笙泉科技股份有限公司

Megawin Technology Co., Ltd.

MG64F237

Data Sheet

65C02 MCU with an USB 2.0 low-speed interface

Version 1.00

List of Contents

1 Description

The MG64F237 is a 65C02 MCU with an USB 2.0 low-speed interface. A PS/2 connection can be established on USB DP & DM pins by user firmware. It will be very suitable for low-cost keyboard and products like hand-held game, data bank, and I-toy, which need to download/upload data from the PC host.

2 Features

- 6502 8-bit CPU Core
- 8K Bytes MTP ROM
  - Flash write/erase cycle: 100
  - Flash data retention: 100 years at 25℃
- 256 bytes Data SRAM
  - Data RAM (0100H to 01BFH) and stacks RAM (01C0H to 01FFH).
  - Address 0100h~01BFH and 0000h~00BFH share the same memory block.
- 34+2 Programmable GPIO
  - Port 0 shared 1 pin with F_{OSC} output (6MHz / 3MHz / 1.5MHz)
  - Port 1 shared 2 pins with T0CKO/PWM0 and T1CKO/PWM1.
  - Port 2 shared with SPI interface (3 pins) and ICP interface (3 pins)
  - Port 3 shared with external crystal and external interrupt.
  - Port 0/1/2/3 support Wakeup Function.
  - Port 0/1/2/3/4 LED direct sink pins.
  - Support VDDO pin to setup output voltage.
- Master Mode SPI Interface
  - Clock Rate : 1.5MHz, 750KHz
  - MSB / LSB of the data byte is transmitted first.
- Timer / PWM
  - 8-bit auto-reload timer (Timer0) support T0CKO/PWM0 to P1.1
  - 8-bit auto-reload timer (Timer1) support T1CKO/PWM1 to P1.2
- Programmable Watch-dog Timer (WDT)
- Programmable system clock (6MHz / 3MHz)
- USB 2.0 low speed device controller
  - Built-in USB low-speed (1.5Mbps) transceiver
  - 8-bytes FIFO for endpoint 0 Control IN/OUT.
  - 8-bytes FIFO for endpoint 1 Interrupt IN.
  - 8-bytes FIFO for endpoint 2 Interrupt IN/OUT, default is IN.
  - Supports USB suspend/resume and remote wake-up event.
  - Software-controlled USB disconnection mechanism.
  - DP/DM combine with PS/2 Mode (PS2_CLK and PS2_Data)
- Built-in 5v to 3.3v regulator.
- Built-in 6Mhz±1.5% IHRCO with temperature -20℃ ~ 85℃.
- Low-Voltage detect: LVDF (Low Voltage detect Flag): 3.6V±5%
- Power saving modes
  - Halt(Idle) mode
  - Stop(Power-down) mode
- Operating condition:
  - Operating voltage: 2.7V ~ 5.5V with USB off-line application
  - Operating voltage: 4.0V ~ 5.5V with USB on-line application
  - Operating speed range: DC to 6MHz @VDD>2.7V
  - Operating ambient temperature: -20℃ ~ 85℃ for internal oscillator mode
  - Operating ambient temperature: -20℃ ~ 85℃ for external crystal mode
- Package Type
  - DICE : MG64F237H
  - LQFP-48 : MG64F237AD48
  - SSOP-16 : MG64F237AL16

3 Block Diagram

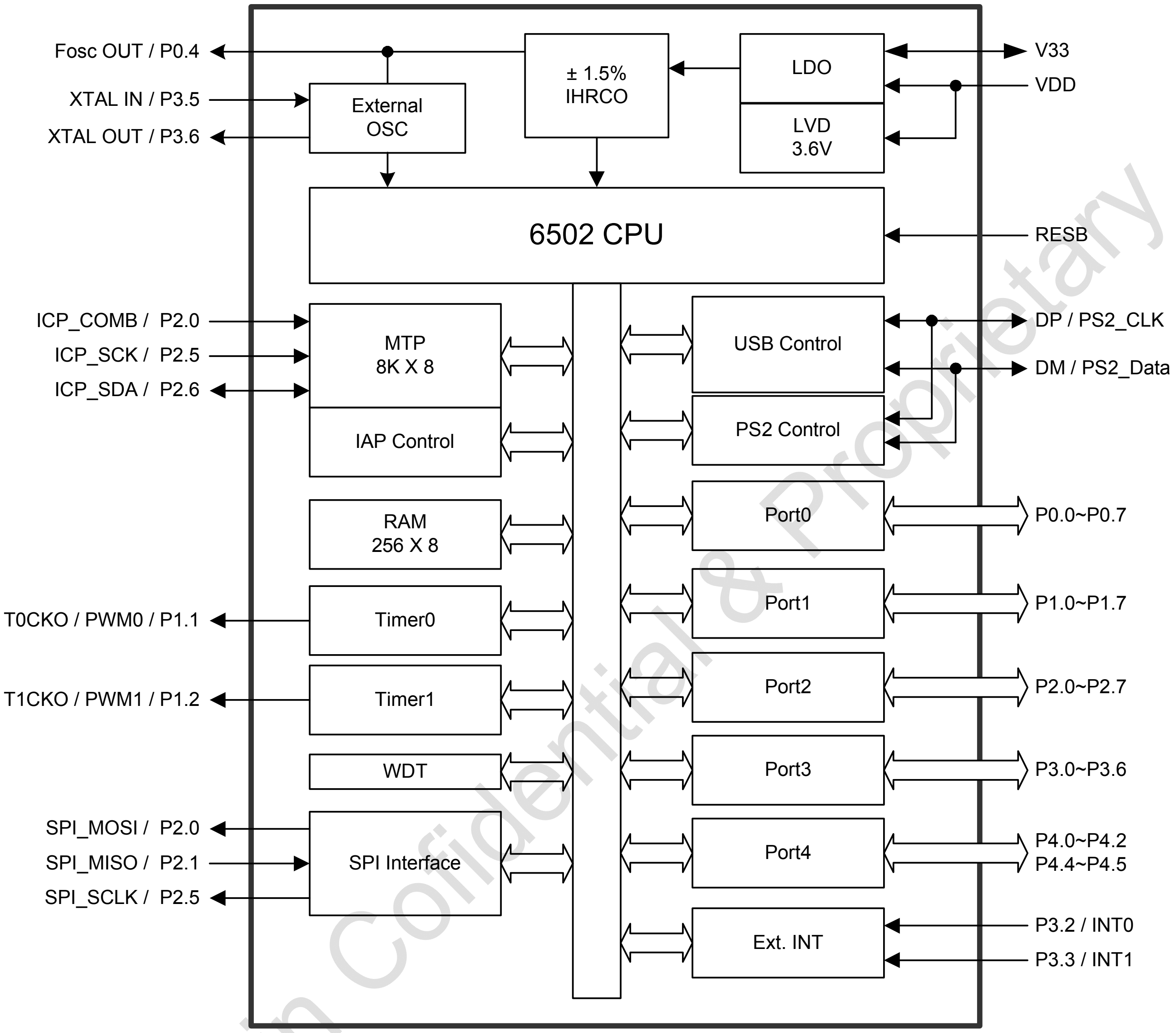


Figure 3-1 Block Diagram

4 Pin Description

4.1 Pin Definition

Table 4-1 Pin Definition Table

Pin Name	Pin Number			Type	Description
	DICE	SSOP16	LQFP48		
P45	1		1	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P44	44		46	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P42	36		38	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P41	35		37	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P40	34		36	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P36 / XTAL2	2		2	O / B	6MHz crystal output with high sink current GPIO P36. External 6MHz crystal output.
P35 / XTAL1	3		3	I / B	6MHz crystal input with high sink current GPIO P35. External 6MHz crystal input.
P34	4	7	4	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P33 / INT1	5	8	5	B	Bi-directional I/O, with wakeup function, and sink LED directly. External interrupt (INT1)
P32 / INT0	6	9	6	B	Bi-directional I/O, with wakeup function, and sink LED directly. External interrupt (INT0)
P31	7		7	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P30	8		8	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P27	9		9	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P26 / ICP_SDA	10	10	10	B	Bi-directional I/O, with wakeup function, and sink LED directly. ICP interface, ICP_SDA.
P25 / ICP_SCK / SPI_SCLK	11	11	11	B	Bi-directional I/O, with wakeup function, and sink LED directly. ICP interface, ICP_SCK. SPI_SCLK
P24	12		13	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P23	13		14	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P22	14		15	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P21 / SPI_MISO	15	12	16	B	Bi-directional I/O, with wakeup function, and sink LED directly. SPI_MISO
P20 / ICP_COMB / SPI_MOSI	16	13	17	B	Bi-directional I/O, with wakeup function, and sink LED directly. ICP interface, ICP_COMB. SPI_MOSI
P17	17		18	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P16	18		19	B	Bi-directional I/O, with wakeup function, and sink LED directly
P15	19		20	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P14	20		21	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P13	21		22	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P12 / T1CKO / PWM1	22	14	23	B	Bi-directional I/O, with wakeup function, and sink LED directly. Timer 1 underflow output. PWM 1 output.

Pin Name	Pin Number			Type	Description
	DICE	SSOP16	LQFP48		
P11 / T0CKO / PWM0	23	15	24	B	Bi-directional I/O, with wakeup function, and sink LED directly. Timer 0 underflow output. PWM 0 output.
P10	24		26	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P07	25		27	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P06	26		28	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P05	27		29	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P04 / ICKO	28	16	30	B	Bi-directional I/O, with wakeup function, and sink LED directly. F_{OSC} clock output.
P03	29		31	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P02	30		32	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P01	31		33	B	Bi-directional I/O, with wakeup function, and sink LED directly.
P00	32		34	B	Bi-directional I/O, with wakeup function, and sink LED directly.
RESB	33	1	35	I	Reset pin, low action, have internal pull high resistor.
VSSO	37	2	39	G	Ground for IO
VSS	38	2	40	G	Ground for others
V33	39	3	41	P	3.3V regulator output, a capacitor should be added on this pin.
DM / PS2_Data	40	4	42	B	USB DM(D-) combo with PS/2 Data pin.
DP / PS2_CLK	41	5	43	B	USB DP(D+) combo with PS/2 CLK pin.
VDD5V	42	6	44	P	5V Power for others
VDDO	43	6	45	P	Power for GPIO

Note: In the “Type” field,
“I” means input only.
“O” means output only.
“B” means bi-direction.
“P” means Power, “G” means Ground.

4.2 Pin Configuration

Please visit MEGAWIN website to download package dimension.
http://www.megawin.com.tw/Download_Grid.asp?BigClassName=Package%20Dimension

4.2.1 Package LQFP48

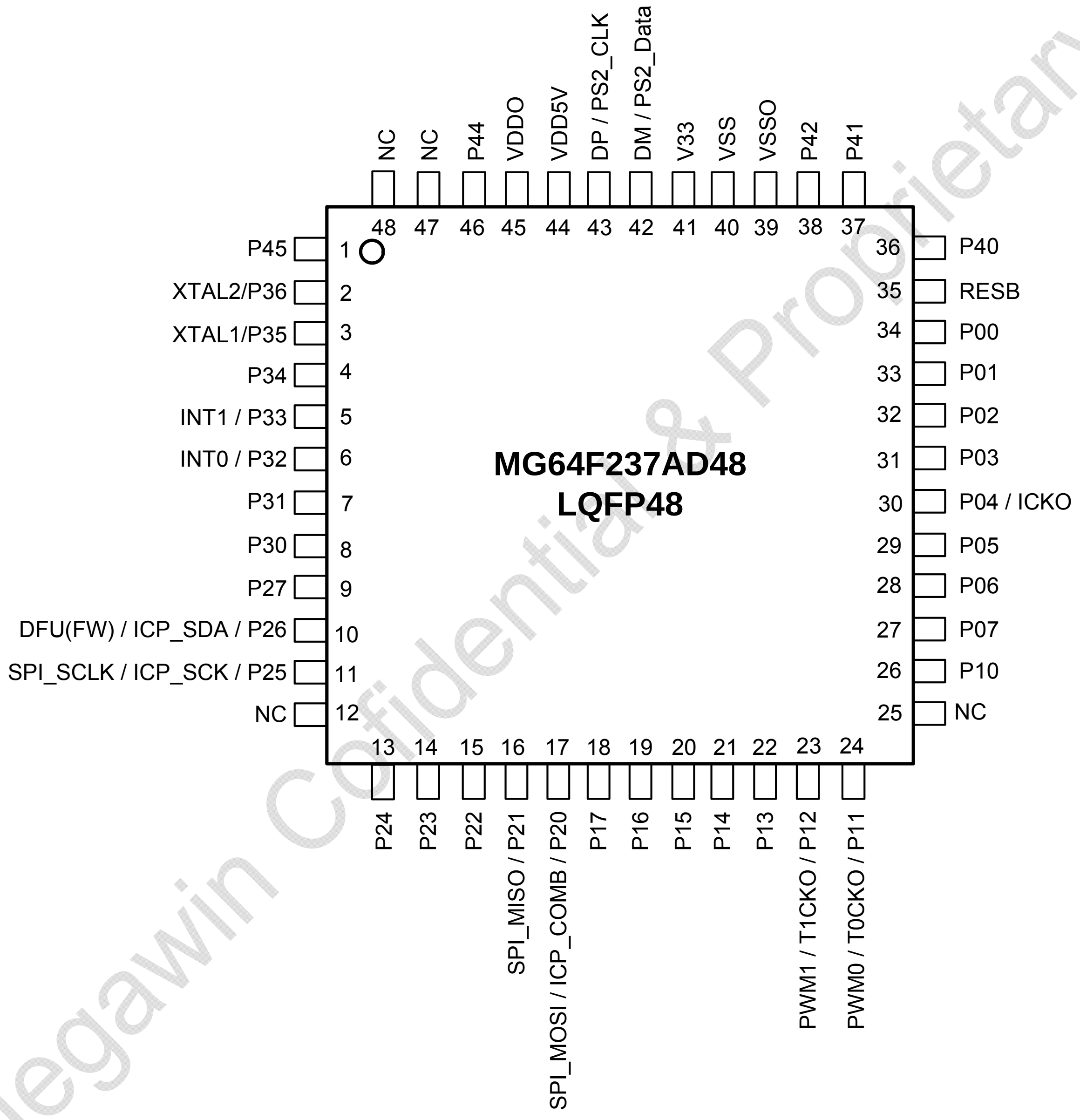


Figure 4-1 Package LQFP48

4.2.2 Package SSOP16

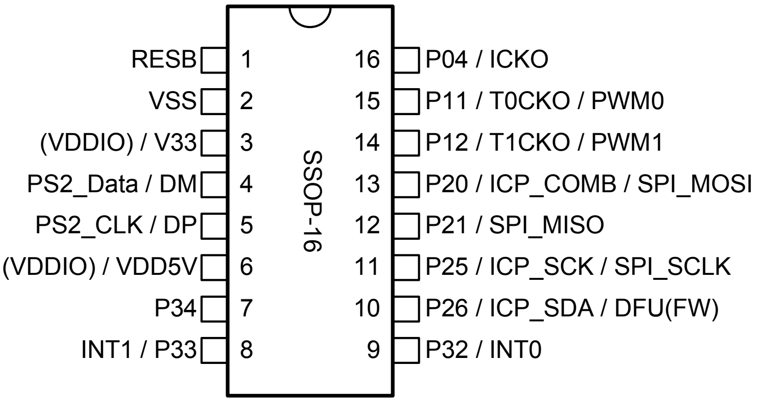


Figure 4-2 Package SSOP16

4.2.3 Dice

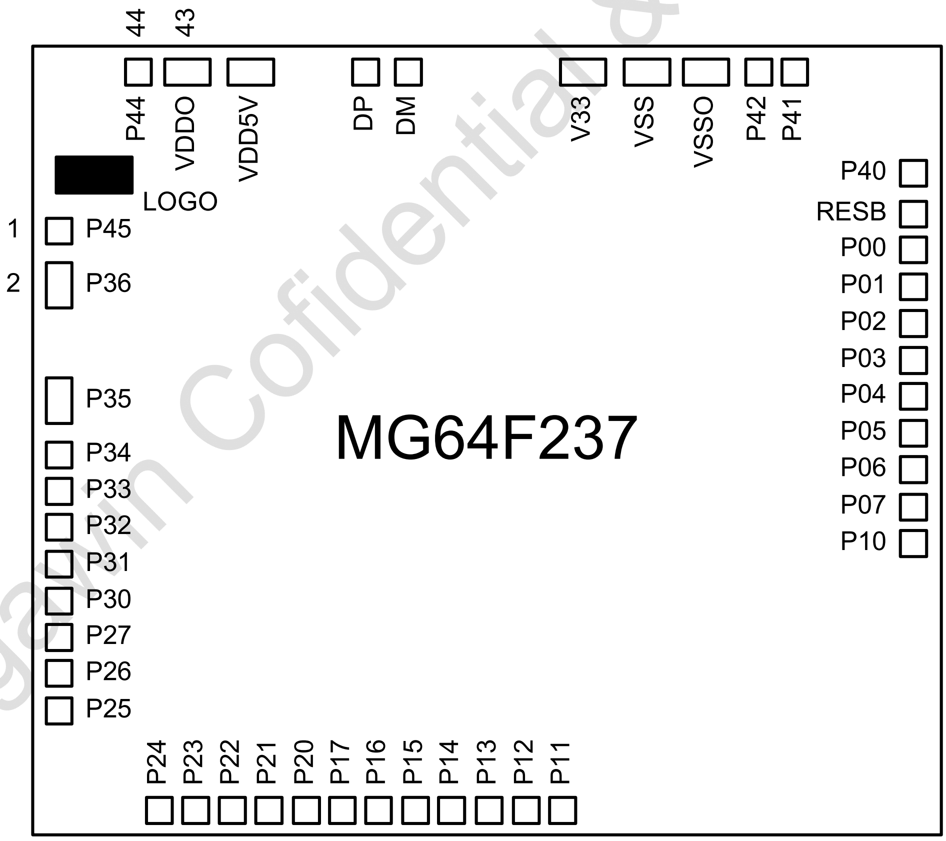


Figure 4-3 Dice

5 6502 Function Description

5.1 Registers

	A
	Y
	X
	P
PCH	PCL
1	S

5.1.1 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.

