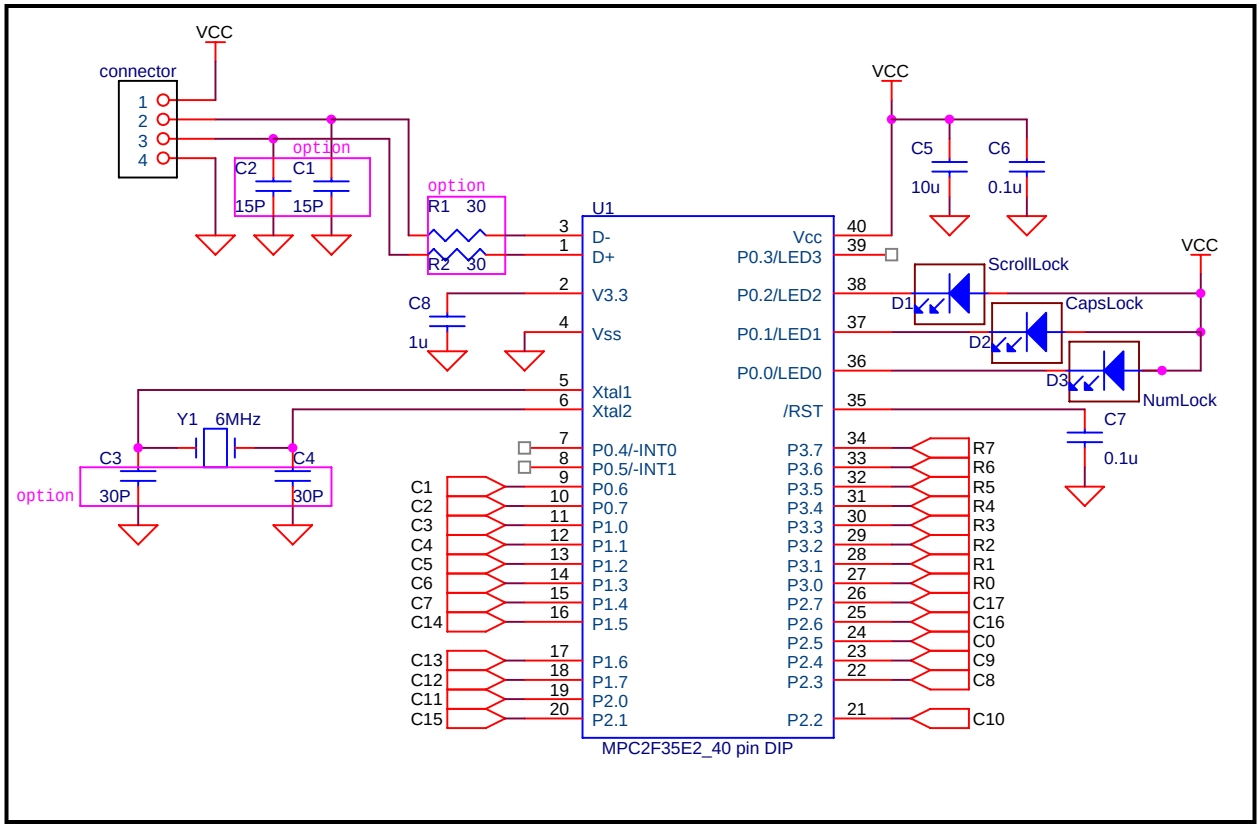
APPLICATION CIRCUIT

1. Normal:



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

2. USB keyboard:



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	-40 to +125	°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)

