DATA SHEET

M151A

MTP-BASED 8-BIT MICROCONTROLLER

Tel:886-3-5510850 Fax:886-3-5510860

TABLE OF CONTENTS

Revision History:

Date	Revision#	Description	Page
2012/03/14	100	Original	
2012/07/11	101	Revise ● BR16MHz：High accuracy 16MHz built-in OSC (± 3% @ 3V 25℃, after calibration) ● Operation temperature: -25℃~75℃ ● Chapter 1: General Description	 4,13 36 4

M151A

MTP-Based 8-bit CMOS Microcontroller

1. General Description

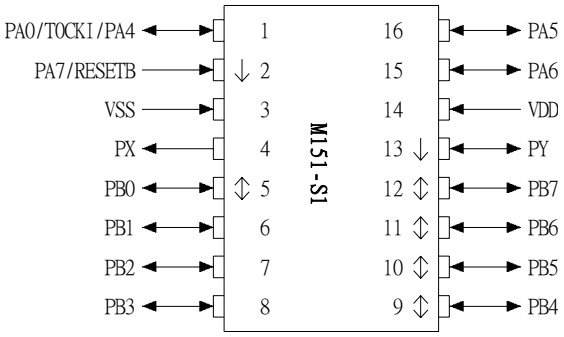
M151A is an 8-bit microprocessor with low-power and high-speed CMOS technology. It is equipped with a MTP memory and a data memory. It is advantageous in low-power application, like remote controller and battery-powered system.

2. Features

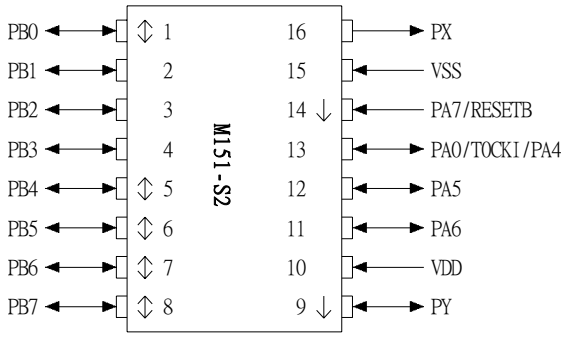
1. **Operate voltage**
   - 1.8V ~ 3.6V @ 4MHz
   - 2.2V ~ 3.6V @ 16MHz
2. **Built-in Main-clock oscillator**
   - BR4MHz：High accuracy 4MHz built-in OSC (± 1% @ 3V 25℃, after calibration)
   - BR16MHz：High accuracy 16MHz built-in OSC (± 3% @ 3V 25℃, after calibration)
3. **2 kinds of CPU speed：Fsys=Fosc/1 , Fosc/2**
4. **2 kinds of CPU operating mode：Normal or Sleep**
5. **512 x 16 bits MTP ROM**
6. **24 x 8 bits General Purpose RAM**
7. **CPU core**
   - P9 CPU core：62 powerful instructions
   - Instructions：1 instruction cycle
   - Branches instructions：2 instruction cycles
8. **Timer**
   - Timer 0：8-bits Timer
9. **I/O Pins**
   - Bi-directional I/O： PA0/PA4/T0CKI, PA5, PA6, Port B, PY
   - Built-in pull-up： PA0/PA4/T0CKI, PA7/ RESETB, Port B, PY
   - Wake-up/Interrupt：PA7/ RESETB, PB4, PB5, PB6, PB7, PY
   - Built-in transistor for IR LED drive: PX (320mA@3V with Vol=1.5V)
10. **Interrupt handling**
    - PB7:PB4 interrupt : Interrupt on change trigger
    - PB0/INT external interrupt : Rising/Falling edge trigger
    - Timer 0 interrupt
    - Two-level deep hardware stack
11. **Power-on Reset (POR)**
12. **Device Reset Timer (DRT)**
13. **Oscillator Start-up Timer (OST)**
14. **Watchdog Timer (WDT) with its own on-chip RC oscillator**
15. **Configurable WDT wake-up: Reset/Continue**
16. **CMOS MTP technology with fully static design**
17. **Wide temperature range: -40 to 85 degree C**
18. **I/O Driving capability**

    VDD=3.0V
    - Source Current：-4mA (Voh = 2.7v)
    - Sink Current ：10.0mA (Vol = 0.3v)
19. **16 SOP package form**

3. Pin Description



16 PIN



16 PIN

Table 3-1: Device Pin Description

Pin Name	Pin Type	Buffer Type	Description
VDD	P	–	Power supply
VSS	P	–	Ground
PB0/INT	I/O	ST	General purpose I/O pin (IPU) External interrupt pin
PB1	I/O	ST	General purpose I/O pin (IPU)
PB2	I/O	ST	General purpose I/O pin (IPU)
PB3	I/O	ST	General purpose I/O pin (IPU)
PB4	I/O	ST	General purpose I/O pin (IPU) Interrupt on change pin
PB5	I/O	ST	General purpose I/O pin (IPU) Interrupt on change pin
PB6	I/O	ST	General purpose I/O pin (IPU) Interrupt on change pin
PB7	I/O	ST	General purpose I/O pin (IPU) Interrupt on change pin
PX	O	–	General purpose output pin IR LED drive pin
PY	I/O	ST	General purpose I/O (IPU) with wake-up function pin
PA0/T0CKI/PA4	I/O	ST	General purpose I/O pin (IPU) External timer/counter input
PA5	I/O	ST	General purpose I/O pin
PA6	I/O	ST	General purpose I/O pin
PA7/RESETB	I/P	CMOS	External reset input with pull-up resistor, active low General purpose input (IPU) with wake-up function pin