5.1.2 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.

5.1.3 Processor Status Register

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

5.1.4 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.

5.1.5 Stack Point(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.

6 Memory Mapping

There are 256 bytes SRAM in MG64F237. They are working RAM (0100H to 01BFH) and stacks (01C0H to 01FFH). Locations 0100h to 01BFH and the locations 0000h to 00BFH share the same memory block. The address 00C0H to 00FFH are special function registers area. The 8K bytes MTP ROM are addressed from E000H to FFFBH. The address mapping of MG64F237 is shown as below.

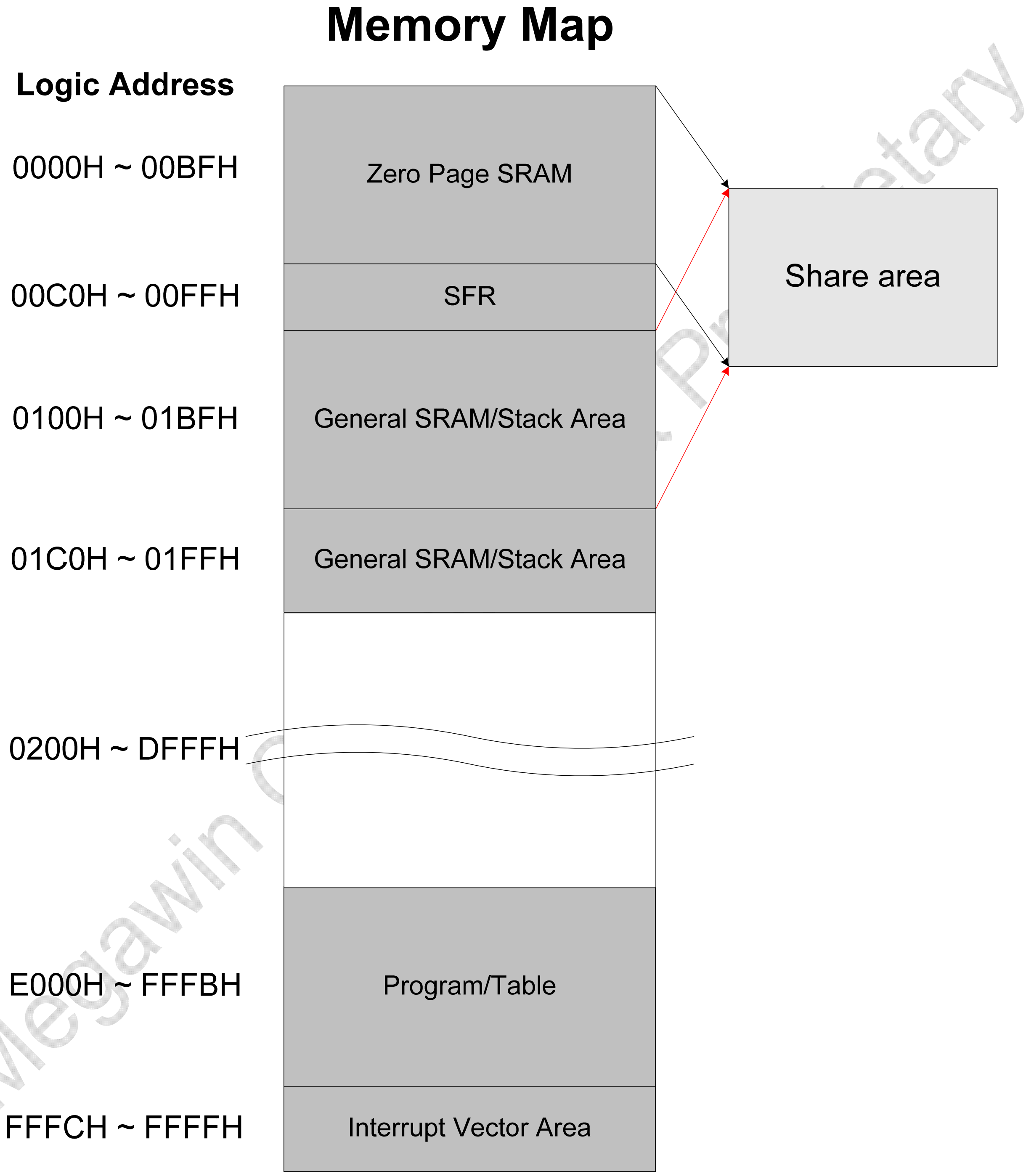


Figure 6-1 Memory Map

6.1 SFR Mapping

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

6.1.1 SFR Table

Table 6-1 SFR Table

SYMBOL	DESCRIPTION	ADDR	BIT ADDRESS & SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
Interrupt											
IRQ_EN	Interrupt Request Enable flag	**00C0H**	--	--	P33	P32	TM1	TM0	USB	SPI	xx00 0000B(R/W)
IRQ_STS	Interrupt Request Status flag	**00C1H**	--	--	P33	P32	TM1	TM0	USB	SPI	xx00 0000B(R)
IRQ_CLR	Interrupt Request Clear flag	**00C1H**	--	--	P33	P32	TM1	TM0	--	SPI	xxxx xxxxB(W)
Timer 0 / PWM0											
TM0	Timer0 Buffer	**00C3H**	T07	T06	T05	T04	T03	T02	T01	T00	1111 1111B(W)
	Timer0 Counter Register	**00C3H**	T07	T06	T05	T04	T03	T02	T01	T00	1111 1111B(R)
TM0_CTL	Timer0 Control Register	**00C4H**	ENT0	T0RL	--	--	--	T0K2	T0K1	T0K0	00xx x000B(R/W)
Timer 1 / PWM1											
TM1	Timer1 Buffer	**00C5H**	T17	T16	T15	T14	T13	T12	T11	T10	1111 1111B(W)
	Timer1 Counter Register	**00C5H**	T17	T16	T15	T14	T13	T12	T11	T10	1111 1111B(R)
TM1_CTL	Timer1 Control Register	**00C6H**	ENT1	T1RL	--	--	--	T1K2	T1K1	T1K0	00xx x000B(R/W)
Power & Clock											
PWR_CTL	Power Control Register	**00C8H**	--	ICKO1	ICKO0	CKS0	ENPS2	ENUSB	STOP	HALT	x000 0000B(W)
Reset											
RST_TRG	Reset Trigger source Register	**00C9H**	POF	EXRF	SWRF	WRF	--	--	LVDF	--	xxxx xxxxB(R/W)
RST_CTL	Reset Control Register	**00CAH**	--	USBR	--	--	--	P4RST	0	--	x0xx x00xB(R)
			SWR	USBR	--	--	--	P4RST	0	--	x0xx x00xB(W)
Watch Dog Reset											
WDT_ST	WDT Setup Register	**00CDH**	--	--	--	--	--	--	PS1	PS0	xxxx xx00B(R)
			CLR	--	--	--	--	--	PS1	PS0	xxxx xxxxB(W)
I/O Port Control											
WKPS	Wakeup port Source Register	**00D0H**	P33R	P32R	WKP3	WKP2	WKP1	WKP0	PR41	PR40	0000 0011B (R/W)
PR_EN1	Pull-up Resistor Enable Register (50KΩ)	**00D1H**	PR31	PR30	PR21	PR20	PR11	PR10	PR01	PR00	1111 1111B (R/W)
PR_EN2	Pull-up Resistor Enable Register (3MΩ)	**00D2H**	--	--	PRM21	PRM20	PRM11	PRM10	PRM01	PRM00	xx00 0000B(R/W)
MFR	Port Multi-Function Control Register	**00D8H**	--	--	--	--	T1M1	T1M0	T0M1	T0M0	000x 0000B(R/W)
I/O Port Data											
P0	Port0 output Buffer	**00D3H**	P07	P06	P05	P04	P03	P02	P01	P00	1111 1111B(W)
	Port0 input Pad	**00D3H**	P07	P06	P05	P04	P03	P02	P01	P00	xxxx xxxxB(R)
P1	Port1 output Buffer	**00D4H**	P17	P16	P15	P14	P13	P12	P11	P10	1111 1111B(W)
	Port1 input Pad	**00D4H**	P17	P16	P15	P14	P13	P12	P11	P10	xxxx xxxxB(R)
P2	Port2 output Buffer	**00D5H**	P27	P26	P25	P24	P23	P22	P21	P20	1111 1111B(W)

SYMBOL	DESCRIPTION	ADDR	BIT ADDRESS & SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
	Port2 input Pad	**00D5H**	P27	P26	P25	P24	P23	P22	P21	P20	xxxx xxxxB(R)
P3	Port3 output Buffer	**00D6H**	--	P36	P35	P34	P33	P32	P31	P30	x111 111B(W)
	Port3 input Pad	**00D6H**	--	P36	P35	P34	P33	P32	P31	P30	xxxx xxxxB(R)
P4	Port4 output Buffer	**00D7H**	--	--	P45	P44	--	P42	P41	P40	xx11 x111B(W)
	Port4 input Pad	**00D7H**	--	--	P45	P44	--	P42	P41	P40	xxxx xxxxB(R)
USB											
USB_CTL	USB Control Register	**00D9H**	--	--	--	--	--	--	UWT	URD	xxxx xx00B(R/W)
USB_ADDR	USB SFR Address Register	**00DAH**	--	--	UA5	UA4	UA3	UA2	UA1	UA0	xx00 0000B(W)
USB_DI	USB SFR Data Input Register	**00DBH**	UDI7	UDI6	UDI5	UDI4	UDI3	UDI2	UDI1	UDI0	xxxx xxxxB(W)
USB_DO	USB SFR Data Output Register	**00DBH**	UDO7	UDO6	UDO5	UDO4	UDO3	UDO2	UDO1	UDO0	xxxx xxxxB(R)
DPM (PS/2)											
DPMO	DP/DM Output Data Buffer Register	**00DDH**	--	--	--	--	--	--	DPO	DMO	xxxx xx00(W)
DPMI	DP/DM Input Pad	**00DDH**	--	--	--	--	--	--	DPI	DMI	xxxx xx00(R)
Protect Write											
PWPR	Protect Write Pattern Register	**00DFH**	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0	xxxx xxxxB(W)
In Application Programming (IAP)											
IAP_PR	IAP Write Protect Register	**00E0H**	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	xxxx xxxxB(W)
SPI Interface											
SPICTL	Serial Interface Control Register	00E7H	SPIEN	SCLK	DORD	OPD	--	--	--	--	0000 xxxxB(R/W)
SPIDAT	SIDAT : SPI_MOSI	00E8H	(MSB)							(LSB)	0000 0000B(W)
	SIDAT : SPI_MISO	00E8H	(MSB)							(LSB)	0000 0000B(R)

Note: PWPR will protect address from 00C8H ~ 00CFH.

7 Configurable I/O Ports

This chip has 5 IO ports, Port0~Port4(P0[7:0], P1[7:0], P2[7:0], P3[6:0], P4[5,4], P4[2:0], P35 and P36 shared with XTAL_IN/OUT), total 34 + 2 programmable IO and user can select enable/disable internal pull-up resistor. User should be careful on setting pin as input with no pull high resistor since this setting has potential to cause leakage. User can set different voltage to GPIO by VDDO pin. The IO structure is illustrated as the following figure.

7.1 IO Structure Diagram

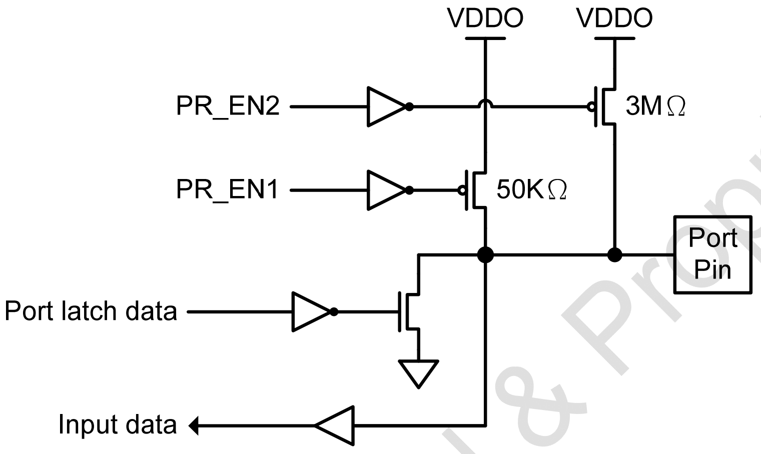


Figure 7-1 GPIO Diagram

7.2 IO Port Register

Firmware can read Px(x=0~4) to get the data on each pin and write Px(x=0~4) to output the data on each pin.

When P32 and P33 are set as input pins, they are external interrupt sources. A falling/rising edge which setup by P33R and P32R bits in WKPS register at these two pins will set corresponding IRQ_STS bits to 1, and their interrupt subroutines will be executed if corresponding IRQ_EN bits are set.

Note : P04 is shared with F_{OSC} clock output.
Note : P11 is multi-function with Timer 0 underflow / PWM0 output.
Note : P12 is multi-function with Timer 1 underflow / PWM1 output.
Note : P20 / P21 / P25 are shared with SPI interface.
Note : P32 and P33 are external interrupt sources.
Note : P35 and P36 are shared with XTAL_IN/OUT function if ENRCO = 0 (Hardware Option).
Note : P40~P42, P44, P45 are LED IO.

P0(Port 0 input/output register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D3H	P0 output Buffer	P07	P06	P05	P04	P03	P02	P01	P00		√	1111 1111B
00D3H	P0 input Pad	P07	P06	P05	P04	P03	P02	P01	P00	√		xxxx xxxxB

Bit[7:0] : P0[7:0] -- Port 0 input/output data.

P1(Port 1 input/output register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D4H	P1 output Buffer	P17	P16	P15	P14	P13	P12	P11	P10		√	1111 1111B
00D4H	P1 input Pad	P17	P16	P15	P14	P13	P12	P11	P10	√		xxxx xxxxB

Bit[7:0] : P1[7:0] -- Port 1 input/output data.

P2(Port 2 input/output buffer register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D5H	P2 output Buffer	P27	P26	P25	P24	P23	P22	P21	P20		√	1111 1111B
00D5H	P2 input Pad	P27	P26	P25	P24	P23	P22	P21	P20	√		xxxx xxxxB

Bit[7:0] : P2[7:0] -- Port 2 input/output data.

P3 (Port 3 input/output register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D6H	P3 output Buffer	--	P36	P35	P34	P33	P32	P31	P30		√	x111 1111B
00D6H	P3 input Pad	--	P36	P35	P34	P33	P32	P31	P30	√		xxxx xxxxB

Bit7 : Reserved
Bit[6:0] : P3[6:0] -- Port 3 input/output data.

P4 (Port 4 input/output register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D7H	P4 output Buffer	--	--	P45	P44	--	P42	P41	P40		√	xx11 x111B
00D7H	P4 input Pad	--	--	P45	P44	--	P42	P41	P40	√		xxxx xxxxB

Bit7 and Bit6 : Reserved
Bit[5:4] : P4[5:4] -- P4.5 and P4.4 input/output data.
Bit3 : Reserved
Bit[2:0] : P4[2:0] -- P4.2, P4.1 and P4.0 input/output data.
Note : Reset source is selected by P4RST setting.