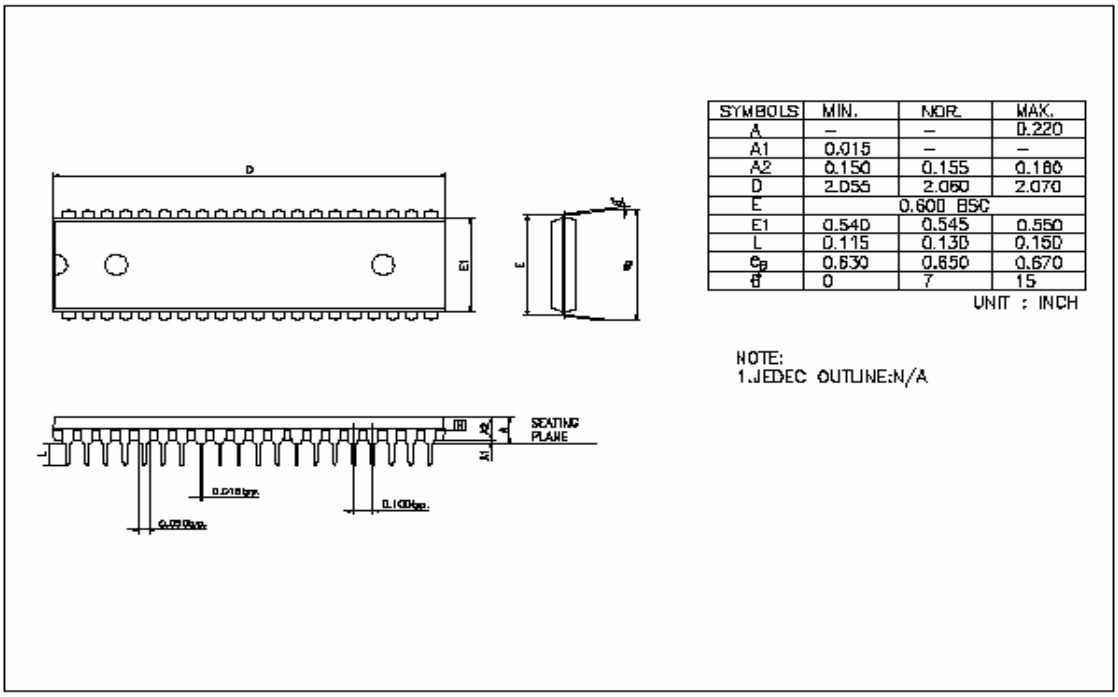
PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Op. Voltage	V_{DD}	3.0V≦V3.3≧3.6V	4.35	-	5.5	V
Op. Current	I_{OP}	No load (Ext.-V) In normal operation	-	12.5	20	mA
Suspend Current	I_{STB}	Internal 7.5KΩ with No load	-	-	500	μA
Input High Voltage	V_{IH}	-	2	-	V_{DD}	V
Input Low Voltage	V_{IL}	-	0	-	0.8	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

PACKAGE DIMENSIONS

40-pin DIP

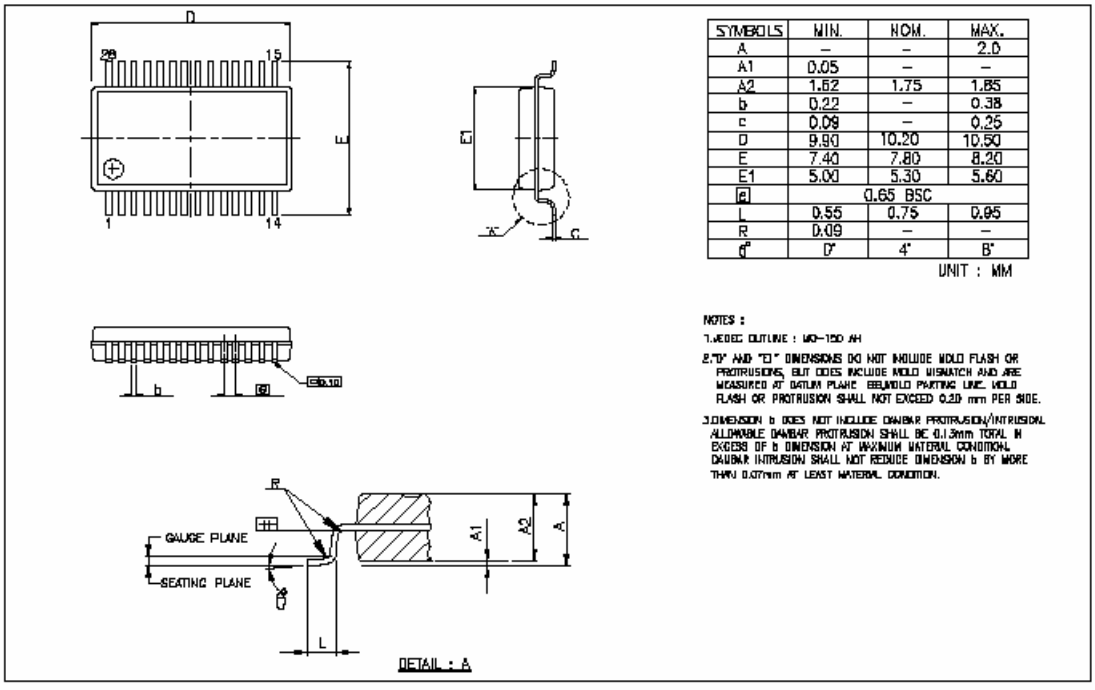


SYMBOLS	MIN.	NOR.	MAX.
A	–	–	0.220
A1	0.015	–	–
A2	0.150	0.155	0.160
D	2.055	2.060	2.070
E		0.600 BSC	
E1	0.540	0.545	0.550
L	0.115	0.130	0.150
eB	0.630	0.650	0.670
θ°	0	7	15

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 EEJMOLD 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.