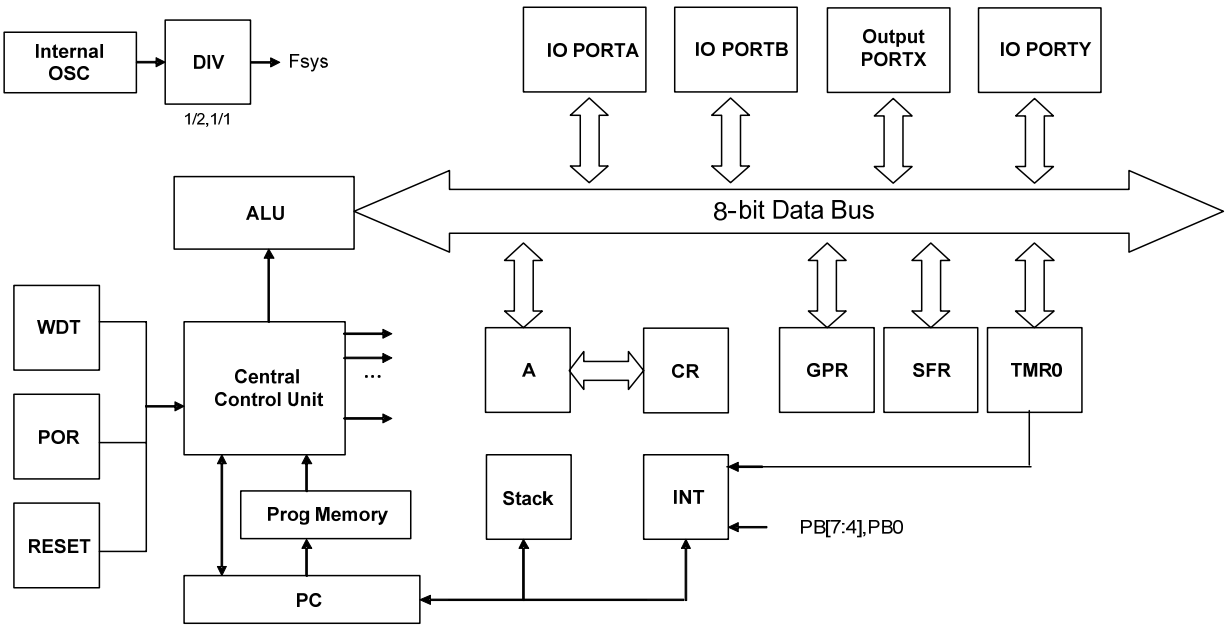
Abbreviations: I=input, O=output, P=supply, IPU=input pull-up, ST=Schmitt Trigger input

4. Architecture

The block diagram of M151A is shown below. Harvard architecture is adopted, which referred to computer architectures that uses physically separate storage and signal pathways for their instructions and data. Therefore, program and data memory blocks are separated and can be organized with different bus width. 16-bit and 8-bit wide buses are used in the program and data memory, respectively. The program memory size of the M151A is 512 words. Data memory is divided into two parts: special function registers (SFR) and general purpose registers (GPR). The ALU is 8 bits wide and can do operations with operands from A (accumulator register), GPR, SFR, or immediate constant. Register A is a special register used for ALU operations.

Internal power-on reset (POR) and external reset can be used as device reset source. Watchdog timer (WDT), Timer 0(TMR 0), internal OSC, two-level deep stack and interrupt handling capability are helpful to improve system cost and power. Bi-directional I/O pins are grouped into A, B, and PY. Oscillator start-up timer (OST) increases the reliability of the oscillators.

Figure 4-1: M151A Block Diagram



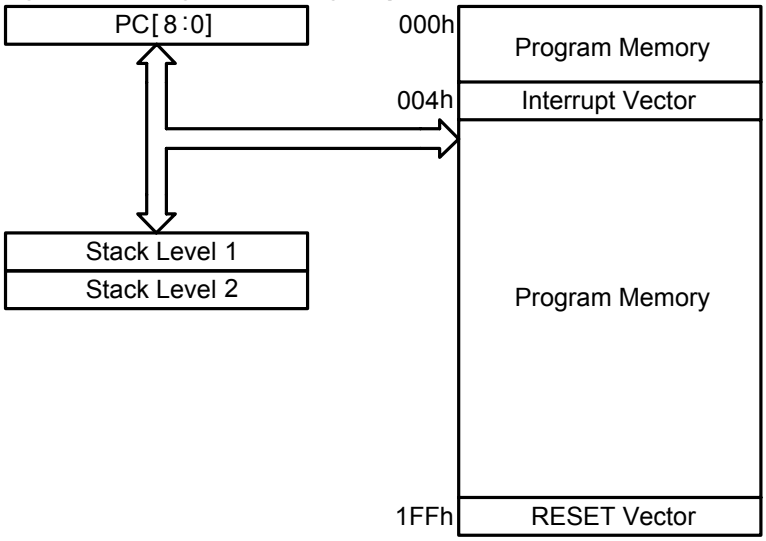
5. Memory Maps and Registers

Memory in M151A includes program memory, configuration memory, data memory, and control registers.