PR_EN1(Pull-up Resistor Enable register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D1H	PR_EN1	PR31	PR30	PR21	PR20	PR11	PR10	PR01	PR00	√	√	1111 1111B

Bit[7:0] : PR_EN1[7:0] -- 50KΩ Pull-up Resistor Control Register.
0: disable.
1: enable. (default)
PR31: P34~P37 50KΩ Pull-up Resistor Enable bit.
PR30: P30~P33 50KΩ Pull-up Resistor Enable bit.
PR21: P24~P27 50KΩ Pull-up Resistor Enable bit.
PR20: P20~P23 50KΩ Pull-up Resistor Enable bit.
PR11: P14~P17 50KΩ Pull-up Resistor Enable bit.
PR10: P10~P13 50KΩ Pull-up Resistor Enable bit.
PR01: P04~P07 50KΩ Pull-up Resistor Enable bit.
PR00: P00~P03 50KΩ Pull-up Resistor Enable bit.
Note :The pull-up resistor is enable in default.

PR_EN2(Pull-up Resistor Enable register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D2H	PR_EN2	--	--	PRM21	PRM20	PRM11	PRM10	PRM01	PRM00	√	√	xx00 0000B

Bit7 and Bit6 : Reserved
Bit[5:0] : PR_EN2[5:0] -- 3MΩ Pull-up Resistor Control Register.
0: disable.(Default)
1: enable.
PRM21: P2[4:7] 3MΩ Pull-up Resistor Enable bit.
PRM20: P2[0:3] 3MΩ Pull-up Resistor Enable bit.
PRM11: P1[4:7] 3MΩ Pull-up Resistor Enable bit.
PRM10: P1[0:3] 3MΩ Pull-up Resistor Enable bit.
PRM01: P0[4:7] 3MΩ Pull-up Resistor Enable bit.
PRM00: P0[0:3] 3MΩ Pull-up Resistor Enable bit.

MFR(Multi Function Control Register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D8H	MFR	--	--	--	--	T1M1	T1M0	T0M1	T0M0	√	√	000x 0000B

Bit[7: 4] : Reserved.

Bit [3:2] : T1M[1:0] -- Timer1 / PWM1 output function.

T1M1	T1M0	P1.2 function
0	0	Normal GPIO (Default)
0	1	Timer1 under flow toggle P1.2 (T1CKO)
1	0	PWM1 output to P1.2
1	1	Reserved

Bit [1:0] : T0M[1:0] -- Timer0 / PWM0 output function.

T0M1	T0M0	P1.1 function
0	0	Normal GPIO (Default)
0	1	Timer0 under flow toggle P1.1 (T0CKO)
1	0	PWM0 output to P1.1
1	1	Reserved

WKPS(Wakeup Port Select register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D0H	WKPS	P33R	P32R	WKP3	WKP2	WKP1	WKP0	PR41	PR40	√	√	0000 0011B

Bit7: P33R -- P33 Rising edge interrupt/wakeup.
0: Falling edge wakeup/interrupt (default).
1: Rising edge wakeup/interrupt.

Bit6: P32R -- P32 Rising edge interrupt/wakeup.
0: Falling edge wakeup/interrupt (default).
1: Rising edge wakeup/interrupt.

Bit[5:2] : WKP[3:0] -- Wakeup Port enable.
0: disable. (default)
1: enable.
WKP3: Port 3 low-level wakeup enable.
WKP2: Port 2 low-level wakeup enable.
WKP1: Port 1 low-level wakeup enable.
WKP0: Port 0 low-level wakeup enable.

WKP3	IRQ_EN P32 , P33	P30, P31, P34~P36 (Low-level)	P32		P33	
			P32R=0 (Falling Edge)	P32R=1 (Rising Edge)	P32R=0 (Falling Edge)	P32R=1 (Rising Edge)
0	0 , 0	No wakeup	No Wakeup	No Wakeup	No Wakeup	No Wakeup
0	0 , 1	No wakeup	No Wakeup	No Wakeup	Wakeup **with interrupt**	Wakeup **with interrupt**
0	1 , 0	No wakeup	Wakeup **with interrupt**	Wakeup **with interrupt**	No Wakeup	No Wakeup
0	1 , 1	No wakeup	Wakeup **with interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**
1	0 , 0	Wakeup **No interrupt**	Wakeup **No interrupt**	Wakeup **No interrupt**	Wakeup **No interrupt**	Wakeup **No interrupt**
1	0 , 1	Wakeup **No interrupt**	Wakeup **No interrupt**	Wakeup **No interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**
1	1 , 0	Wakeup **No interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**	Wakeup **No interrupt**	Wakeup **No interrupt**
1	1 , 1	Wakeup **No interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**	Wakeup **with interrupt**

Blue color: low-level trigger
Green color: by P32R/P33R setting

Bit1: PR41 -- P44~P45 50KΩ Pull-up Resistor Enable bit.
0: disable.
1: enable. (default)

Bit0: PR40 -- P40~P42 50KΩ Pull-up Resistor Enable bit.
0: disable.
1: enable. (default)
Note : The PR40 and PR41 pull-up resistor is enable in default. Reset source is selected by P4RST setting.

8 Interrupt

There are 1 interrupt sources provided in this chip. The flag **IRQ_EN** and **IRQ_STS** are used to control the interrupts. When any flag in **IRQ_STS** register is set to '1' by hardware and the corresponding bits of flag **IRQ_EN** has been set by firmware, an interrupt is generated. When an interrupt occurs, all of the interrupts are inhibited until the **CLI** or **STA IRQ_EN, #I** instruction is invoked. Executing the **SEI** instruction can also disable the interrupts.

Vector Address	item	Priority	Properties	Description
FFFCH, FFFDH	RESET	1	Ext.	Initial reset
FFFEH, FFFFH	INT	2	Int.	Interrupt vector

Note : The RESET interrupt include: External reset, LVR, POR, WDT, SWR, IAR.

8.1 Interrupt Register

IRQ_EN (Interrupt Request Enable flag)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C0H	IRQ_EN	--	--	P33	P32	TM1	TM0	USB	SPI	√	√	xx00 0000B

Bit7 and Bit6 : Reserved

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

0: Disable (default "0" at initialization)

1: Enable

Bit5 : P33 -- Enable P3.3 falling/rising edge interrupt.
Bit4 : P32 -- Enable P3.2 falling/rising edge interrupt
Bit3 : TM1 -- Enable Timer1 interrupt.
Bit2 : TM0 -- Enable Timer0 interrupt.
Bit1 : USB -- Enable USB interrupt.
Bit0 : SPI -- Enable SPI interrupt.

IRQ_STS (Interrupt Request Status flag)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C1H	IRQ_STS	--	--	P33	P32	TM1	TM0	USB	SPI	√		xx00 0000B
00C1H	IRQ_CLR	--	--	P33	P32	TM1	TM0	--	SPI		√	xxxx xxxxB

Bit7 and Bit6 : Reserved

When IRQ occurs, program can read this register to know which source triggering IRQ. Firmware can clear the interrupt event by writing "1" into the corresponding bit. USB interrupt flag is included in USB SFR.

0: default value

1: Interrupt event trigger set by hardware. Firmware write "1" to clear this bit.

Bit5 : P33 -- P33 falling/rising edge occurs.
Bit4 : P32 -- P32 falling/rising edge occurs.
Bit3 : TM1 -- Timer1 underflow.
Bit2 : TM0 -- Timer0 underflow.
Bit1 : USB -- USB finished RX or TX data.
Bit0 : SPI -- SPI finished RX and TX data.

8.2 Interrupt System

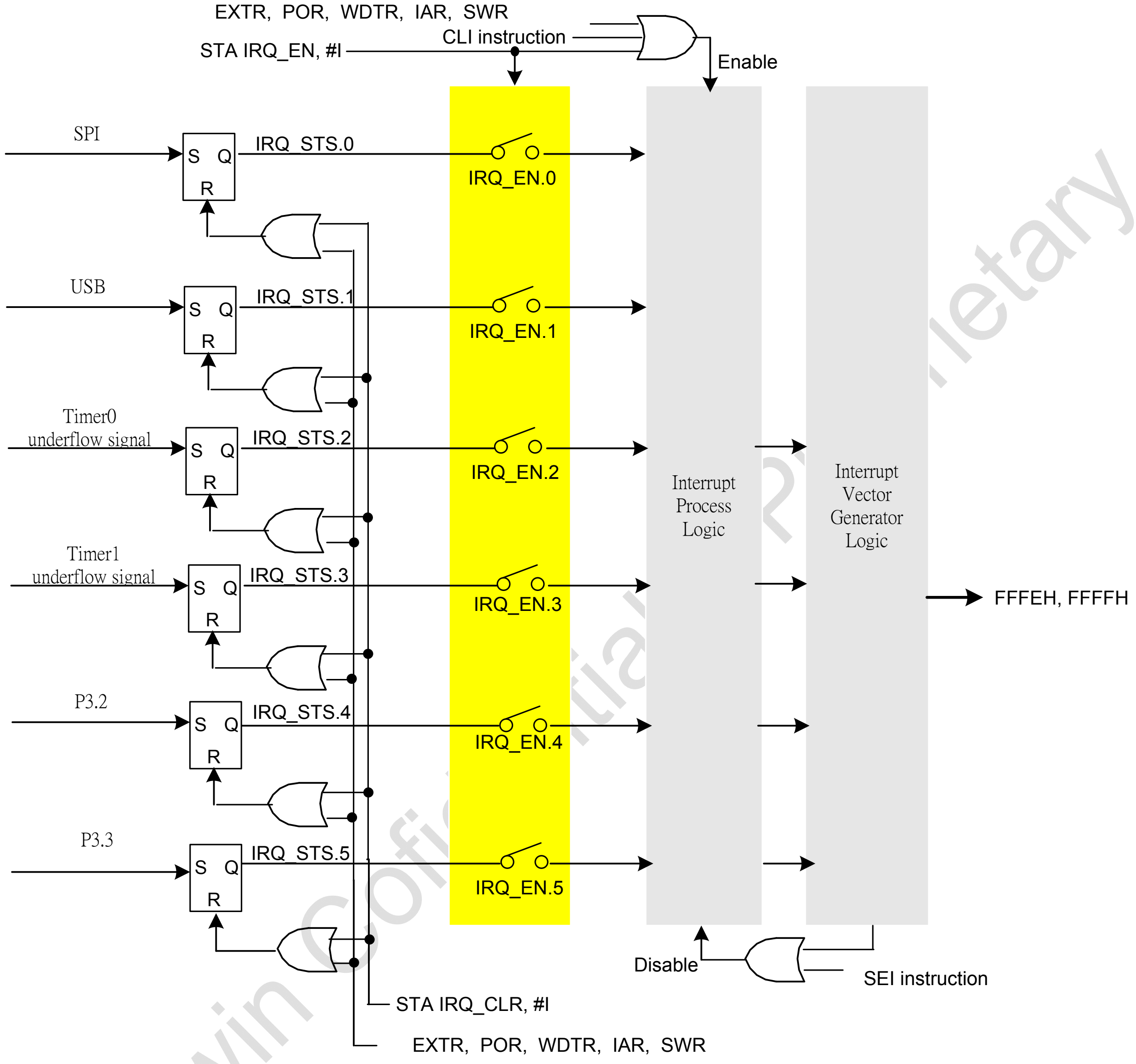


Figure 8-1 Interrupt Diagram

9 Timers / PWM

There are two 8-bit timers on this chip.

9.1 8-bit Timer0 / PWM0

Timer 0 is a 8-bit counter. It can be a programmable down-count counter. When timer 0 is under user's firmware control, it will pre-load value to counter at the rising edge of TM0_CTL.ENT0 and its underflow frequency of timer 0 can be calculated with the following equation:

$$F_{TM0\_UV} = F_{TM0CK} / (TM0+1)$$

For example: if FTM0_UV = F_{OSC}/32 = 6MHz/32 = 187.5KHz

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

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

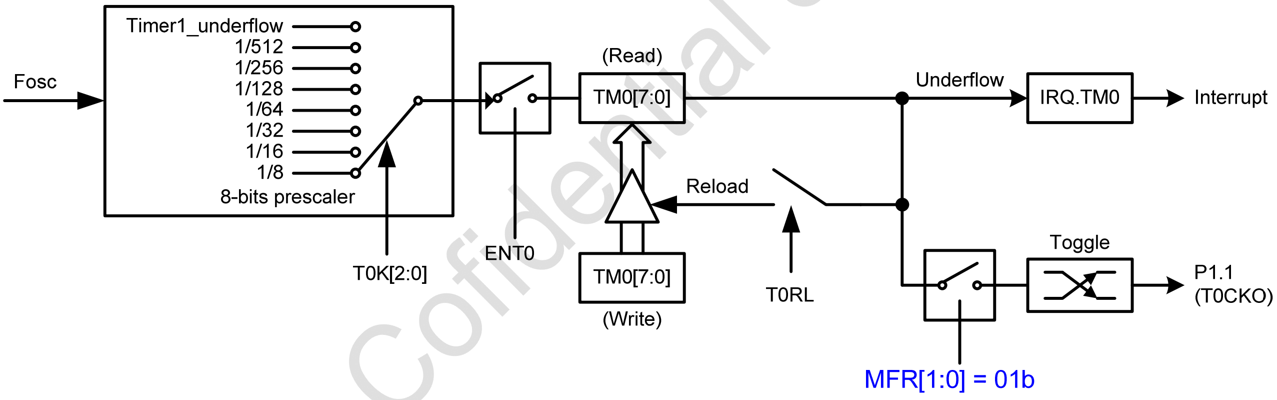


Figure 9-1 Timer0 / PWM0 Structure – Timer Mode

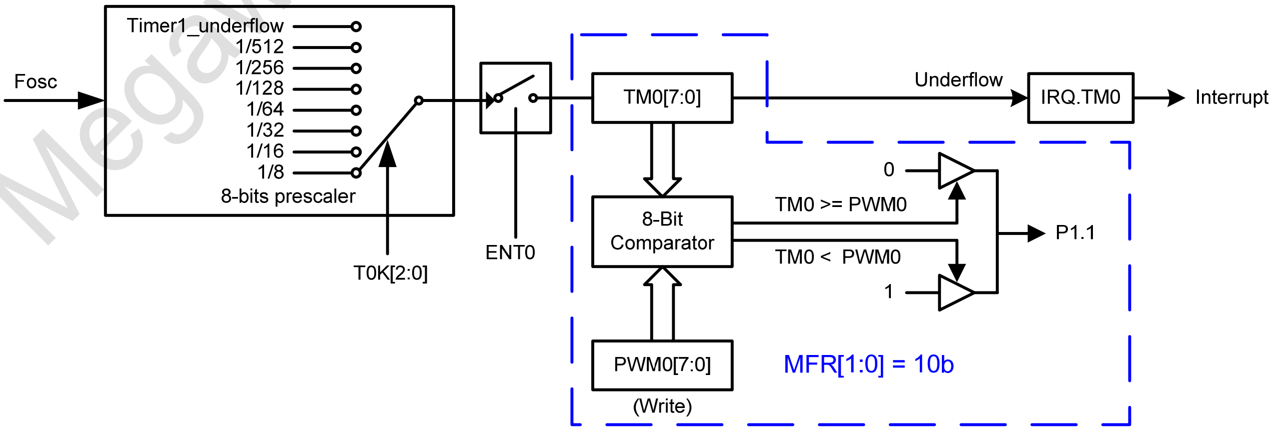


Figure 9-2 Timer0 / PWM0 Structure – PWM Mode

9.1.1 Timer0 / PWM0 register

TM0(Timer 0 count register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C3H	TM0 / PWM0	T07	T06	T05	T04	T03	T02	T01	T00		√	1111 1111B
00C3H	TM0 counter	T07	T06	T05	T04	T03	T02	T01	T00	√		1111 1111B

Bit[7:0] :T0[7:0] -- Timer 0 count value

TM0_CTL(Timer 0 Control register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C4H	TM0_CTL	ENT0	T0RL	--	--	--	T0K2	T0K1	T0K0	√	√	00xx x000B

Bit7 : ENT0 -- Timer0 / PWM0 clock disable/enable.
0: Disable. (Default)
1: Enable.

Bit6 : T0RL -- Timer0 auto-reload enable/disable.
0: Enable. (Default)
1: Disable.

Bit[5:3] : Reserved

Bit[2:0] : T0K[2:0] -- Timer0 / PWM0 clock source selector.