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
addr+(index):	Combined Address
addr $\rightarrow_{16}$:	Address Extend to Absolute Address (Get two $addr_8$ contents continuously)
label:	Address Variable
~:	1's compliment
$\cap$:	AND
$\cup$:	OR
$\oplus$:	Exclusive OR
$\leftarrow$:	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	$addr_8$	(ACC) ← (ACC) ∪ ($addr_8$)	Z, S	2	3
	#data	(ACC) ← (ACC) ∪ #data	Z, S	2	2
	($addr_8$)	(ACC) ← (ACC) ∪ [($addr_8$)]	Z, S	2	5
	$addr_8$, X	(ACC) ← (ACC) ∪ [$addr_8$ + (X)]	Z, S	2	4
	($addr_8$, X)	(ACC) ← (ACC) ∪ {[$addr_8$ + (X) $\rightarrow_{16}$]}	Z, S	2	6
	($addr_8$), Y	(ACC) ← (ACC) ∪ [($addr_8 \rightarrow_{16}$) + (Y)]	Z, S	2	5*
	$addr_{16}$	(ACC) ← (ACC) ∪ ($addr_{16}$)	Z, S	3	4
	$addr_{16}$, X	(ACC) ← (ACC) ∪ [$addr_{16}$ + (X)]	Z, S	3	4*
	$addr_{16}$, Y	(ACC) ← (ACC) ∪ [$addr_{16}$ + (Y)]	Z, S	3	4*
EOR	$addr_8$	(ACC) ← (ACC) ⊕ ($addr_8$)	Z, S	2	3
	#data	(ACC) ← (ACC) ⊕ #data	Z, S	2	2
	($addr_8$)	(ACC) ← (ACC) ⊕ [($addr_8$)]	Z, S	2	5
	$addr_8$, X	(ACC) ← (ACC) ⊕ [$addr_8$ + (X)]	Z, S	2	4
	($addr_8$, X)	(ACC) ← (ACC) ⊕ {[$addr_8$ + (X) $\rightarrow_{16}$]}	Z, S	2	6
	($addr_8$), Y	(ACC) ← (ACC) ⊕ [($addr_8 \rightarrow_{16}$) + (Y)]	Z, S	2	5*
	$addr_{16}$	(ACC) ← (ACC) ⊕ ($addr_{16}$)	Z, S	3	4
	$addr_{16}$, X	(ACC) ← (ACC) ⊕ [$addr_{16}$ + (X)]	Z, S	3	4*
	$addr_{16}$, Y	(ACC) ← (ACC) ⊕ [$addr_{16}$ + (Y)]	Z, S	3	4*
CMP	$addr_8$	(ACC) – ($addr_8$) – (~C)	C, Z, S	2	3
	#data	(ACC) – #data – (~C)	C, Z, S	2	2
	($addr_8$)	(ACC) – [($addr_8$)] – (~C)	C, Z, S	2	5
	$addr_8$, X	(ACC) – [$addr_8$ + (X)] – (~C)	C, Z, S	2	3
	($addr_8$, X)	(ACC) – {[$addr_8$ + (X) $\rightarrow_{16}$]} – (~C)	C, Z, S	2	6
	($addr_8$), Y	(ACC) – [($addr_8 \rightarrow_{16}$) + (Y)] – (~C)	C, Z, S	2	5*
	$addr_{16}$	(ACC) – ($addr_{16}$) – (~C)	C, Z, S	3	4
	$addr_{16}$, X	(ACC) – [$addr_{16}$ + (X)] – (~C)	C, Z, S	3	4*
	$addr_{16}$, Y	(ACC) – [$addr_{16}$ + (Y)] – (~C)	C, Z, S	3	4*
CPX	#data	(X) – #data	C, Z, S	2	2
	$addr_8$	(X) – ($addr_8$)	C, Z, S	2	3
	$addr_{16}$	(X) – ($addr_{16}$)	C, Z, S	3	4
CPY	#data	(Y) – #data	C, Z, S	2	2
	$addr_8$	(Y) – ($addr_8$)	C, Z, S	2	3
	$addr_{16}$	(Y) – ($addr_{16}$)	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*	$addr_8$	($addr_{8.0}$) ← 0	Z	2	5
…					
RMB7	$addr_8$	($addr_{8.7}$) ← 0	Z	2	5
SEC		(C) ← 1	C	1	2
SEI		(I) ← 1	I	1	2
SED		(D) ← 1	D	1	2
SMB0*	$addr_8$	($addr_{8.0}$) ← 1	Z	2	5
…					
SMB7	$addr_8$	($addr_{8.7}$) ← 1	Z	2	5
INC	A	(ACC) ← (ACC) + 1	C, Z	1	2
INC	$addr_8$	($addr_8$) ← ($addr_8$) + 1	Z, S	2	5
	$addr_8$, X	[$addr_8$ + (X)] ← [$addr_8$ + (X)] + 1	Z, S	2	6
	$addr_{16}$	($addr_{16}$) ← ($addr_{16}$) + 1	Z, S	3	6
	$addr_{16}$, X	[$addr_{16}$ + (X)] ← [$addr_{16}$ + (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	$addr_8$	($addr_8$) ← ($addr_8$) – 1	Z, S	2	5
	$addr_8$, X	[$addr_8$ + (X)] ← [$addr_8$ + (X)] – 1	Z, S	2	6
	$addr_{16}$	($addr_{16}$) ← ($addr_{16}$) – 1	Z, S	3	6
	$addr_{16}$, X	[$addr_{16}$ + (X)] ← [$addr_{16}$ + (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	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*
	addr16	(ACC) ← (addr16)	Z, S	3	4
	addr16, X	(ACC) ← [addr16 + (X)]	Z, S	3	4*
	addr16, Y	(ACC) ← [addr16 + (Y)]	Z, S	3	4*
LDX	#data	(X) ← #data	Z, S	2	2
	addr8	(X) ← (addr8)	Z, S	2	3
	addr8, Y	(X) ← [addr8 + (Y)]	Z, S	2	4
	addr16	(X) ← (addr16)	Z, S	3	4
	addr16, Y	(X) ← [addr16 + (Y)]	Z, S	3	4*
LDY	#data	(Y) ← #data	Z, S	2	2
	addr8	(Y) ← (addr8)	Z, S	2	3
	addr8, X	(Y) ← [addr8 + (X)]	Z, S	2	4
	addr16	(Y) ← (addr16)	Z, S	3	4
	addr16, X	(Y) ← [addr16 + (X)]	Z, S	3	4*