5.1. Program Memory Map

Program memory contains user program code data in MTP, which is addressed via the program counter (PC) register. Its capacity is 512 words with length of 16 bits. While the PC overflows during execution, it will cause a wraparound. Two locations are reserved as default vector. The RESET vector is at 1FFh, and the interrupt vector is at 004h. Figure 5-1 and 5-2 show the program memory map.

Figure 5-1: Program Memory Map



Program Counter (PC). The PC contains the MTP address of the next instruction to be executed. The PC is auto-incremented after the content of current address is fetched. The least significant byte of the PC (PCL) is accessible and mapped in data memory at address 02h. To execute a relative jump, operate PCL and offset in ALU and store result in PCL, which causes update of PC value. Ways to change program counter includes:

JMP Instruction	PC = instruction word[8:0]
JSR Instruction	PC = instruction word[8:0]
SJMP Instruction	PC = instruction word[8:0]
SJSR Instruction	PC = {0,instruction word[7:0]}
Relative Branch (modify PCL)	PC = +/- offset ({PCH[0], PCL} → PC)
Reset	PC = 1FFh
Interrupt	PC = 004h
RET and RETI Instructions	PC = STACK[top]
Normal Instructions	PC = PC + 1

STACK. The M151A includes a two-level deep hardware stack. The stack consists of two separate register and a 1-bit counter. The counter always points an empty location. When a subroutine call or an interrupt request occurs, the content of PC is firstly pushed into STACK and then the counter is incremented. When a return occurs (either from subroutine or interrupt), the counter is firstly decremented and STACK value pointed by the counter is then stored back into the PC.

5.2. Configuration Memory Map

Configuration memory contains device configuration data in MTP, which is addressed only in device programming mode.

Table 5-1: Configuration 0

						9			6	5		3			
–	–	–	–	–	–	S16M	–	–	OST	WDT		DIV			

S16M: Internal OSC frequency selection
1 = 16MHz
0 = 4MHz

OST: Oscillator start-up timer enable bit
1 = 128 cycles for BR16M and BR4M
0 = 8 cycles for BR16M and BR4M

WDT: Watchdog Timer enable bit
1 = Enable
0 = Disable

DIV: Main clock divider
1 = Divide ratio 1
0 = Divide ratio 2

Table 5-2: Configuration 1

				11	10	9		7	5	0
–	–	–	–	ENPA0	ENPA7B	DRT	–	WUP	–	OC

ENPA0: PA0/T0CKI/PA4 select bit
1 = PA0/T0CKI/PA4 is configured as PA0
0 = PA0/T0CKI/PA4 is configured as T0CKI/PA4

ENPA7B: PA7/ RESETB select bit
1 = PA7/ RESETB is configured as RESETB
0 = PA7/ RESETB is configured as PA7

DRT: Device reset timer period
1 = 29ms
0 = 3.7ms

WUP: Watchdog timer wakes CPU up under HALT mode
1 = CPU continues to execute next instruction after Watchdog Timer is overflow
0 = CPU is reset after Watchdog Timer is overflow

OC: Internal OSC calibration data
The output frequency of internal OSC is proportional to OC[5:0]. The tuning step is 1.5%.
111111 = Slowest
000000 = Fastest

5.3. Data Memory Map and Registers

Data memory contains the General Purpose Registers (GPR) and the Special Functions Registers (SFR). The General Purpose Registers are used for data under command of the instructions and can be accessed directly or indirectly. The Special Function Registers have various specific functions, like addressing control, flags indicator, or peripheral control. Figure 5-2 shows the SFR & GPR memory map.

Figure 5-2: SFR & GPR Memory Map

00h	INDX
01h	TMR0
02h	PCL
03h	STATUS
04h	FSR
05h	PORTA
06h	PORTB
07h	General Register
08h	INTCON
09h 1Fh	General Registers
	Unimplemented
50h	OPTION
51h	OC
	Unimplemented
55h	TRISA
56h	TRISB
	Unimplemented
5Ah	PCH
	Unimplemented
5Dh	PSCCNT
	Unimplemented
5Fh	PCON
	Unimplemented
65h	PAPU
	Unimplemented
74h	ACC
75h	TBDH

Table 5-3: Special Function Registers summary

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
00h	INDX	Address this location for indirect addressing								0000 0000	0000 0000
01h	TMR0	Timer 0 counter value								xxxx xxxx	uuuu uuuu
02h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	1111 1111	1111 1111
03h	STATUS(1)	PG2	PG1	PG0	TO_	PD_	Z	H	C	0001 1xxx	000p puuu
04h	FSR	1	b6	b5	b4	b3	b2	b1	b0	1000 0000	1uuu uuuu
05h	PORTA(2)	PA7	PA6	PA5	PA4	b3	b2	b1	PA0	xxxx 000x	uuuu 000u
06h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx	uuuu uuuu
08h	INTCON	GIE	-	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 0000	0000 0000
50h	OPTION	PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
51h	OC	-	-	b5	b4	b3	b2	b1	b0	--cc cccc(4)	--cc cccc(4)
55h	TRISA(3)	WKRST	b6	b5	b4	b3	b2	b1	b0	0111 1111	0111 1111
56h	TRISB	b7	b6	b5	b4	b3	b2	b1	b0	1111 1111	1111 1111
5Ah	PCH	-	-	-	-	-	-	-	PCH0	---- ---0	---- ---0
5Dh	PSCCNT	Prescaler counter value								0000 0000	0000 0000
5Fh	PCON	PYW	PYH	PYC	PYD	-	-	PXC	PXD	001x --1x	001u --1u
65h	PAPU	-	-	-	-	-	-	-	PA0H	---- ---0	---- ---0
74h	ACC	b7	b6	b5	b4	b3	b2	b1	b0	xxxx xxxx	uuuu uuuu
75h	TBDH	b7	b6	b5	b4	b3	b2	b1	b0	xxxx xxxx	uuuu uuuu

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.
Note 1: STATUS[7:5] is general purpose read/write bits.
Note 2: PORTA[3:1] is general purpose read/write bits.
Note 3: TRISA[3:1] is general purpose read/write bits.