T0K2	T0K1	T0K0	Selected TM0 input clock source
0	0	0	F_{OSC} / 8 (Default)
0	0	1	F_{OSC} / 16
0	1	0	F_{OSC} / 32
0	1	1	F_{OSC} / 64
1	0	0	F_{OSC} / 128
1	0	1	F_{OSC} / 256
1	1	0	F_{OSC} / 512
1	1	1	Timer 1 underflow

9.2 8-bit Timer1 / PWM1

Timer 1 is a 8-bit counter. It can be a programmable down-count counter. When timer 1 is under user's firmware control, it will pre-load value to counter at the rising edge of TM1_CTL.ENT1 and its underflow frequency of timer 1 can be calculated with the following equation:

$$F_{TM1\_UV} = F_{TM1CK} / (TM1+1)$$

For example: if FTM1_UV = F_{OSC}/32 = 6MHz/32 = 187.5KHz

TM1	F_{TM1_UV} Frequency
00H	invalid
01H	93.75 KHz
02H	62.5 KHz
…	…
FFH	732.42 Hz

Writing data to the TM1 would write data to the reload buffer of the timer 1. Reading data from the TM1 would read the run-time value from the counter.

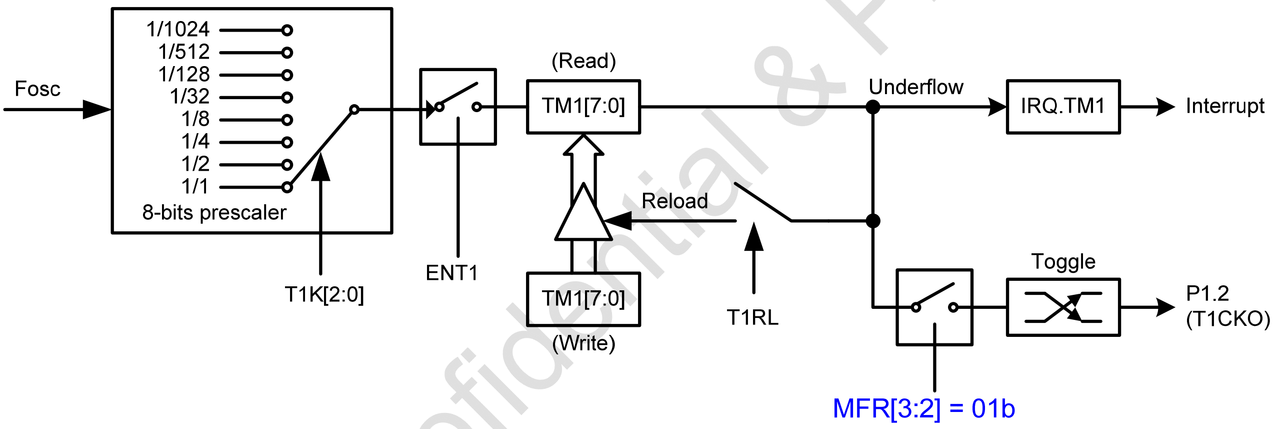


Figure 9-3 Timer1 / PWM1 Structure – Timer Mode

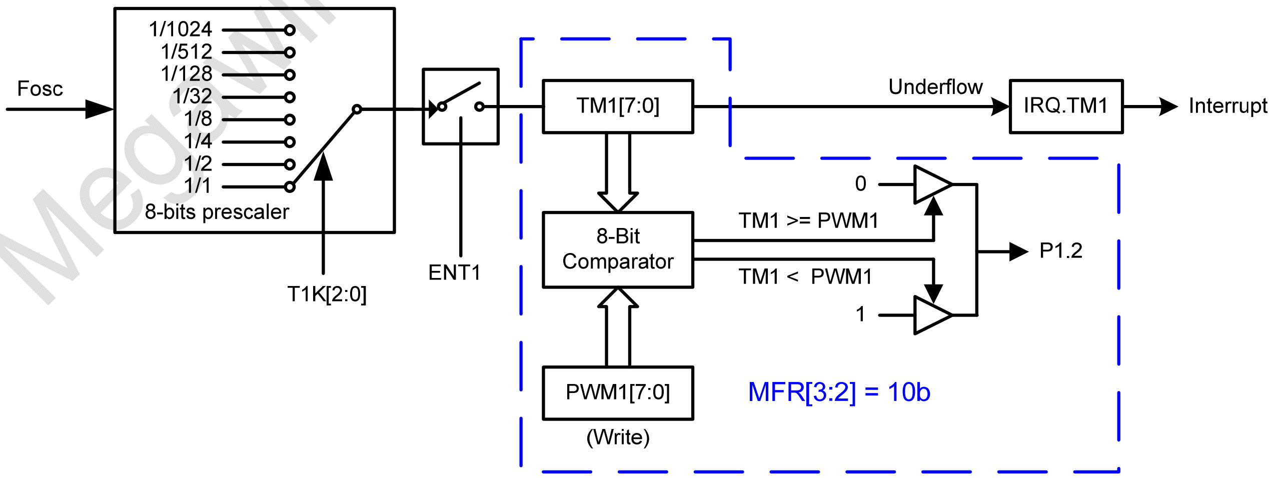


Figure 9-4 Timer1 / PWM1 Structure – PWM Mode

9.2.1 Timer1 / PWM1 register

TM1(Timer 1 count register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C5H	TM1 / PWM1	T17	T16	T15	T14	T13	T12	T11	T10		√	1111 1111B
00C5H	TM1 Counter	T17	T16	T15	T14	T13	T12	T11	T10	√		1111 1111B

Bit[7:0] : T1[7:0] -- Timer1 count value.

TM1_CTL(Timer 1 Control register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C6H	TM1_CTL	ENT1	T1RL	--	--	--	T1K2	T1K1	T1K0	√	√	00xx x000B

Bit7 : ENT1 -- Timer1 / PWM1 clock disable/enable.
0: Disable. (Default)
1: Enable.

Bit6 : T1RL -- Timer1 auto-reload enable/disable.
0: Enable. (Default)
1: Disable.

Bit[5:3] : Reserved

Bit[2:0] : T1K[2:0] -- Timer1 / PWM1 clock source selector.

T1K2	T1K1	T1K0	Selected TM0 input clock source
0	0	0	F_{OSC} / 1 (Default)
0	0	1	F_{OSC} / 2
0	1	0	F_{OSC} / 4
0	1	1	F_{OSC} / 8
1	0	0	F_{OSC} / 32
1	0	1	F_{OSC} / 128
1	1	0	F_{OSC} / 512
1	1	1	F_{OSC} / 1024

10 SPI Serial Interface

There is one 8-bit SPI interface on this chip. Share Interface pin to GPIO, SPI_SCLK(P25), SPI_MISO(P21), SPI_MOSI(P20).

10.1 SFR Control Register

SPIDAT (Serial Data TX/RX Buffer)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00E8H	SPIDAT : SPI_MOSI	(MSB)							(LSB)		√	0000 0000B
00E8H	SPIDAT : SPI_MISO	(MSB)							(LSB)	√		0000 0000B

Write data to this SFR will start transfer data to serial output pad. (P20)
Read data from this SFR will always read data from serial input pad. (P21)

SPICTL(SPI Serial Interface Control Register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00E7H	SPICTL	SPIEN	SCLK	DORD	OPD	--	--	--	--	√	√	0000 xxxxB

Bit7 : SPIEN – SPI Mode Enable/Disable Control Register.
0 : Disable. (Default)
1 : Enable.

Bit6 : SCLK -- SPI Clock Select.
0 : 1.5MHz bit rate. (Default)
1 : 750KHz bit rate.

Bit5 : DORD -- SPI data order.
0 : The MSB of the data byte is transmitted first.(Default)
1 : The LSB of the data byte is transmitted first.

Bit4 : OPD -- SPI interface open-drain select.
0 : CMOS output for MOSI and SCLK.(Default)
1 : NMOS(Open-drain) output for MOSI and SCLK.
Note : Set OPD=0 will auto disable MOSI/MISO/SCLK internal 50KΩ Pull-up Resistor.

Bit[3:0] : Reserved

11 Reset

There are six reset sources for whole chip as list below.

- Power-on Reset
- External Reset
- Watch-Dog-Timer Reset
- Software Reset
- USB Macro Reset (USBR)

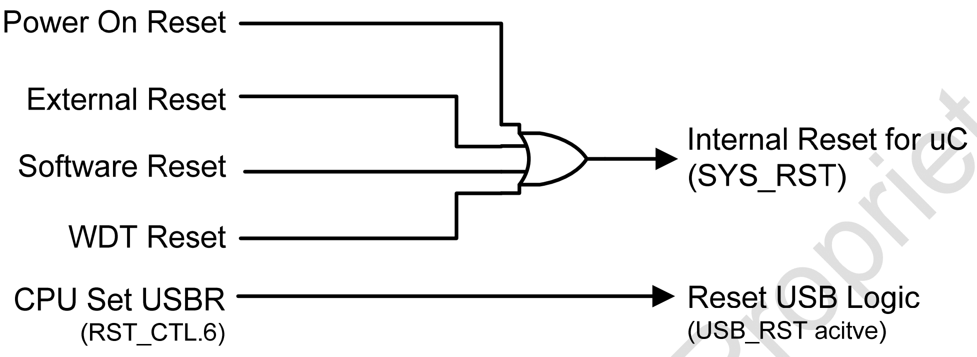


Figure 11-1 Reset Source

11.1 Reset Register

RST_TRG(Reset Trigger source register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C9H	RST_TRG	POF	EXRF	SWRF	WRF	--	--	LVDF	--	√	√	xxxx xxxxB

Note : RST_TRG can be write by firmware only when PWPR is equal to “5AH”.

Bit7: POF -- Power-on Reset Flag.
0: This bit must be clear by firmware write “1”.
1: This bit is set if a Power-on reset or LVR occurs.

Bit6: EXRF -- External Reset Flag.
0: This bit must be clear by firmware write “1”.
1: This bit is set if a external reset occurs.

Bit5: SWRF -- Software Reset Flag.
0: This bit must be clear by firmware write “1”.
1: This bit is set if a software reset occurs.

Bit4: WRF -- WDT Reset Flag.
0: This bit must be clear by firmware write “1”.
1: This bit is set by hardware if a WDT reset occurs.

Bit[3:2]: Reserved

Bit1: LVDF -- Low voltage Detection Flag.
0: This bit must be clear by firmware write “1”. (Default)
1: This bit is set by hardware when VDD had drop below the V_{LVF}

Bit0: Reserved

RST_CTL(Reset Control register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00CAH	RST_CTL	--	USBR	--	--	--	P4RST	0	--	√		x0xx x00xB
		SWR	USBR	--	--	--	P4RST	0	--		√	x0xx x00xB

Note : RST_CTL can be write by firmware only when PWPR is equal to “5AH”.

Bit7 : SWR -- Soft-ware Reset.
0: no effect
1: Firmware write “1” to trigger a soft-ware reset event to reset chip.

Bit6 : USBR -- USB Reset.
0: Firmware write “0” to finish USB module Reset. (Default)
1: Firmware write “1” to start reset USB module.

Bit[5:3] : Reserved

Bit2 : P4RST -- Port 4 Reset status Flag.
0: P4 output buffer will be cleared by POR, External reset, Software reset, WDT reset, illegal address reset. (Default)
1: P4 output buffer will be cleared by POR, External reset.

Bit1 : Reserved, software must write “0” on these bits.

Bit0 : Reserved

11.2 Watchdog Timer Reset

WDT_ST (WDT Setup register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00CDH	WDT_ST	--	--	--	--	--	--	PS1	PS0	√	√	xxxx xx00B
		CLR	--	--	--	--	--	--	--		√	xxxx xxxxB

Note: This SFR is protected by **PWPR**.

Bit7 : CLR-- WDT Clear bit.
Write “1” to this bit will clear WDT.
Write “0” has no effect.

Bit[6:2] : Reserved.

Bit[1:0] : PS[1:0] -- The WDT Period Selector.
00: about 21.845ms (Default)
01: about 43.690ms
10: about 87.380ms
11: about 174.762ms

12 Power

12.1 Regulator

There is a built-in 5V to 3.3V regulator to supply USB transceiver power.

12.2 HALT Mode(Idle Mode)

Setting the HALT bit in PWR_CTL enters idle mode. Idle mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The uC can be awakened from halt mode by the following ways:

- Interrupts (USB, Timer0, Timer1, INT0, INT1) assigned in IRQ_EN.
- IO wakeup assigned in WKPS register with low-level.
- WDT reset
- External reset

12.3 STOP Mode(Power-Down Mode)

Setting the STOP bit in PWR_CTL register enters STOP(power-down) mode. STOP mode stops the oscillator circuit to minimize power consumption. The uC can be awakened from stop mode by the following ways:

- Interrupts (USB, INT0, INT1) assigned in IRQ_EN.
- IO wakeup assigned in WKPS register with low-level.
- External reset

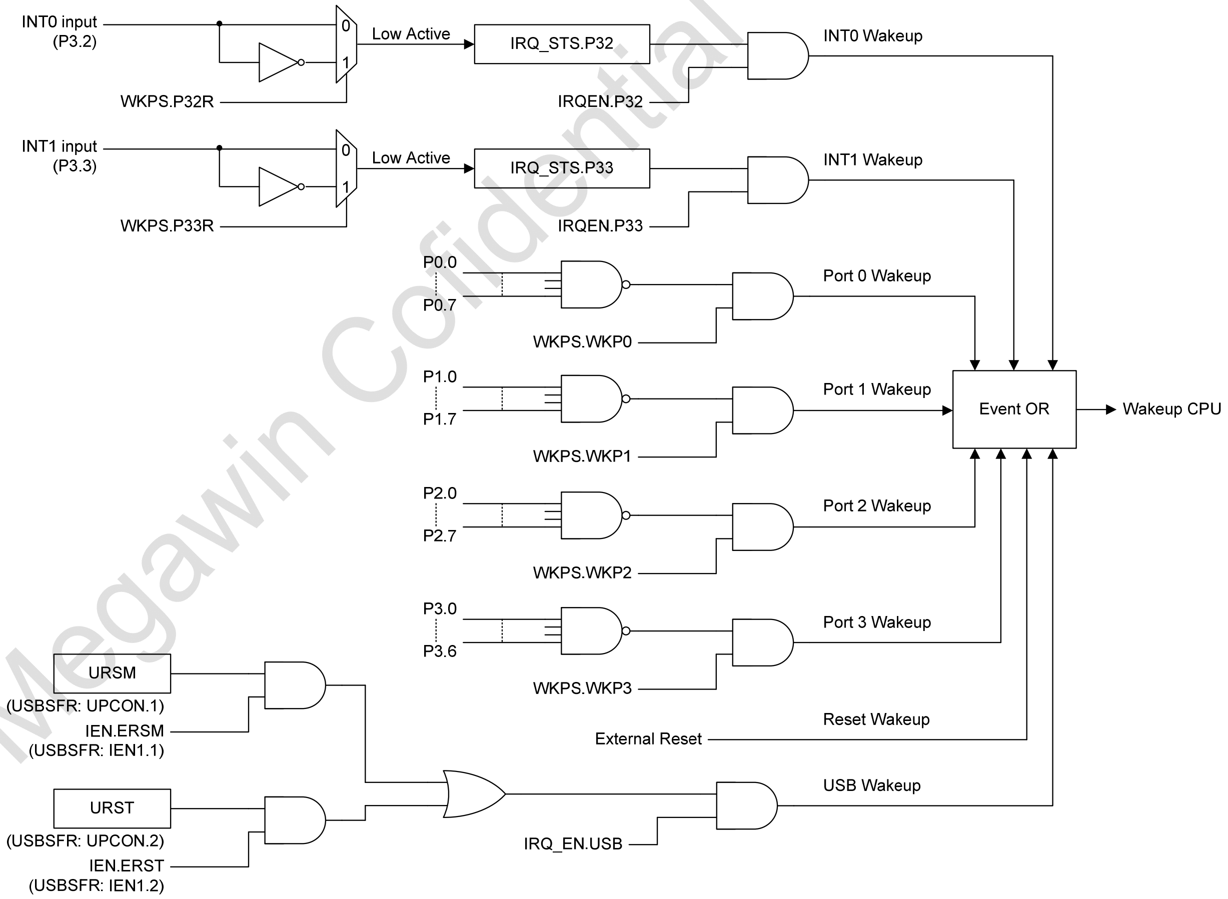


Figure 12-1 Stop Mode Wakeup Source

12.4 Power Control Register

Program can switch the normal operation mode to the power-saving mode for saving power consumption through the following register.

PWR_CTL(Power Control register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00C8H	PWR_CTL	--	ICKO1	ICKO0	CKS0	ENPS2	ENUSB	STOP	HALT		√	x000 0000B

Note : PWR_CTL can be write by firmware only when PWPR is equal to "5AH".