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

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
STA	$addr_8$	$(addr_8) \leftarrow (ACC)$	-	2	3
	$(addr_8)$	$[(addr_8)] \leftarrow (ACC)$	-	2	5
	$addr_8$, X	$[addr_8 + (X)] \leftarrow (ACC)$	-	2	4
	($addr_8$, X)	$\{[\underline{addr_8 + (X)} \rightarrow_{16}]\} \leftarrow (ACC)$	-	2	6
	($addr_8$), Y	$[(\underline{addr_8} \rightarrow_{16}) + (Y)] \leftarrow (ACC)$	-	2	6*
	$addr_{16}$	$(addr_{16}) \leftarrow (ACC)$	-	3	4
	$addr_{16}$, X	$[addr_{16} + (X)] \leftarrow (ACC)$	-	3	4*
	$addr_{16}$, Y	$[addr_{16} + (Y)] \leftarrow (ACC)$	-	3	4*
STX	$addr_8$	$(addr_8) \leftarrow (X)$	-	2	3
	$addr_8$, Y	$[addr_8 + (Y)] \leftarrow (X)$	-	2	4
	$addr_{16}$	$(addr_{16}) \leftarrow (X)$	-	3	4
STY	$addr_8$	$(addr_8) \leftarrow (Y)$	-	2	3
	$addr_8$, X	$[addr_8 + (X)] \leftarrow (Y)$	-	2	4
	$addr_{16}$	$(addr_{16}) \leftarrow (Y)$	-	3	4
STZ	$addr_8$	$(addr_8) \leftarrow 00H$	-	2	3
	$addr_8$, X	$[addr_8 + (X)] \leftarrow 00H$	-	2	4
	$addr_{16}$	$(addr_{16}) \leftarrow 00H$	-	3	4
	$addr_{16}$, X	$[addr_{16} + (X)] \leftarrow 00H$	-	3	5
TAX		$(X) \leftarrow (ACC)$	Z, S	1	2
TXA		$(ACC) \leftarrow (X)$	Z, S	1	2
TAY		$(Y) \leftarrow (ACC)$	Z, S	1	2
TYA		$(ACC) \leftarrow (Y)$	Z, S	1	2
TSX		$(X) \leftarrow (SP)$	Z, S	1	2
TXS		$(SP) \leftarrow (X)$	-	1	2
TRB	$addr_8$	$(addr_8) \leftarrow (\sim ACC) \cap (addr_8)$	-	2	5
	$addr_{16}$	$(addr_{16}) \leftarrow (\sim ACC) \cap (addr_{16})$	-	3	6
TSB	$addr_8$	$(addr_8) \leftarrow (ACC) \cup (addr_8)$	-	2	5
	$addr_{16}$	$(addr_{16}) \leftarrow (ACC) \cup (addr_{16})$	-	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) $\rightarrow_{16}$]}	-	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
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.

Revision History

VERSION	DATE	PAGE	DESCRIPTION
A1	2004/04		Initial issue
A2	2005/01	39	Application circuit has been modified.
A3	2005/07	17~33	Revised USB Special Function Registers (SFRs) Summary and USB SFR Description.
A4	2008/12		Formatting