◆ **INDX (00h)**

Indirect data addressing uses the registers INDX(00h) and FSR(04h). INDX is not physically implemented and FSR is a pointer. Any instruction addressing INDX actually accesses data pointed by the FSR register.

◆ **PCL (02h)**

Program counter (PC) contains the address of the next instruction to be executed. The least significant byte of the PC (PC[7:0]) is defined as PCL which is accessible in date memory address 02h.

◆ **STATUS (03h)**

7	6	5	4	3	2	1	0
b7	b6	b5	TO_	PD_	Z	H	C

C: Carry/NOT borrow flag.
0: No carry-out has occurred
1: A carry-out has occurred

H: Half carry/NOT borrow flag.
0: No carry-out has occurred from the lower nibble
1: A carry-out has occurred from the lower nibble

Z: Zero flag.
0: The result of an arithmetic or logical operation is not zero
1: The result of an arithmetic or logical operation is zero

PD_: Power down bit, read only
0: By executing HALT instruction
1: After power-up, or executing CLRW instruction

TO_: Time-out bit, read only
0: Reset by watchdog timer time-out
1: After power-up, or executing CLRW instruction or HALT instruction

b[7:5]: General purpose read/write bits.

◆ **FSR (04h)**

The FSR is a 7-bit index register used in indirect addressing mode as pointers to memory locations. Any instruction addressing INDX(00h) actually accesses data pointed by the FSR(04h) register.

◆ **PCH (5Ah)**

The PCH register is used in two situations: (1) when a write to PCL, PC is updated with PCH and PCL (PC[8] ← PCH[0], PC[7:0] ← PCL[7:0].) (2) when executing a MOVC instruction, PCH is used as upper bits of address (TBDH, A ← ROM[PCH, A])

◆ **ACC (74h)**

ACC is the accumulator register. If executing MOVC instruction, it contains the low byte of table data after operation.

◆ **TBDH (75h)**

TBDH contains the high byte of table data after executing MOVC instruction.

5.4. Control Registers

M151A uses PEEK and POKE instruction to load control registers to A and save A to control registers, respectively. Figures 5-4 shows the Control Registers memory map. Note that Registers locate at 00h-2Fh are not only Control Registers, but also Special Functions Registers.

Figure 5-4: Control Registers Memory Map

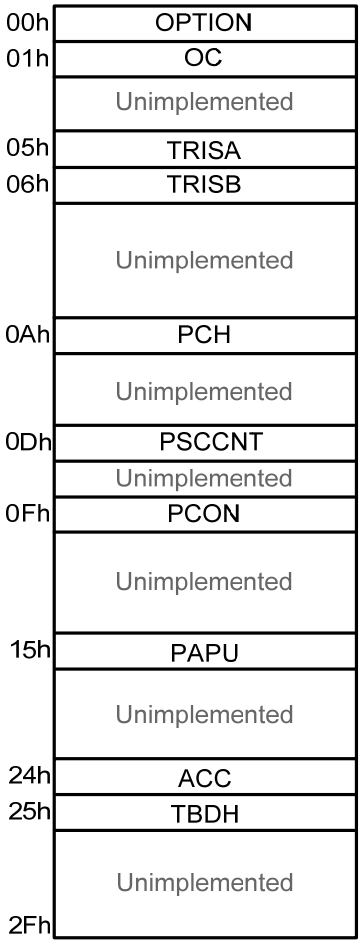


Table 5-5: Control Registers summary

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
00h	OPTION	PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
01h	OC	-	-	b5	b4	b3	b2	b1	b0	--cc cccc(4)	--cc cccc(4)
05h	TRISA(3)	WKRST	b6	b5	b4	b3	b2	b1	b0	0111 1111	0111 1111
06h	TRISB	b7	b6	b5	b4	b3	b2	b1	b0	1111 1111	1111 1111
0Ah	PCH	-	-	-	-	-	-	-	PCH0	---- ---0	---- ---0
0Dh	PSCCNT	Prescaler counter value								0000 0000	0000 0000
0Fh	PCON	PYW	PYH	PYC	PYD	-	-	PXC	PXD	001x --1x	001u --1u
15h	PAPU	-	-	-	-	-	-	-	PA0H	---- ---0	---- ---0
24h	ACC	b7	b6	b5	b4	b3	b2	b1	b0	xxxx xxxx	uuuu uuuu
25h	TBDH	b7	b6	b5	b4	b3	b2	b1	b0	xxxx xxxx	uuuu uuuu

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.
Note 3: TRISA[3:1] is general purpose read/write bits.

Note: Any read from unimplemented locations returns unknown data.

6. Clocks

6.1. Clock System

An internal 4MHz/16MHz RC oscillator is integrated in this CPU. An optional oscillator start-up timer (OST) is used to ensure that the oscillator has started and stabilized. It provides a delay of 8 or 128 cycles after oscillator is enable for OST configuration bit being clear or set, respectively. A clock divider provides divide ratio of 1 or 2. The CPU clock is derived after this divider.