Bit7 : Reserved

Bit[6:5] : ICKO[1:0] -- Internal Clock Output

ICKO1	ICKO0	Clock Output to P04
0	0	Disable (Default)
0	1	Output F_{OSC} (6MHz)
1	0	Output F_{OSC} /2 (3MHz)
1	1	Output F_{OSC} /4 (1.5MHz)

Bit4 : CKS0 -- F_{CPU} Clock divider Selector.
0: F_{CPU} = F_{OSC} (default).
1: F_{CPU} = F_{OSC} /2

Bit3 : ENPS2 -- Enable PS/2
0: Disable clock of PS/2 module. (Default)
1: Enable clock of PS/2 module.

Bit2 : ENUSB -- Enable USB Clock
0: Disable clock of USB module. (Default)
1: Enable clock of USB module.

Table 12-1 USB/PS2 control table

ENPS2	ENUSB	FSE0(in USB SFR)	DM/DP
1	X	X	PS2 mode
0	1	1	Force SE0
0	1	0	USB mode
0	0	X	Reserved

Bit1 : STOP -- chip will into STOP mode (power-down mode)
0: enable OSC (Default)
1: disable OSC

Bit0 : Halt -- FCPU off-line control bit.
0: FCPU on-line (Default)
1: FCPU off-line

PWPR(Protect Write Pattern register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00DFH	PWPR	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0		√	xxxx xxxxB

When this byte is wrote by firmware, it would be automatically cleared by hardware after the "next write action" of firmware. Before setting **RST_CTL** / **WDT_ST** / **PWR_CTL**, user must write **0x5A** to **PWPR**. Write PWPR is only enable next write instruction.

Bit[7:0] : PT[7:0] -- Protect Write Pattern.

Sample Code: Trigger software reset on MCU.

```
        LDA     #5Ah
        STA     PWPR
        LDA     #80h
        STA     RST_CTL         ;Software Reset
```

13 System Clock

There are two clock source in MG64F237, one is external clock source like resonator/crystal (EXTOSC) and two is build-in 6MHz oscillator (IHRCO). Which clock source into F_{OSC} is decided by **ENRCO** bit. For F_{CPU} clock, it would be divided by 2 if CKS0 is set by firmware.

After power on, the clock would be de-bounce 16384 clocks (2.736ms).
After wakeup, the clock would be de-bounce 16384 clocks (2.736ms).

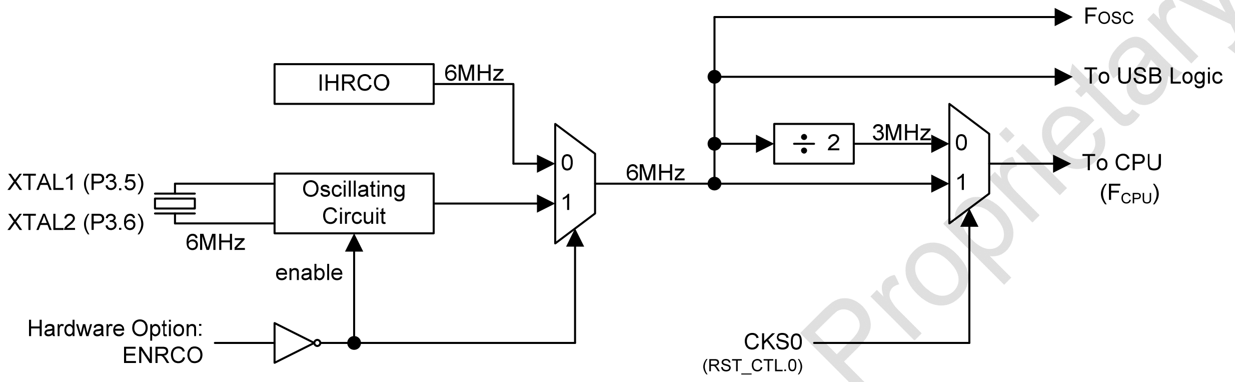


Figure 13-1 System Clock Diagram

14 DPM Control (PS/2 Control)

For USB and PS/2 combo application, the chip provides a way to control DP/PS2_CLK and DM/PS2_Data pins by user's firmware. The control focuses on PS/2 interface and in system program operations. The DPMI record the DP/PS2_CLK and DM/PS2_Data pin value respectively. For PS/2 interface, firmware can judge the DP/PS2_CLK and DM/PS2_Data pins' connection be USB or PS/2 protocol by reading the value of DPI and DMI. The ENPS2 and ENUSB in PWR_CTL register set the controller of DP and DM pins. If they are set to 10, the DP and DM pins are under firmware's control, thus the USB function is unavailable. DPMO sets the value of DP/PS2_CLK and DM/PS2_Data pins when firmware controls the DP/DM pin.

14.1 DPM Register

{ENPS2, ENUSB}: See *12.4 Power Control Register* or See *15.3.1 USB SFR R/W Procedure*

DPMO/DPMI (DP/DM Output data register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00DDH	DPMO	--	--	--	--	--	--	DPO	DMO		√	xxxx xx00B
	DPMI	--	--	--	--	--	--	DPI	DMI	√		xxxx xx00B

Note: The DPMO function are only valid in PS2 mode.

Bit[7:2] : Reserved

Bit1 : DPO -- PS2_CLK/DP output data
0: output low (Default)
1: pull-high(input mode)

Bit0 : DMO -- PS2_Data/DM output data
0: output low (Default)
1: pull-high(input mode)

Bit1 : DPI -- PS2_CLK/DP pin data
Bit0 : DMI -- PS2_Data/DM pin data

15 USB (Universal Serial Bus)

15.1 Features

- Compliant with USB specification v2.0
- Compliant with USB HID device class specification v1.11.
- USB bus-powered or self-powered option
- Supports USB suspend/resume and remote wake-up

15.2 Block Diagram

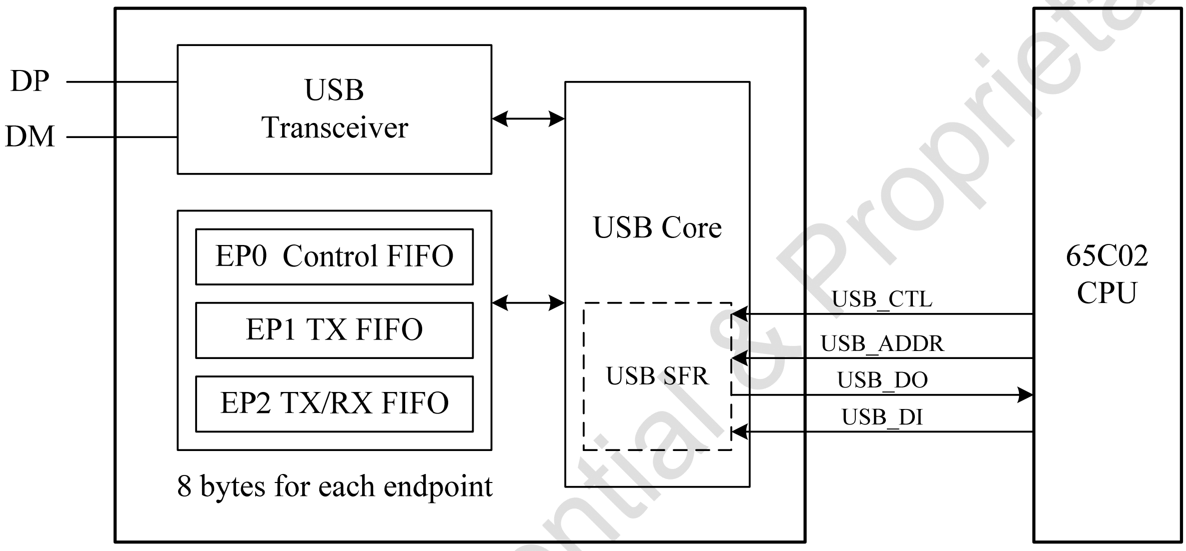


Figure 15-1 USB Macro Read/Write Interface

15.3 Special Function Registers

The USB block contains SFR of its own as description in the next page. User can access the USB SFR to implement USB operation with host. Before activating the USB operation, the user should enable USB 1.1 transceiver by ENUSB bit in PWR_CTL SFR.(See *12.4 Power Control Register*)

USB_ADDR (USB SFR Address register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00DAH	USB_ADDR	0	0	UA5	UA4	UA3	UA2	UA1	UA0		√	xx00 0000B

Bit[7:6] : Reserved. Software must write “0” on these bits.

Bit[5:0] : UA[6:0] -- USB SFR Address.

USB_DI/USB_DO (USB SFR write Data register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00DBH	USB_DI	UDI7	UDI6	UDI5	UDI4	UDI3	UDI2	UDI1	UDI0		√	0000 0000B
	USB_DO	UDO7	UDO6	UDO5	UDO4	UDO3	UDO2	UDO1	UDO0	√		0000 0000B

Bit[7:0] : UDI[7:0] -- USB SFR write Data.

Bit[7:0] : UDO[7:0] -- USB SFR read Data.

USB_CTL (USB Control register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00D9H	USB_CTL	--	--	--	--	--	--	UWT	URD		√	xxxx xx00B
		--	--	--	--	--	--	UWT	URD	√		xxxx xx00B

Bit[7:2] : Reserved.

Bit1: UWT -- USB Write trigger.

Write “1” to start USB SFR write procedure.
Read “1” : Hardware is in busy USB SFR write procedure.
Write “0” : Reserve
Read “0” : Hardware is ready for next USB SFR read procedure.(Default)

Bit0: URD -- USB Read trigger.

Write “1” to start USB SFR read procedure.
Read “1” : Hardware is in busy USB SFR read procedure.
Write “0” : Reserve
Read “0” : Hardware is ready for next USB SFR read procedure.(Default)

15.3.1 USB SFR R/W Procedure

USB Write Procedure:

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

USB Read Procedure:

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

15.3.2 USB Interrupt

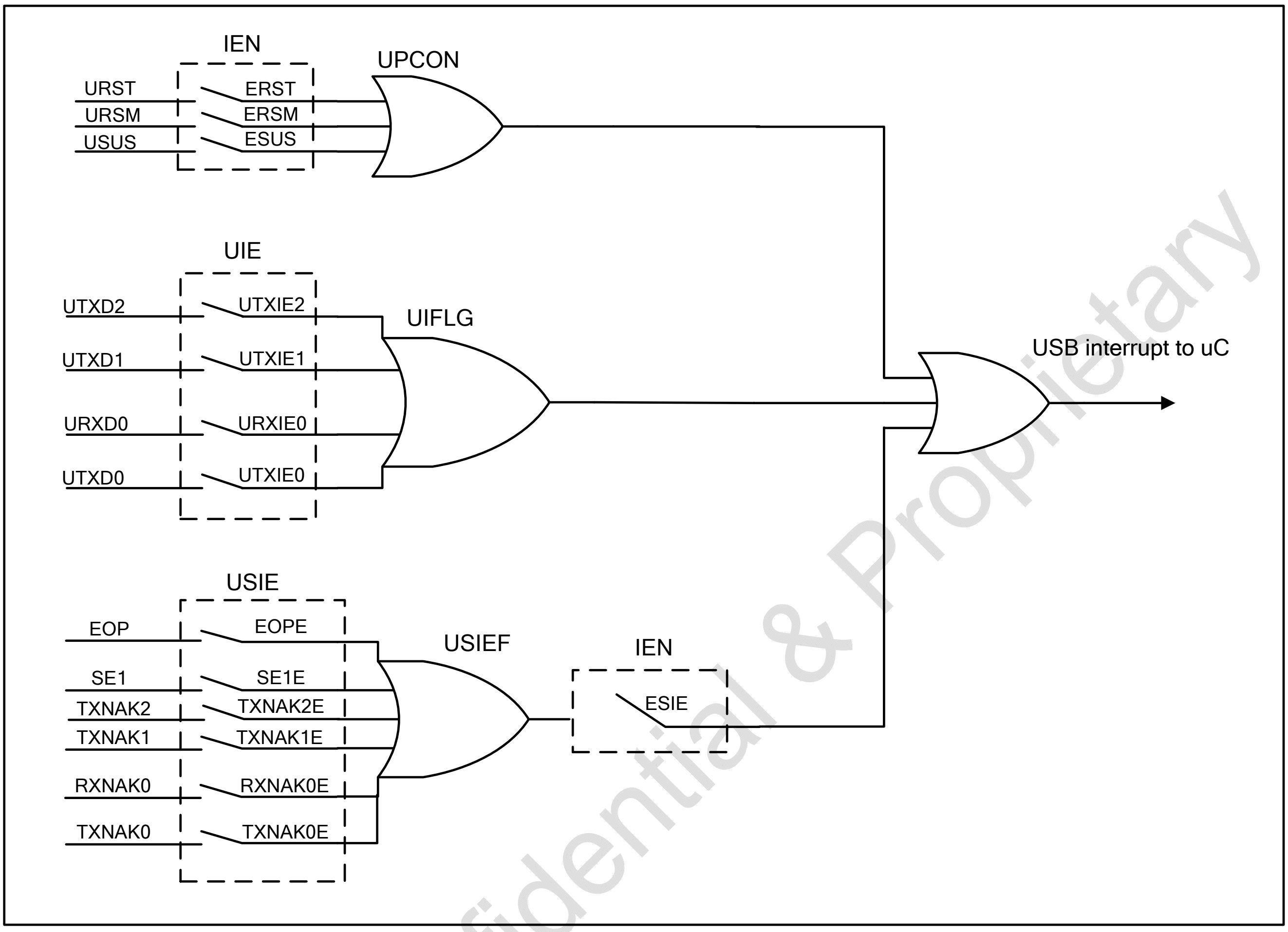


Figure 15-2 USB Interrupt Source

15.3.3 SFR Memory Mapping

Table 15-1 USB SFR Mapping Table

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
xx30H	--	EPINDEX	TXSTAT	TXDAT	TXCON	--	TXCNT	--	xx37H
xx28H	--	--	--	--	--	--	--	--	xx2FH
xx20H		EPCON	RXSTAT	RXDAT	RXCON	--	RXCNT	--	xx27H
xx18H	UIE	--	UIFLG	--	--	--	--	--	xx1FH
xx10H	IEN	--	UPCON	--	--	--	--	--	xx17H
xx08H	UADDR	--	--	--	--	--	--	--	xx0FH
xx00H	--	DCON	--	--	--	--	--	--	xx07H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

15.3.4 USB Reset event

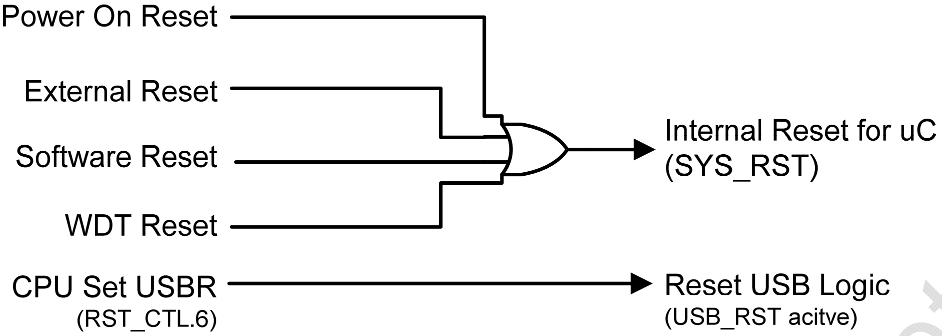


Figure 15-3 USB Macro Reset Source

15.3.5 SFR Description

Table 15-2 USB SFR Table

SYMBOL	DESCRIPTION	ADDR	BIT SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
DCON	Device Control Register	01H	0	0	0	PUREN	RPD	SETNO	STLDEN	EP2DIR	0000 0000B (R/W)
UADDR	USB Address Register	08H	0	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0	0000 0000B (R/W)
UPCON	USB Power Control Register	12H	--	FSE0	URWU	--	--	URST	URSM	USUS	x00x x000B (R/W)
IEN	Interrupt Enable Register	10H	--	--	--	--	ESIE	ERST	ERSM	ESUS	xxxx 0000B (R/W)
UIE	USB Interrupt Enable Register	18H	--	--	--	--	UTXIE2 URXIE2	UTXIE1	URXIE0	UTXIE0	xxxx 0000B (R/W)
UIFLG	USB Interrupt Flag Register	1AH	--	--	--	--	UTXD2 URXD2	UTXD1	URXD0	UTXD0	xxxx 0000B (R/W)
USIE	USB SIE event Enable Register	1BH	EOPE	SE1E	--	--	TXNAK2E	TXNAK1E	RXNAK0E	TXNAK0E	00xx 0000B (R/W)
USIEF	USB SIE Event Flag	1CH	EOP	SE1	--	--	TXNAK2	TXNAK1	RXNAK0	TXNAK0	00xx 0000B (R/W)
EPINDEX	Endpoint Index Register	31H	--	--	--	--	--	--	EPINX1	EPINX0	xxxx xx00B (R/W)

Note : Un-defined bit must be written “0” when uC program the SFR.

Table 15-3 USB SFR Table (Continue)

SYMBOL	DESCRIPTION	ADDR	BIT SYMBOL								RESET VALUE
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
EPINDEX=0, Endpoint 0 Input / Output Control SFR											
EPCON	Endpoint Control Register	21H	RXSTL	TXSTL	--	--	--	RXEPEN	--	TXEPEN	00xx 0101B (R/W)
RXSTAT	Endpoint Receive Status Register	22H	RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	--	--	--	0000 0xxxB (R/W)
RXDAT	FIFO Receive Data Register	23H	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0	xxxx xxxxB (R/W)
RXCON	FIFO Receive Control Register	24H	RXCLR	--	--	RXFFRC	--	--	--	--	0xx0 xxxxB (R/W)
RXCNT	FIFO Receive Byte Count Register	26H	--	--	--	--	RXBC3	RXBC2	RXBC1	RXBC0	xxxx 0000B (R/W)
TXSTAT	Endpoint Transmit Status Register	32H	TXSEQ	--	--	--	TXSOVW	--	TXERR	--	0xxx 0x0xB (R/W)
TXDAT	FIFO Transmit Data Register	33H	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0	xxxx xxxxB (R/W)
TXCON	FIFO Transmit Control Register	34H	TXCLR	--	--	TXFFRC	--	--	--	--	0xx0 xxxxB (R/W)
EPINDEX=1, Endpoint 1 Output Control SFR											
EPCON	Endpoint Control Register	21H	--	TXSTL	--	--	--	--	--	TXEPEN	x0xx xxx0B (R/W)
TXSTAT	Endpoint Transmit Status Register	32H	TXSEQ	--	--	--	TXSOVW	--	TXERR	--	0xxx 0x0xB (R/W)
TXDAT	FIFO Transmit Data Register	33H	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0	xxxx xxxxB (R/W)
TXCON	FIFO Transmit Control Register	34H	TXCLR	--	--	TXFFRC	--	--	--	--	0xx0 xxxxB (R/W)
EPINDEX=2, DCON.EP2DIR=0 : Endpoint 2 Output Control SFR											
EPCON	Endpoint Control Register	21H	--	TXSTL	--	--	--	--	--	TXEPEN	x0xx xxx0B (R/W)
TXSTAT	Endpoint Transmit Status Register	32H	TXSEQ	--	--	--	TXSOVW	--	TXERR	--	0xxx 0x0xB (R/W)
TXDAT	FIFO Transmit Data Register	33H	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0	xxxx xxxxB (R/W)
TXCON	FIFO Transmit Control Register	34H	TXCLR	--	--	TXFFRC	--	--	--	--	0xx0 xxxxB (R/W)
EPINDEX=2, DCON.EP2DIR=1 : Endpoint 2 Input Control SFR											
EPCON	Endpoint Control Register	21H	RXSTL	--	--	--	--	RXEPEN	--	--	0xxx x0xxB (R/W)
RXSTAT	Endpoint Receive Status Register	22H	RXSEQ	--	--	--	RXSOVW	--	--	--	0xxx 0xxxB (R/W)
RXDAT	FIFO Receive Data Register	23H	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0	xxxx xxxxB (R/W)
RXCON	FIFO Receive Control Register	24H	RXCLR	--	--	RXFFRC	--	--	--	--	0xx0 xxxxB (R/W)
RXCNT	FIFO Receive Byte Count Register	26H	--	--	--	--	RXBC3	RXBC2	RXBC1	RXBC0	xxxx 0000B (R/W)

Note: Un-defined bit must be written “0” when uC program the SFR.

DCON (Device Control Register, Address=01H, **SYS_RST**, Read/Write)

7	6	5	4	3	2	1	0
0	0	0	PUREN	RPD	SETNO	STLDEN	EP2DIR
w	w	w	R / W	R / W	R / W	R / W	R / W

Bit[7:5] : Reserved. Software must write “0” on these bits.

Bit4 : PUREN -- Enable USB DM 1.5K Pull-up resistor
0: Disable. (Default)
1: Enable.

Bit3 : RPD -- USB DP/DM 500K Pull-down resistor.
0: Disable. (Default)
1: Enable.

Bit2 : SETNO -- Set No-response in EP0 IN/OUT transaction.
1: Device will be just only response ACK packet with SETUP transaction but no response with EP0 IN/OUT transaction.
0: Device will sent ACK/NAK/STALL packet in IN/OUT transaction.
Note: This bit will be clear by HW when Device receive an SETUP token.

Bit1 : STLDEN -- STALL Done interrupt enable.
0: Disable IN/OUT STALL transaction flag setting.
1: IN/OUT STALL transaction will set TXD0/RXD0 in FIFLG.

Bit0 : EP2DIR -- USB Endpoint 2 Direction select.
0: EP2 will behave as an TX(IN) endpoint. Default is TX(IN). Only TX SFRs are valid.
1: EP2 will behave as an RX(OUT) endpoint. Only RX SFRs are valid.

UADDR (USB Address Register, Address=08H, **SYS_RST / USB_RST**, Read/Write)

7	6	5	4	3	2	1	0
0	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
w	R / W	R / W	R / W	R / W	R / W	R / W	R / W

Bit7: Reserved. Software must write “0” on these bits.

Bit[6:0] : UADD[6:0] -- USB Function Address.

UPCON (USB Power Control Register, Address=09H, **SYS_RST / USB_RST**, Read/Write)

7	6	5	4	3	2	1	0
--	FSE0	URWU	--	--	URST	URSM	USUS
	R / W	R / W			R / W	R / W	R / W

Bit7 : Reserved.

Bit6 : FSE0 -- Force SE0 (Only reset by SYS_RST)
0 : no effect. (Default)
1 : Force SE0 on USB bus

Bit5: URWU -- USB Remote Wake-Up Trigger.
0: End driving Remote Wake-Up signal on USB bus. (Default)
1: Start driving Remote Wake-Up signal on USB bus.

Bit[4:3] : Reserved.

Bit2: URST -- USB Reset Flag.
0: This bit is cleared when firmware writes '1' to it.
1: Set by hardware when the function detects the USB bus reset.

Bit1: URSM -- USB Resume Flag.
0: This bit is cleared when firmware writes '1' to it.
1: Set by hardware when the function detects the resume state on the USB bus from host.

Bit0: USUS -- USB Suspend Flag.
0: This bit is cleared when firmware writes '1' to it.
1: Set by hardware when the function detects the suspend state on the USB bus

IEN (Interrupt Enable Register, Address=10H, **SYS_RST**, Read/Write)

7	6	5	4	3	2	1	0
--	--	--	--	ESIE	ERST	ERSM	ESUS
				R / W	R / W	R / W	R / W

Bit[7:4] : Reserved.

Bit3 : ESIE -- Enable USIE event interrupt.
0: Disable (Default)
1: Enable

Bit2 : ERST -- Enable UPCON.URST interrupt.
0: Disable (Default)
1: Enable

Bit1 : ERSM -- Enable UPCON.URSM interrupt.
0: Disable (Default)
1: Enable

Bit0 : ESUS -- Enable UPCON.USUS interrupt.
0: Disable (Default)
1: Enable

UIE (USB Interrupt Enable Register, Address=18H, **SYS_RST**, Read/Write)

7	6	5	4	3	2	1	0
--	--	--	--	UTXIE2 URXIE2	UTXIE1	URXIE0	UTXIE0
				R / W	R / W	R / W	R / W

Bit[7:4] : Reserved.

Bit3: Select TX / RX by DCON.EP2DIR setting. (Default DCON.EP2DIR=0)
UTXIE2 -- Enable UIFLG.UTXD2 Interrupt.
URXIE2 -- Enable UIFLG.URXD2 Interrupt.
0: Disable. (Default)
1: Enable.

Bit2: UTXIE1 -- Enable UIFLG.UTXD1 Interrupt.
0: Disable. (Default)
1: Enable.

Bit1: URXIE0 -- Enable UIFLG.URXD0 Interrupt.
0: Disable. (Default)
1: Enable.

Bit0: UTXIE0-- Enable UIFLG.UTXD0 Interrupt.
0: Disable. (Default)
1: Enable.

UIFLG (USB Interrupt Flag Register, Address=1AH, **SYS_RST / USB_RST**, Read/Write)

7	6	5	4	3	2	1	0
--	--	--	--	UTXD2 URXD2	UTXD1	URXD0	UTXD0
				R / W	R / W	R / W	R / W

Bit[7:4] : Reserved.

Bit3 : Select TX / RX by DCON.EP2DIR setting. (Default DCON.EP2DIR=0)
UTXD2 -- Endpoint 2 Transmit done flag.
URXD2 -- Endpoint 2 Receive done flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 2 Transmit / Receive done flag.

Bit2 : UTXD1 -- Endpoint 1 Transmit done flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 1 Transmit done flag.

Bit1 : URXD0 -- Endpoint 0 Receive done flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 0 Receive done flag.

Bit0 : UTXD0 -- Endpoint 0 Transmit done flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 0 Transmit done flag.

USIE

(USB SIE Interrupt Enable Register, Address=1BH, **SYS_RST / USB_RST** = 00xx-0000, Read/Write)

7	6	5	4	3	2	1	0
EOPE	SE1E	--	--	TXNAK2E	TXNAK1E	RXNAK0E	TXNAK0E
R / W	R / W			R / W	R / W	R / W	R / W

Bit7: EOPE -- Enable USIEF.EOP Interrupt.
0: Disable. (Default)
1: Enable.

Bit6 : SE1E -- Enable USIEF.SE1 Interrupt.
0: Disable. (Default)
1: Enable.

Bit[5:4] : Reserved.

Bit3 : TXNAK2E -- Enable USIEF.TXNAK2 Interrupt.
0: Disable. (Default)
1: Enable.
Note : Endpoint 2 have TX NAK flag，not have RX NAK flag。

Bit2 : TXNAK1E -- Enable USIEF.TXNAK1 Interrupt.
0: Disable. (Default)
1: Enable.

Bit1 : RXNAK0E -- Enable USIEF.RXNAK0 Interrupt.
0: Disable. (Default)
1: Enable.

Bit0: TXNAK0E -- Enable USIEF.TXNAK0 Interrupt.
0: Disable. (Default)
1: Enable.

USIEF

(USB SIE Interrupt Flag Register, Address=1CH, **SYS_RST / USB_RST** = 00xx-0000, Read/Write)

7	6	5	4	3	2	1	0
EOP	SE1	--	--	TXNAK2	TXNAK1	RXNAK0	TXNAK0
R / W	R / W			R / W	R / W	R / W	R / W

Bit7 : EOP -- EOP event flag.
0: This bit is cleared when firmware writes '1' to it.
1: EOP event detected flag.

Bit6 : SE1 -- SE1 event flag. Hardware will set this flag to “1” when DP/DM input voltage is both higher than V_{IH} and remains the same more than 1 USB bit time.
0: This bit is cleared when firmware writes '1' to it.
1: SE1 event detected flag.

Bit[5:4] : Reserved.

Bit3 : TXNAK2 -- Endpoint 2 TX NAK event flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 2 TX NAK flag.
Note : Endpoint 2 only have TX NAK flag, not have RX NAK flag。

Bit2 : TXNAK1 -- Endpoint 1 TX NAK event flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 1 TX NAK flag.

Bit1 : RXNAK0 -- Endpoint 0 RX NAK event flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 0 RX NAK flag.

Bit0 : TXNAK0 -- Endpoint 0 TX NAK event flag.
0: This bit is cleared when firmware writes '1' to it.
1: Endpoint 0 TX NAK flag.

EPINDEX

(Endpoint Index Register, Address=31H, **SYS_RST / USB_RST**, Read/Write)

7	6	5	4	3	2	1	0
--	--	--	--	--	--	EPINX1	EPINX0
						R / W	R / W

Bit[7:2] : Reserved.

Bit[1:0] : EPINX[1:0] -- Endpoint Index Bits [2:0]
  2'b00: Function Endpoint 0. (Default)
  2'b01: Function Endpoint 1.
  2'b10: Function Endpoint 2.
  2'b11: Reserved.

EPCON (Endpoint Control Register, Endpoint-Indexed, Address=21H, **SYS_RST / USB_RST**, Read/Write)

7	6	5	4	3	2	1	0
RXSTL	TXSTL	--	--	--	RXEPEN	--	TXEPEN
R / W	R / W				R / W		R / W

Endpoint 0 (EPINDEX=0)

Bit7 : RXSTL -- Receive Endpoint Stall.
  0: Disable. (Default)
  1: Enable.
  Note : Clear this bit only when the host has intervened through commands sent down endpoint 0. When this bit is set and RXSETUP is clear, the receive endpoint will respond with a STALL handshake to a valid OUT token. When this bit is set and RXSETUP is set, the receive endpoint will NAK. This bit does not affect the reception of SETUP tokens by a control endpoint.

Bit6 : TXSTL -- Transmit Endpoint Stall.
  0: Disable. (Default)
  1: Enable.
  Note : Clear this bit only when the host has intervened through commands sent down endpoint 0. When this bit is set and RXSETUP is clear, the transmit endpoint will respond with a STALL handshake to a valid IN token. When this bit is set and RXSETUP is set, the transmit endpoint will NAK.

Bit[5:3] : Reserved.

Bit2 : RXEPEN -- Receive Endpoint Enable.
  0: Disable.
  1: Enable. (Default)

Bit1 : Reserved.

Bit0 : TXEPEN -- Transmit Endpoint Enable.
  0: Disable.
  1: Enable. (Default)

Endpoint 1 (EPINDEX=1)

Bit7 : Reserved.

Bit6 : TXSTL -- Transmit Endpoint Stall.
  0: Disable. (Default)
  1: Enable.

Bit[5:1] : Reserved.

Bit0 : TXEPEN -- Transmit Endpoint Enable.
  0: Disable. (Default)
  1: Enable.

Endpoint 2 (EPINDEX=2, DCON.EP2DIR=0)

Bit7 : Reserved.

Bit6 : TXSTL -- Transmit Endpoint Stall.
0: Disable. (Default)
1: Enable.

Bit[5:1] : Reserved.

Bit0 : TXEPEN -- Transmit Endpoint Enable.
0: Disable. (Default)
1: Enable.

Endpoint 2 (EPINDEX=2, DCON.EP2DIR=1)

Bit7 : RXSTL -- Receive Endpoint Stall.
0: Disable. (Default)
1: Enable.

Bit[6:3] : Reserved.

Bit2 : RXEPEN -- Receive Endpoint Enable.
0: Disable. (Default)
1: Enable.

Bit[1:0] : Reserved.

RXSTAT

(Endpoint Receive Status Register, Endpoint-Indexed, Address=22H, SYS_RST / USB_RST, Read/Write)

7	6	5	4	3	2	1	0
RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	--	--	--
R / W	R / W	R / W	R / W	R / W			

Endpoint 0 (EPINDEX=0)

Bit7 : 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.

Bit6 : 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 tra or the firmware is ready to handle the data/status stage of control transfer. Firmware write “1” to clear this bit.

Bit5 : STOVW -- Start Overwrite Flag (read-only).
Set by hardware upon receipt of a SETUP token for the control endpoint to indicate that the receive FIFO is being overwritten with new SETUP data. This bit is used only for control endpoints.

Bit4 : EDOVW -- End Overwrite Flag.
This flag is set by hardware during the handshake phase of a SETUP transaction. This bit is cleared by firmware write “1” to read the FIFO data. This bit is only used for control endpoints.

Bit3 : RXSOVW -- Receive Data Sequence Overwrite Bit.
Write '1' to this bit to allow the value of the RXSEQ bit to be overwritten.
Write '0' to this bit has no effect on RXSEQ. This bit always returns '0' when read.

Bit[2:0] : Reserved.