Figure 6-1: Block Diagram of Clock System

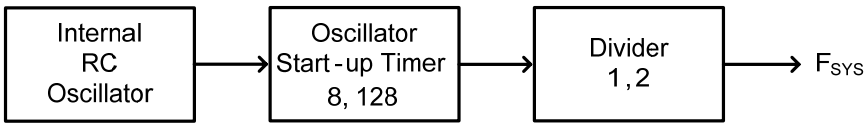


Table 6-1: CPU Start-up Time

Mode	Power-up				Wake-up from HALT	
	OST=0		OST=1		OST=0	OST=1
	DRT=0	DRT=1	DRT=0	DRT=1		
BR16M, BR4M	3.7ms+8 cycles	29ms+8 cycles	3.7ms+128 cycles	29ms+128 cycles	8 cycles	128 cycles

Note: 1 cycle = 1/Fsys

6.2. Internal 4MHz/16MHz RC Oscillator (BR4M and BR16M)

The M151A integrated an internal RC oscillator that provides 4 MHz output clock at VDD = 3V and 25 °C with an accuracy of ± 1% after calibration. Another provides 16 MHz output clock at VDD = 3V and 25 °C with an accuracy of ± 3% after calibration.

The Oscillator Calibration Register (OC) contains the calibration data fetched from the Configuration1[5:0] after power-up. The OC register can be modified dynamically while program is running. The output frequency of internal OSC is proportional to the complement of OC[5:0], that is, the frequency of internal OSC is lowest when OC[5:0]=0x3F and highest frequency when OC[5:0]=0x00. The tuning step of OC is about 1.5%.

◆ **Configuration 0**

						9			6	5		3			
–	–	–	–	–	–	S16M	–	–	OST	WDT		DIV			

S16M: Internal OSC frequency selection
1 = 16MHz
0 = 4MHz

OST: Oscillator start-up timer enable bit
1 = 128 cycles for BR16M and BR4M
0 = 8 cycles for BR16M and BR4M

DIV: Main clock divider
1 = Divide ratio 1
0 = Divide ratio 2

◆ **Configuration 1**

				11	10	9		7	5	0
–	–	–	–	ENPA0	ENPA7B	DRT	–	WUP	–	OC

DRT: Device reset timer period
1 = 29ms
0 = 3.7ms

OC: Internal OSC calibration data
The output frequency of internal OSC is proportional to OC[5:0]. The tuning step is 1.5%.
111111 = Slowest
000000 = Fastest

Special Function Registers

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
51h	OC	-	-	b5	b4	b3	b2	b1	b0	--cc cccc(1)	--cc cccc(1)

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.

◆ **OC (51h)**

OC contains internal OSC calibration data. Setting “111111” causes the slowest of internal OSC and “000000” causes the fastest. The tuning step is 1.5%.

7. I/O Ports

The I/O registers (PORTA and PORTB), I/O control registers (TRISA, TRISB), and PCON are used to manipulate I/O pins. I/O register can be read and written. However, read instructions always read the states of the corresponding I/O pins.

7.1. PORTA

PA0/T0CKI/PA4 can be configured as PA0 or T0CKI/PA4. PA0, shown in Figure 7-1, integrate weak pull-up function, which can be turn on by setting PA0H in PAPU register. Circuit of T0CKI/PA4 pins and corresponding I/O register are shown in Figure 7-2. PA7, shown in Figure 7-4, is an input port with a weak pull-up function and is enabled when bit 10 of Configuration 1 is clear. When WKRST (TRISA.7) is set, a falling edge transition on PA7 will wake CPU from HALT.

7.2. PORTB

PORTB is an 8-bit I/O Register. Each PB pin integrates a weak pull-up function, which can be turn on by clearing PBPUB bit in OPTION register. In output mode, the pull-up is turned off automatically. PB[7:4] pins have an interrupt-on-change feature. Only input pins can cause interrupt to occur. To use the interrupt-on-change feature, user needs to read PORTB first. The value on PORTB will be latched. After that, the input pins of PB[7:4] are compared with the old value latched on the last read of PORTB. An interrupt is generated when a mismatch occurs. And clear the interrupt in the following manner:

(a) Any read or write of PORTB. This will end the mismatch condition.
(b) Clear flag bit RBIF.

PB0/INT is an external interrupt input pin. Figure 7-5, 7-6, and 7-7 show the PB pins circuit diagram.

7.3. PORTX, PORTY

PORTX is an output port and PORTY is a general purpose I/O port with pull-up functions. PORTX can be used for remote control transmitter. The low-level output current of PORTX is 320mA at 3V and Vol is 1.5V. The initial status of PORTY is input mode with pull-up function disabled. In output mode, the pull-up is turned off automatically. In input mode, the pull-up function can be turned on by setting PYH. PORTY is designed to have a wake-up function. When PYW is set, a falling edge transition on PY will wake CPU from HALT. The PX and PY pins circuit diagrams are shown in Figure 7-8 and 7-9.