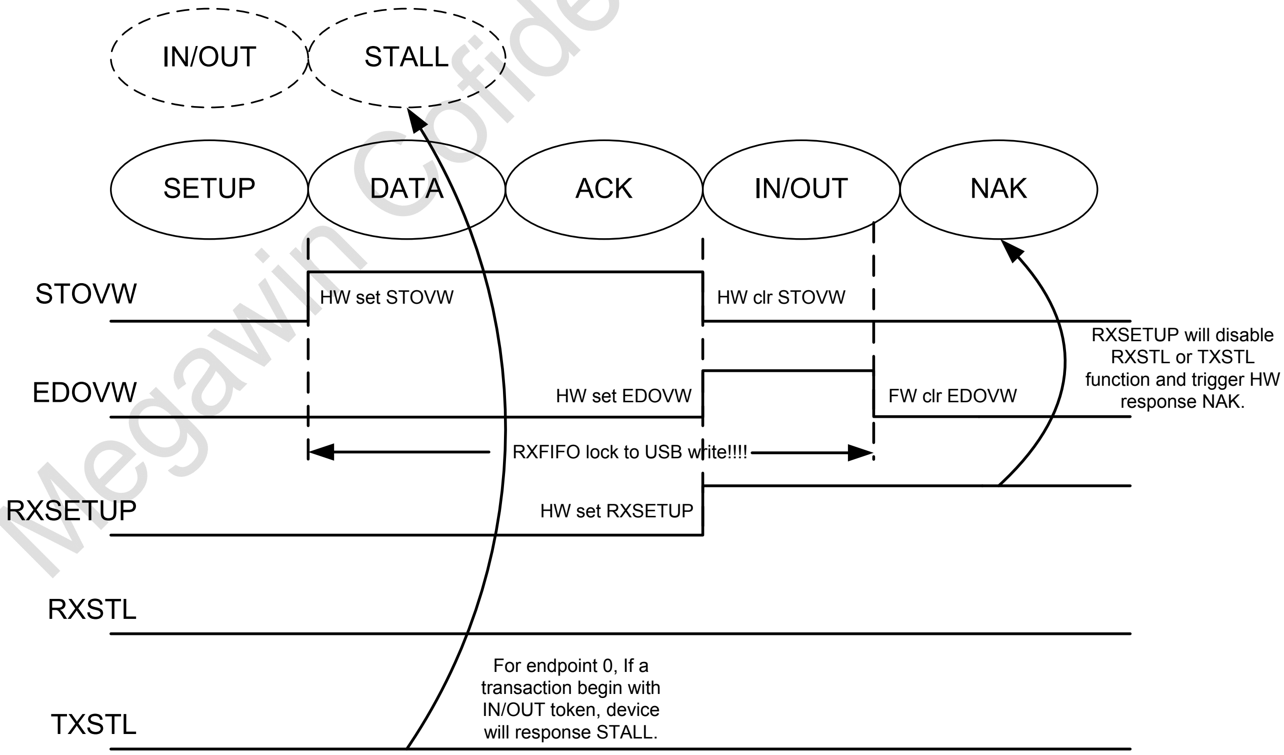

IN/OUT
STALL
SETUP
DATA
ACK
IN/OUT
NAK
STOVW
HW set STOVW
HW clr STOVW
EDOVW
HW set EDOVW
FW clr EDOVW
RXSETUP will disable RXSTL or TXSTL function and trigger HW response NAK.
RXFIFO lock to USB write!!!!
RXSETUP
HW set RXSETUP
RXSTL
TXSTL
For endpoint 0, If a transaction begin with IN/OUT token, device will response STALL.

Figure 15-4 USB Control Transfer Control Description

Endpoint 2 (EPINDEX=2, DCON.EP2DIR=1)

Bit7: 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.

Bit[6:4] : Reserved.

Bit3 : RXSOVW -- Receive Data Sequence Overwrite Bit.
Write '1' to this bit to allow the value of the RXSEQ bit to be overwritten.
Write '0' to this bit has no effect on RXSEQ. This bit always returns '0' when read.

Bit[2:0] : Reserved.

RXDAT

(Receive FIFO Data Register, Endpoint-Indexed, Address=23H, **SYS_RST / USB_RST**, Read-only)

7	6	5	4	3	2	1	0
RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
R	R	R	R	R	R	R	R

Bit[7:0] : RXD[7:0] -- Receive FIFO Data.
Receive FIFO data specified by EPINDEX is stored and read from this register.

Note : RXDAT only services the receive endpoints.

RXCON

(Receive FIFO Control Register, Endpoint-Indexed, Address=24H, **SYS_RST / USB_RST**, Write-only)

7	6	5	4	3	2	1	0
RXCLR	--	--	RXFFRC	--	--	--	--
R / W			R / W				

Bit7 : 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.
Note: The RXSEQ bit in the RXSTAT register are not affected by this operation.

Bit[6:5] : Reserved.

Bit4 : 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.
Note: For Endpoint 0, RXFFRC only works if STOVW and EDOVW are cleared.

Bit[3:0] : Reserved.

Note: RXCON only services the receive endpoints.

RXCNT

(Receive FIFO Byte Count Register, Endpoint-Indexed, Address=26H, **SYS_RST / USB_RST**, Read-only)

7	6	5	4	3	2	1	0
--	--	--	--	RXBC3	RXBC2	RXBC1	RXBC0
				R	R	R	R

Bit[7:4] : Reserved.

Bit[3:0] : RXBC[3:0] -- Receive Byte Count.
Store the byte count for the data packet received in the receive FIFO specified by EPINDEX.

Note : RXCNT only services the receive endpoints.

TXSTAT

(Endpoint Transmit Status Register, Endpoint-Indexed, Address=32H, **SYS_RST / USB_RST**, Read/Write)

7	6	5	4	3	2	1	0
TXSEQ	--	--	--	TXSOVW	--	TXERR	--
R / W				R / W		R / W	

Bit7 : 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 TXOVW bit is set when written along with the new TXSEQ value.

Bit[6:4] : Reserved.

Bit3 : TXSOVW -- Transmit Data Sequence Overwrite Bit.
Write '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.
Note: The SIE will handle all sequence bit tracking. This bit should be used only when initializing a new configuration or interface.

Bit2 : Reserved.

Bit1: TXERR -- Transmit Data timeout error flag.
Only for EP0, EP1 and EP2 TX function. Write “1” to clear it.

Bit0 : Reserved.

Note: TXSTAT only services the transmit endpoints.

TXDAT

(Transmit FIFO Data Register, Endpoint-Indexed, Address=33H, **SYS_RST / USB_RST**, Write-only)

7	6	5	4	3	2	1	0
TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
W	W	W	W	W	W	W	W

Bit[7:0] : TXD[7:0] -- Transmit FIFO Data.
Data to be transmitted in the FIFO specified by EPINDEX is written to this register.

Note : TXDAT only services the transmit endpoints.

TXCON

(Transmit FIFO Control Register, Endpoint-Indexed, Address=34H, **SYS_RST / USB_RST**, Write-only)

7	6	5	4	3	2	1	0
TXCLR	--	--	TXFFRC	--	--	--	--
R / W			R / W				

Bit7 : 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.

Note: TXCLR would clear TXFIFO. The TXSEQ bit in the TXSTAT register are not affected by this operation.

Bit[6:5] : Reserved.

Bit4 : TXFFRC -- TX FIFO Ready Complete.

Bit[3:0] : Reserved.

Note: TXCON only services the transmit endpoints.

16 In Application Programming (IAP)

MG64F237 has a 8K bytes MTP memory. Total 8K bytes support IAP function. The IAP-memory block can be accessed by CPU to store the user data and application program.

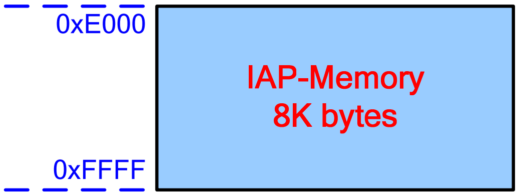


Figure 16-1 IAP Block Diagram

16.1 IAP Register

IAP_PR (IAP Write Protect Register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R	W	Default
00E0H	IAP_PR	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0		√	xxxx xxxxB

Bit[7:0] : PR[7:0] -- Write Protect Pattern.

IAP-memory block would be written by firmware, when IAP_WP is written “46H” then “B9H”.
The IAP_WP will be automatically cleared by next uC write action or flash write time-out.

Note : Clear watch timer before the IAP function is used.
Note : When VDD < VLVD1(2.4V), IAP function would be disabled.

Example:

```
SEI
LDA     #5AH
STA     PWPR
LDA     #80H
STA     WDT_ST
LDA     #46H
STA     IAP_PR
LDA     #B9H
STA     IAP_PR
LDA     #40h            ;The data will be written into flash.
STA     E000h           ;IAP_AREA (E000h ~ FFFFh)
CLI
```

17 Hardware Option

There are two function control by hardware option.

LOCK
Enabled: Code dump on Writer is always 0xFF, page-erase and program is also disabled.
Disabled: Code dump on Writer is transparent. (Default)

ENRCO: Internal OSC Function Enable/Disable
Enable: Enable internal OSC to be a clock source.
Disable: External OSC to be a clock source. (Default)

18 Application Circuit

18.1 USB Keyboard circuit

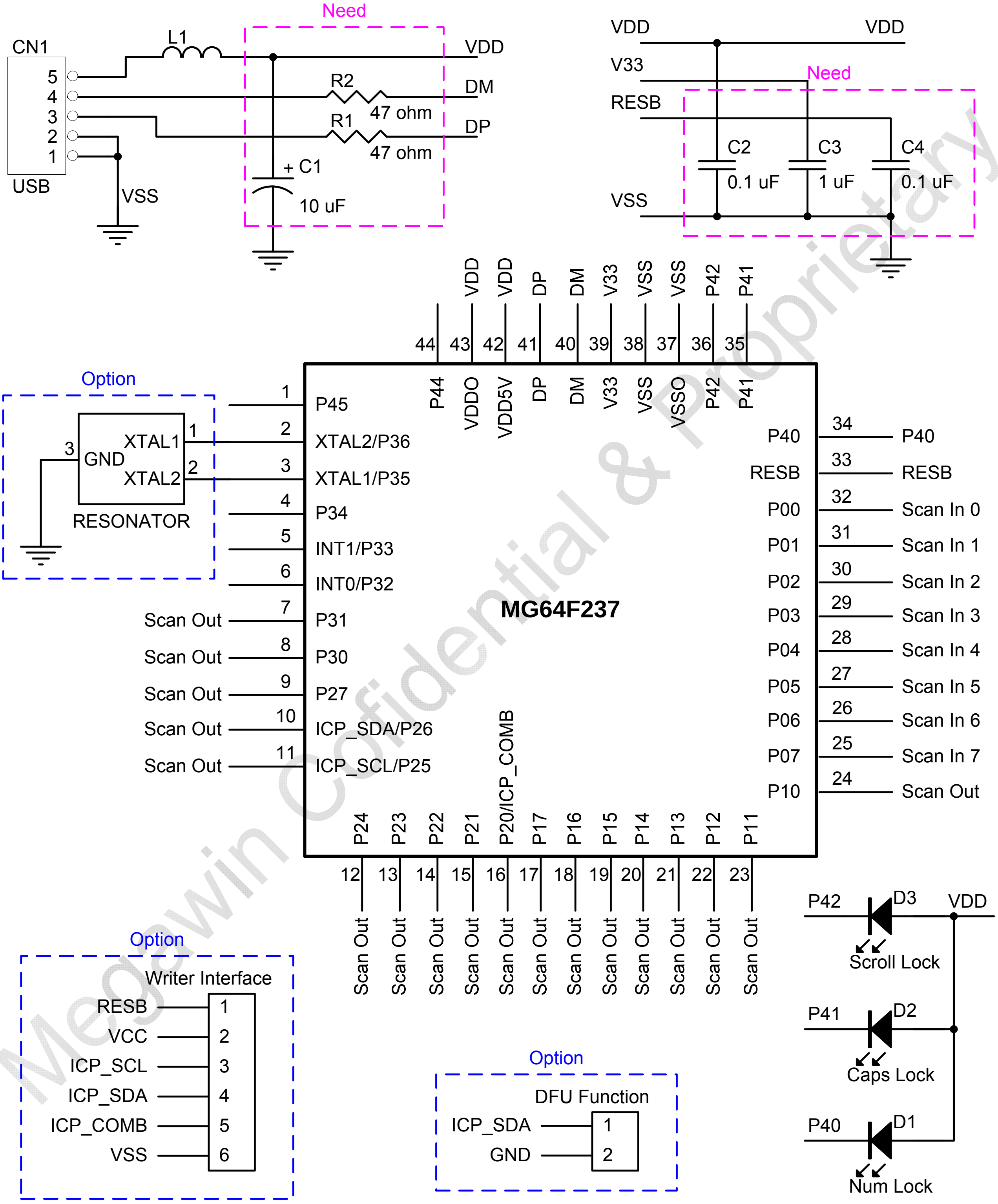


Figure 18-1 Application Circuit - Keyboard

18.2 USB Multi Media Dongle

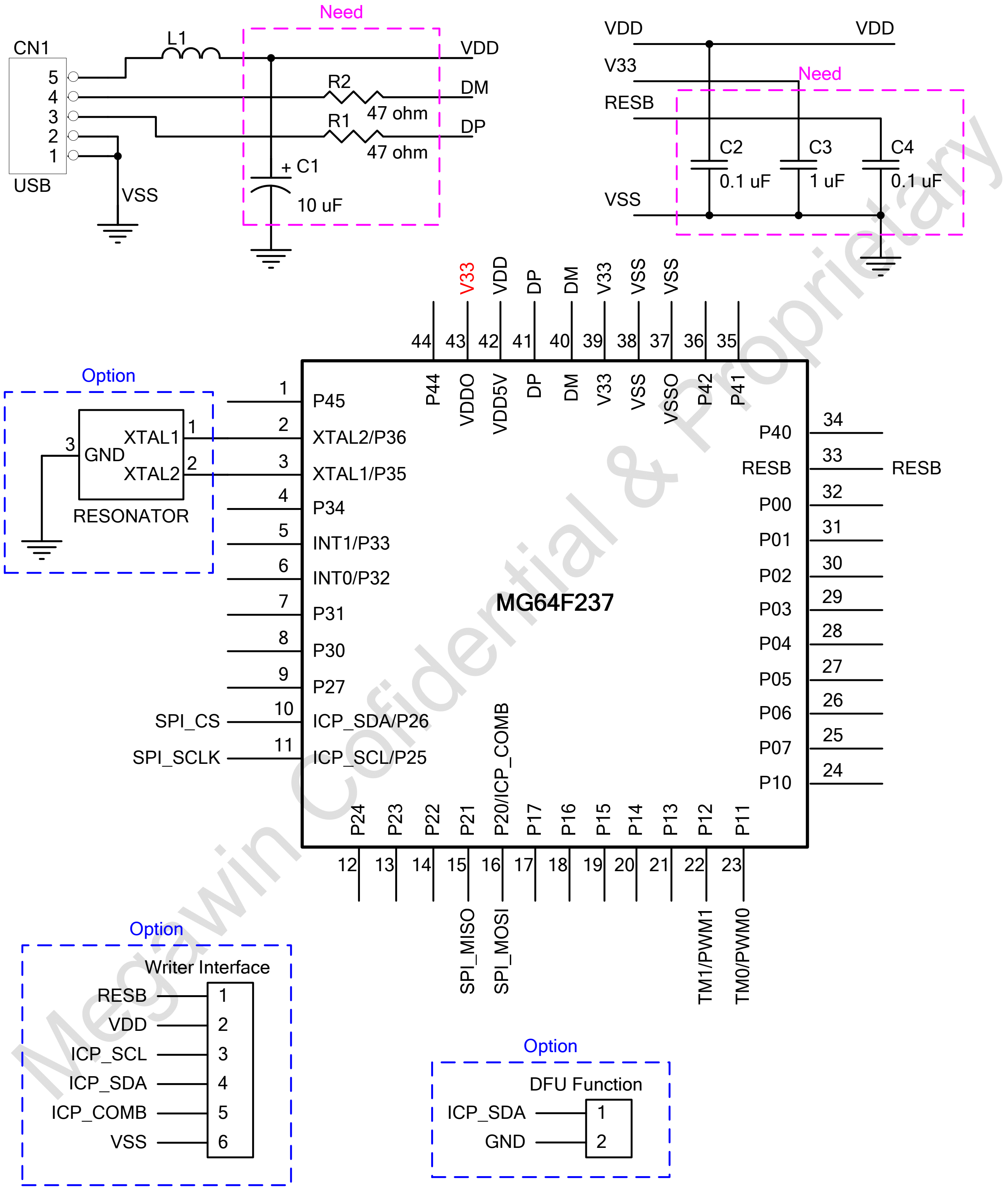
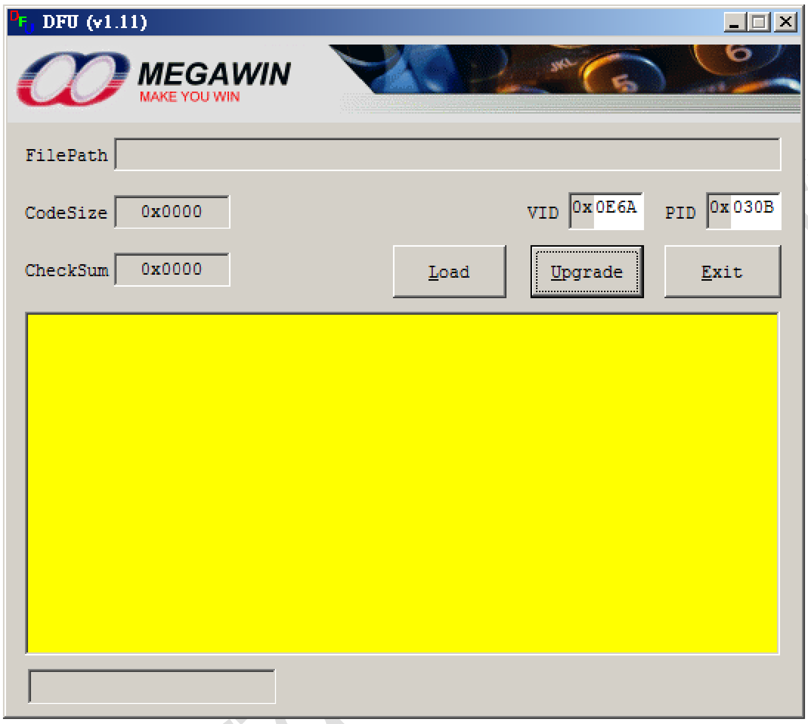


Figure 18-2 Application Circuit - Dongle

19 DFU Function Description

DFU is a USB on-line update firmware function. User can use this function in development or AP on-line check/update device firmware version flow. MEGAWIN provide AP source code and firmware library. User must insert USB HID DFU command, 2 vector command and DFU library code (1.5K Bytes).