Figure 7-1: Block Diagram of PA0 Pin

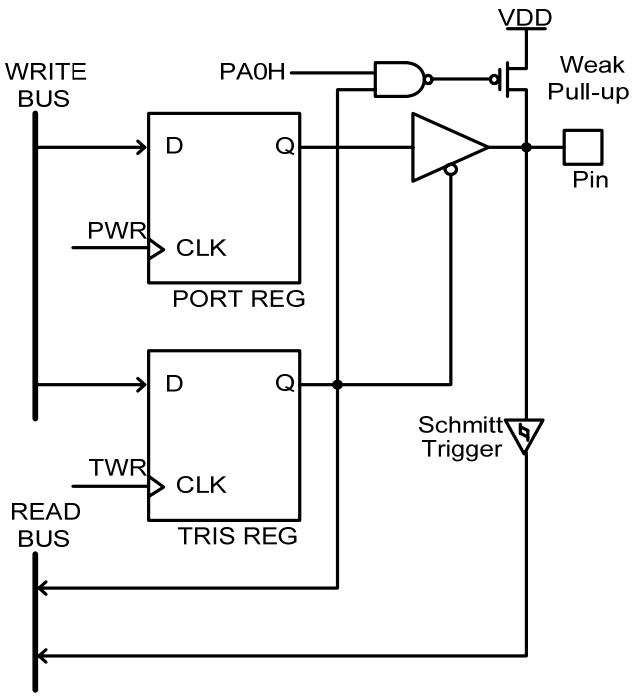


Figure 7-2: Block Diagram of PA4 Pin

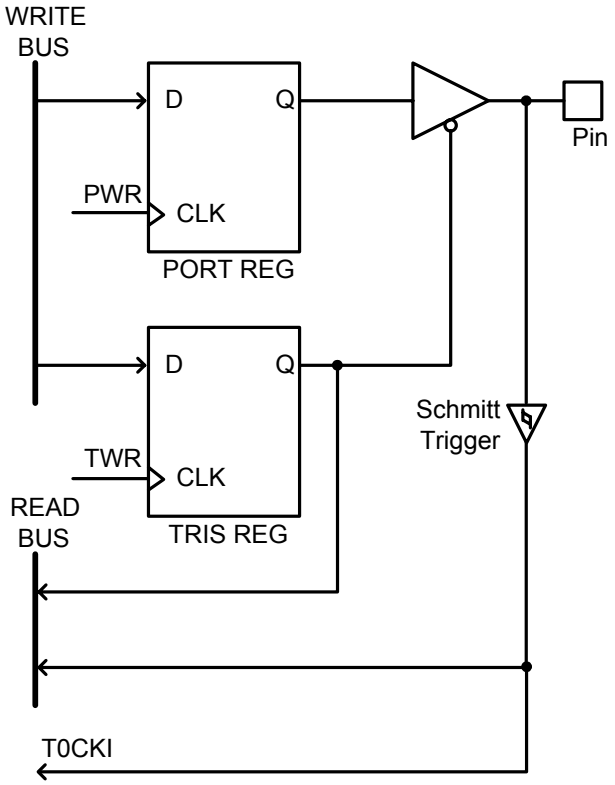


Figure 7-3: Block Diagram of PA5 and PA6 Pins

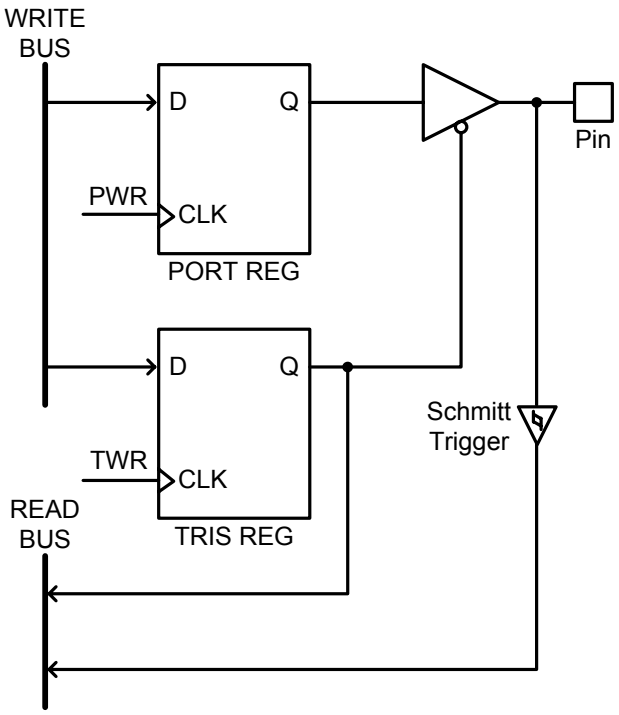


Figure 7-4: Block Diagram of PA7 Pin

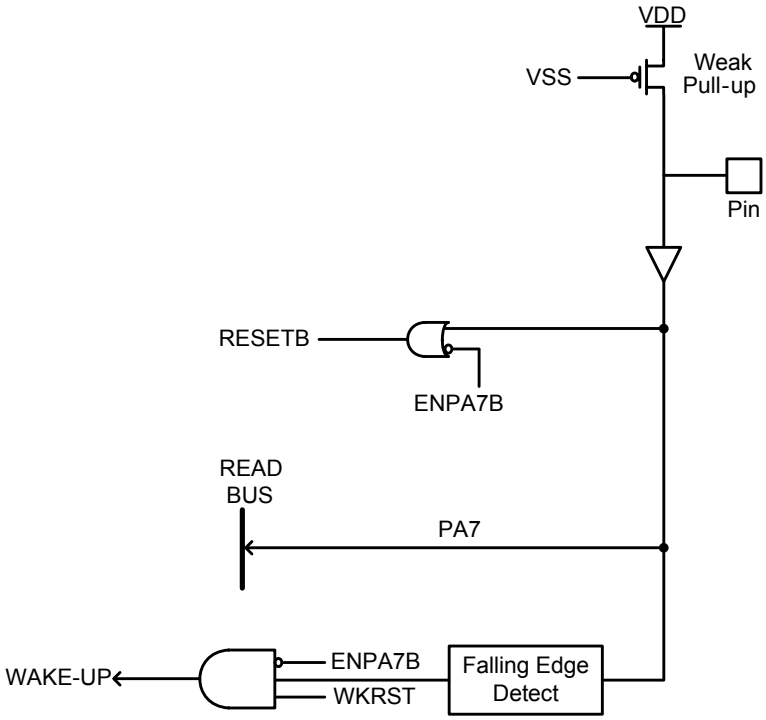


Figure 7-5: Block Diagram of PB0 Pin

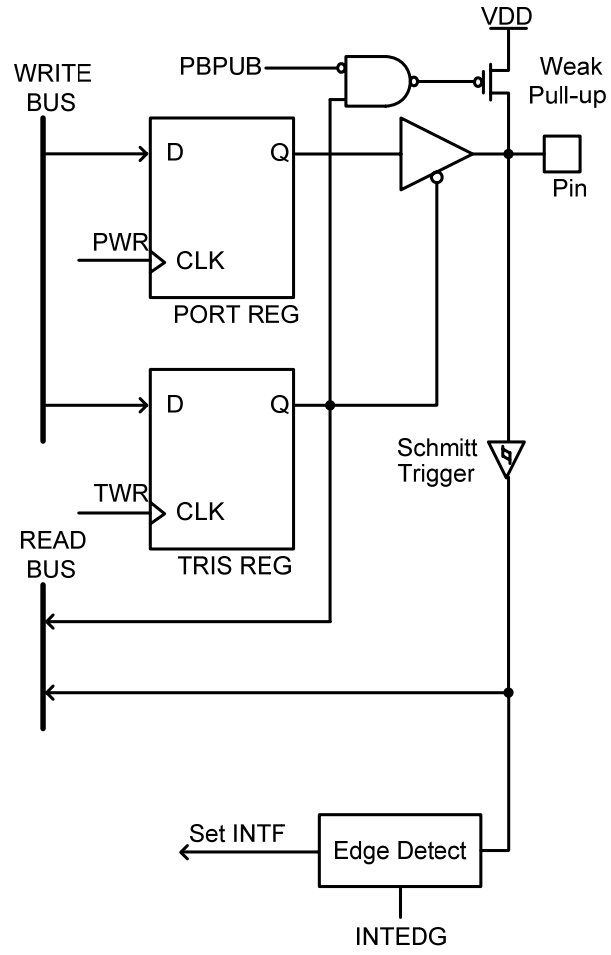


Figure 7-6: Block Diagram of PB[3:1] Pins

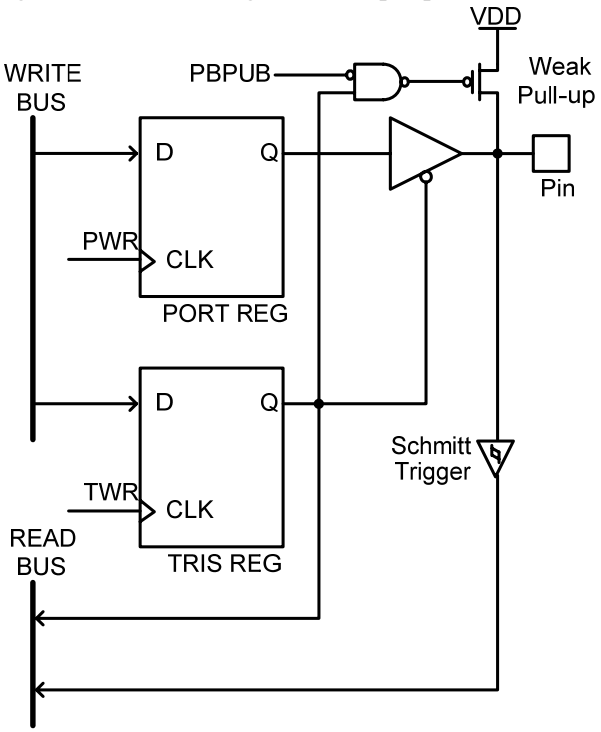


Figure 7-7: Block Diagram of PB[7:4] Pins

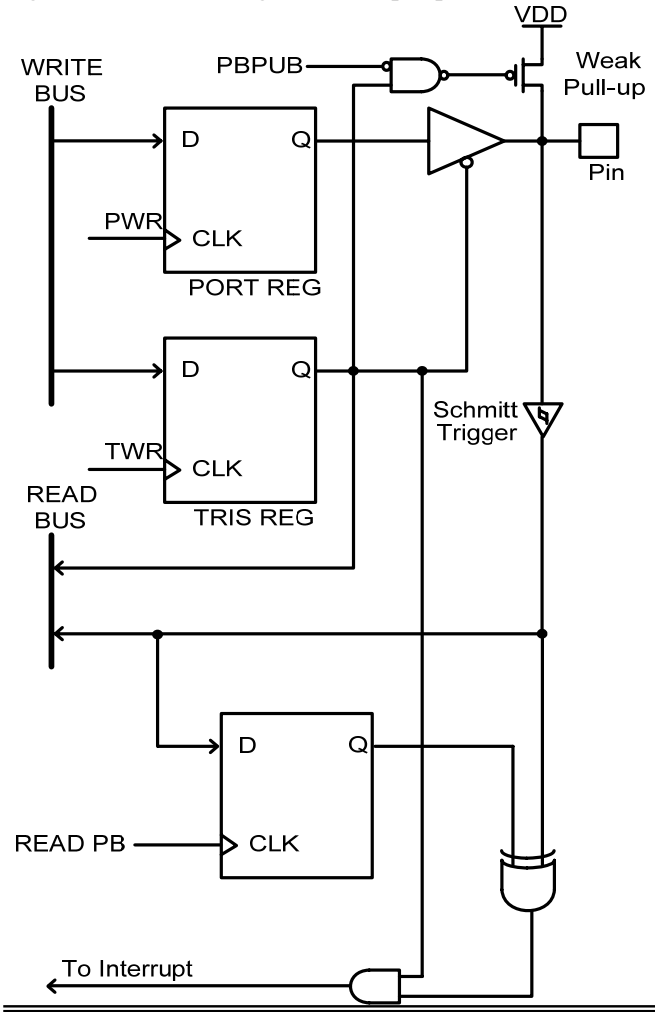


Figure 7-8: Block Diagram of PX Pin