19.1 DFU AP Description



FilePath
CodeSize 0x0000
CheckSum 0x0000
VID 0x0E6A
PID 0x030B

Figure 19-1 MEGAWIN DFU Application Program

VID 0x0E6A PID 0x030B : User's USB device VID(Vender ID) and PID(Product ID).

Load : Load user code (Code size must be 8KBytes)

Upgrade : Start to update firmware.

19.2 DFU Firmware Library

- USB HID DFU command
  - DFU Variable Define

```
m_EP0_Stage        .DS     1       ; Endpoint 0 USB stage
StatusStage        .EQU    0       ;STATUS STAGE
DataStage_R        .EQU    1       ;DATA STAGE ( IN, TXD0 )
DataStage_W        .EQU    2       ;DATA STAGE ( OUT, RXD0 )
SetAddress         .EQU    3
DFU_STAGE          .EQU    33h
DFU_RESET          .EQU    44h
 ;--------------------------------------------
m_EP0_RXCNT        .DS     1
m_EP0_RXTX         .DS     8       ;for save SETUP 8 bytes command
```

- DFU command : Endpoint 0 TX done

```
Endpoint_0_TXD0:
Endpoint_0_IN_token:
        LDA     m_EP0_Stage
        CMP     #DFU_RESET
        BEQ     DFU_CMD_IN_OK_Check_CMD
        ...
 ;---------------------------------------------
 ;(StatusStage)
        JMP     EP0_set_StatusStage_STALL
        ...
 ;---------------------------------------------
 ;(DFU_STAGE) : SETUP + OUT + IN
 DFU_CMD_IN_OK_Check_CMD:
        USB_w_SFR_Data_s #DCON,#00h     ;Disable PUREN
        mov_    01FFh,#5AH              ;DFU Function Flag
        mov_    01FEh,#A5H              ;DFU Function Flag
        mov_    PWPR,#5AH               ;Protect write unlock
        mov_    RST_CTL,#80H            ;Software Reset
        ...
```

- DFU command : Endpoint 0 RX done

```
Endpoint_0_OUT_token:
        LDA     m_EP0_Stage
        CMP     #DFU_STAGE
        BEQ     DFU_Read_CMD_to_FIFO
        ...
 ;---------------------------------------------
 ; (StatusStage)
 ;SETUP + OUT + OUT + .... + IN
        JMP     EP0_set_StatusStage_STALL
        ...
 ;---------------------------------------------
 ;(DFU_STAGE) : SETUP + OUT + IN
 DFU_Read_CMD_to_FIFO:
        JSR     USB_R_RXDAT_to_RAM      ;m_EP0_RXCNT / m_EP0_RXTX
        LDA     m_EP0_RXTX+0
        STA     m_EP0_Stage,
        JSR     USB_W_RXFFRC_RTS        ;set RXFFRC to Clear FIFO
        JMP     USB_W_TXFFRC_RTS        ;Return zero length ACK
```

- DFU command : USB HID Set Report

```
USB_SET_REPORT:
        LDA     m_EP0_RXTX+3            ;Report Type (1=Input, 3=Feature)
        CMP     #3
        BEQ     USB_SET_REPORT_Feature  ;Set_Report_Feature
        JMP     EP0_set_StatusStage_STALL
        ...
 USB_SET_REPORT_Feature:                ;Set DUF DataStage_W Stage
        LDA     #DFU_STAGE
        STA     m_EP0_Stage
        RTS
```

- USB HID Descriptor

```
           Interface_0_HID_Report:

... (User applictaion)...

           ;=== Vendor Defined : DFU Function Command ===
           .DB      09h,00h                ;  Usage Page (Vendor Defined)
           .DB      95h,08h                ;   Report Count (8 Bytes)
           .DB      75h,08h                ;   Report Size (8 Bits)
           .DB      B1h,00h                ;   Feature (Data, Array, Absolute)

           .DB      C0h               ;End Collection
```

- 2 vector and DFU main library

```
           ;=== 2 Vector for Reset and Interrupt ===
           .ORG F9FAh
           JMP      reset             ;Jump to user main program start address
           JMP      irq_isr           ;Jump to user interrupt vector start address


           ;=== DFU Function Library ===
           .ORG FA00h
           BINCLUDE       MG64F237_DFU_V0107.lib
```

19.3 Manual force device into DFU mode

Operation flow:

1. Device connect to USB host.
2. Short P26 to VSS(Ground)
3. Short external reset pad to VSS(Ground) more than 200us, release reset pad to start free run.
4. Release P26.

20 Instruction Set

20.1 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(3)	3
4	Absolute Indexed with Y a,y	4	3
5	Absolute Indirect (a)	4-5(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	2-3	1
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 2-3 cycles

20.2 Instruction Set Table

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

20.3 Instruction Set Summary

MSD																	MSD
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		ORA (zp,x) 6,2				ORA zp 3,2	ASL zp 5,2	RMB0 zp 5,2	PHP s 3,1	ORA # 2,2	ASL A 2,1			ORA a 4,3	ASL a 6,3	BBR0 r 4,3	0
1	BPL r 2,2	ORA (zp),y 5,2	ORA (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			ORA a,x 4,3	ASL a,x 7,3	BBR1 r 4,3	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 3,1	AND # 2,2	ROL A 2,1		BIT a 4,3	AND a 4,3	ROL a 6,3	BBR2 r 4,3	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 7,3	BBR3 r 4,3	3
4	RTI s 5,1	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 4,3	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 7,3	BBR5 r 4,3	5
6	RTS s 5,1	ADC (zp,x) 6,2			STZ zp 3,2	ADC zp 3,2	ROR zp 5,2	RMB6 zp 5,2	PLA s 3,1	ADC # 2,2	ROR A 2,1		JMP (a) 5,3	ADC a 4,3	ROR a 6,3	BBR6 r 4,3	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 3,1		JMP (a,x) 6,3	ADC a,x 4,3	ROR a,x 7,3	BBR7 r 4,3	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 4,3	8
9	BCC r 2,2	STA (zp),y 5,2	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 4,3	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 4,3	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 4,3	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 4,3	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 7,3	BBS5 r 4,3	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 4,3	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 3,1			SBC a,x 4,3	INC a,x 7,3	BBS7 r 4,3	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

A: Accumulator
X: Index Register X
Y: Index Register Y

zp: Address8 or zero page
a: Adress16
s: Stack

r: Relative
i : Implied

20.4 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 →16:	Address Extend to Absolute Address (Get two addr8 contents continuously)
label:	Address Variable
~:	1's compliment
∩:	AND
∪:	OR
⊕ :	Exclusive OR
←:	Transfer direction, result

20.5 Arithmetic Operations

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
ADC	#data	(ACC) ← (ACC) + #data + (C)	C, Z, V, S	2	2
	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
	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	#data	(ACC) ← (ACC) – #data – (~C)	C, Z, V, S	2	2
	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
	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
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	7
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	7
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
	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
	addr16	(ACC) – (addr16)	C, Z, S	3	4
	addr16, X	(ACC) – [addr16 + (X)]	C, Z, S	3	4
	addr16, Y	(ACC) – [addr16 + (Y)]	C, Z, S	3	4

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

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
CPX	#data	(X) – #data	C, Z, S	2	2
	addr8	(X) – (addr8)	C, Z, S	2	3
	addr16	(X) – (addr16)	C, Z, S	3	4
CPY	#data	(Y) – #data	C, Z, S	2	2
	addr8	(Y) – (addr8)	C, Z, S	2	3
	addr16	(Y) – (addr16)	C, Z, S	3	4

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

20.6 Logic Operations

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
AND	#data	(ACC) ← (ACC) ∩ #data	Z, S	2	2
	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
	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
ORA	#data	(ACC) ← (ACC) ∪ #data	Z, S	2	2
	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
	addr16	(ACC) ← (ACC) ∪ (addr16)	Z, S	3	4
	addr16, X	(ACC) ← (ACC) ∪ [addr16 + (X)]	Z, S	3	4
	addr16, Y	(ACC) ← (ACC) ∪ [addr16 + (Y)]	Z, S	3	4
EOR	#data	(ACC) ← (ACC) ⊕ #data	Z, S	2	2
	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
	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
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	7

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

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
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) ← (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	7
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	7
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	7
BIT	#data	(ACC) ∩ #data	Z	2	2
	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

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

20.7 Data Transfer

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
LDA	#data	(ACC) ← #data	Z, S	2	2
	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	5
	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
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	5
	addr16	(addr16) ← (ACC)	-	3	4
	addr16, X	[addr16 + (X)] ← (ACC)	-	3	4
	addr16, Y	[addr16 + (Y)] ← (ACC)	-	3	4
STX	addr8	(addr8) ← (X)	-	2	3
	addr8, Y	[addr8 + (Y)] ← (X)	-	2	4
	addr16	(addr16) ← (X)	-	3	4
STY	addr8	(addr8) ← (Y)	-	2	3
	addr8, X	[addr8 + (X)] ← (Y)	-	2	4
	addr16	(addr16) ← (Y)	-	3	4
STZ	addr8	(addr8) ← 00H	-	2	3
	addr8, X	[addr8 + (X)] ← 00H	-	2	4
	addr16	(addr16) ← 00H	-	3	4
	addr16, X	[addr16 + (X)] ← 00H	-	3	5
TAX		(X) ← (ACC)	Z, S	1	2
TXA		(ACC) ← (X)	Z, S	1	2
TAY		(Y) ← (ACC)	Z, S	1	2
TYA		(ACC) ← (Y)	Z, S	1	2
TSX		(X) ← (SP)	Z, S	1	2
TXS		(SP) ← (X)	-	1	2

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

Mnemonic	Operand(s)	Operation description	Flag	Byte	Cycle
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	3
PLP		(P) ← [(SP+1)], (SP) ← (SP) + 1	C, Z, I, D, V, S	1	3
PLX		(X) ← [(SP+1)], (SP) ← (SP) + 1	Z, S	1	3
PLY		(Y) ← [(SP+1)], (SP) ← (SP) + 1	Z, S	1	3

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

20.8 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.

20.9 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	5
	(addr16, X)	(PC) ← {[label + (X) →16]}	-	3	6
BRA	addr8	(PC) ← (PC)+addr8	-	2	3(1)
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	4(2)(3)
…					
BBR7	addr8	(PC) ← (PC)+addr8 if ACC.7 == 0 (+/- relative)	-	3	4(2)(3)
BBS0	addr8	(PC) ← (PC)+addr8 if ACC.0 == 1 (+/- relative)	-	3	4(2)(3)
…					
BBS7	addr8	(PC) ← (PC)+addr8 if ACC.7 == 1 (+/- relative)	-	3	4(2)(3)
JSR	label	stack ← (PC), (PC) ← label	-	3	6
RTS		(PC) ← pop stack	-	1	5
RTI		(PC) ← pop stack, restore status register P	C, Z, I, D, V, S	1	5
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: 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.

21 Electrical Characteristics

21.1 DC Characteristics

VSS = 0V, TA = 25 ℃, VDD5V = 5.0V , VDDO = 5.0V, F_{OSC} = 6MHz and execute NOP for each machine cycle, unless otherwise specified

Table 21-1 DC Characteristic (VDDO=5V)

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{IH1}	Input High voltage (All I/O ports except P35 & P36)		2.0			V
V_{IH2}	Input High voltage (P35 & P36)		3.2			V
V_{IH3}	Input High voltage (PS2_Data & PS2_CLK)		1.9			V
V_{IH4}	Input High voltage (RESET)		3.9			V
V_{IL1}	Input Low voltage (All I/O ports except P35 & P36)				0.8	V
V_{IL2}	Input Low voltage (P35 & P36)				1.4	V
V_{IL3}	Input Low voltage (PS2_Data & PS2_CLK)				0.6	V
V_{IL4}	Input Low voltage (RESET)				1.1	V
I_{OL1}	Output Low current (All I/O Ports except P35 & P36)	V_{PIN} = 0.4V		4		mA
I_{OL2}	Output Low current (P35 & P36)	V_{PIN} = 0.4V V_{PIN} = 1.2V		20 50		mA
I_{OL3}	Output low current (PS2_Data & PS2_CLK))	V_{PIN} = 0.4V		30		mA
I_{OP}	Operating current			6		mA
I_{IDLE}	Idle mode current			3		mA
I_{PD}	Power down current			150		uA
R_{IO1}	Internal pull-up resistance (P0 / P1 / P2 / P3)			50		KΩ
R_{IO2}	Internal pull-up resistance (P0 / P1 / P2)			3		MΩ
R_{RST}	Internal reset pull-down resistance			50		KΩ
R_{PS2}	PS2_Data & PS2_CLK pull-up resistance in PS2 mode			7		KΩ
R_{DM}	DM pull-up resistance in USB mode			1.1		KΩ
R_{PD}	DP/DM pull-down resistance in USB mode			500		KΩ
V_{33}	3.3V regulator output voltage			3.3		V
I_{33}	3.3V regulator output current				50	mA
V_{POR}	Low-voltage reset			2.7		V
V_{LVDF}	Low-voltage flag			3.6		V
IHRCO	Built-in osc. frequency		5.91	6	6.09	MHz

VSS = 0V, TA = 25 ℃, VDD5V = 5.0V , VDDO = 3.0V, F_{OSC} = 6MHz and execute NOP for each machine cycle, unless otherwise specified

Table 21-2 DC Characteristic (VDDO=3V)

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{IH1}	Input High voltage (All I/O ports except P35 & P36)		1.5			V
V_{IH2}	Input High voltage (P35 & P36)		2.1			V
V_{IH4}	Input High voltage (RESET)		2.4			V
V_{IL1}	Input Low voltage (All I/O ports except P35 & P36)				0.5	V
V_{IL2}	Input Low voltage (P35 & P36)				0.6	V
V_{IL4}	Input Low voltage (RESET)				0.6	V
I_{OL1}	Output Low current (All I/O Ports except P35 & P36)	V_{PIN} = 0.4V		3		mA
I_{OL2}	Output Low current (P35 & P36)	V_{PIN} = 0.4V V_{PIN} = 1.2V		10 30		mA
R_{IO1}	Internal pull-up resistance (P0 / P1 / P2 / P3)			130		KΩ
R_{IO2}	Big Internal pull-up resistance for power down wakeup (only P0 / P1 / P2)			10		MΩ
R_{RST}	Internal reset pull-down resistance			130		KΩ

21.2 USB Transceiver Electrical Characteristics

VSS = 0V, TA = 25 ℃, VDD5V = 5.0V and execute NOP for each machine cycle, unless otherwise specified

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
Transmitter						
V_{OH}	Output High Voltage		2.8			V
V_{OL}	Output Low Voltage				0.8	V
I_{OL}	DP/DM output Low Current	V_{PIN} = 0.4V		30		mA
V_{CRS}	Output Cross Over point		1.3		2.0	V
Z_{DRVH}	Output Impedance on Driving High		28		44	Ω
Z_{DRVL}	Output Impedance on Driving Low		28		44	Ω
T_R	Output Rise Time		75		300	ns
T_F	Output Fall Time		75		300	ns
Receiver						
V_{DI}	Differential Input Sensitivity	| DP – DM |	0.2			V
V_{CM}	Differential Input Common Mode Range		0.8		2.5	V
I_L	Input Leakage current	Pull-up Disabled		<1.0		uA

22 Revision History

Revision	Page	Descriptions	Date
V0.01		Preliminary Version.	2013/06/25
V1.00		Initial Version	2013/09/06