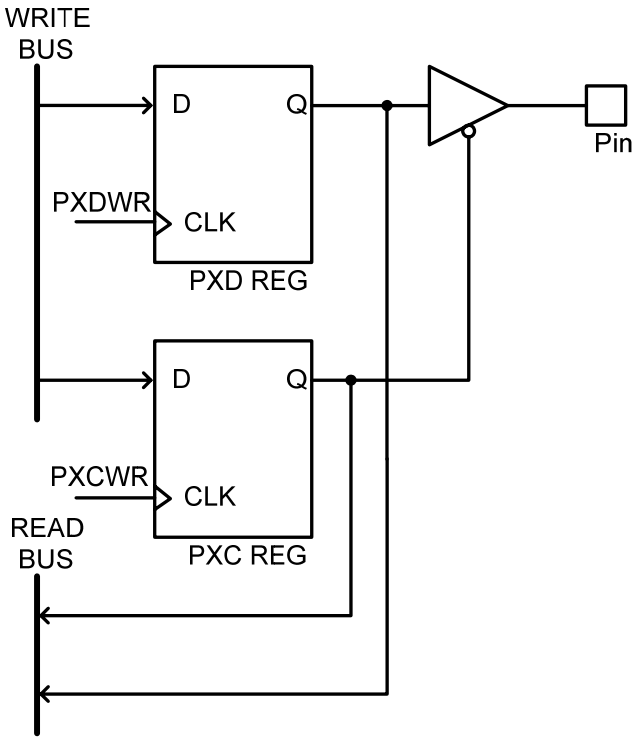
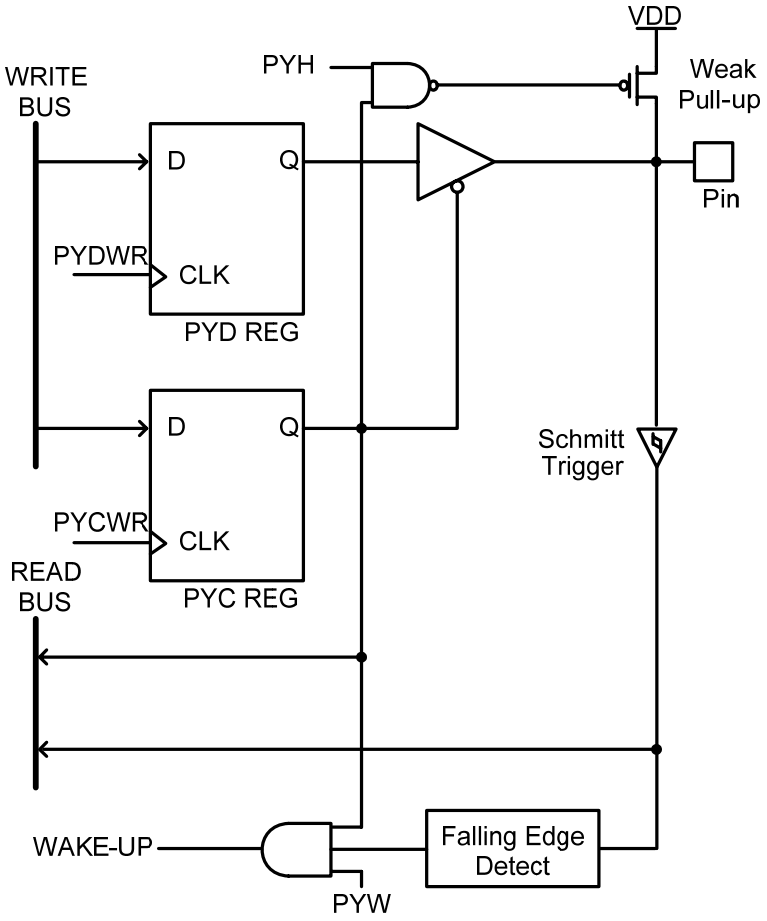


Figure 7-9: Block Diagram of PY Pin



◆ **Configuration 1**

				11	10	9		7	5	0
–	–	–	–	ENPA0	ENPA7B	DRT	–	WUP	–	OC

ENPA0: PA0/T0CKI/PA4 select bit
1 = PA0/T0CKI/PA4 is configured as PA0
0 = PA0/T0CKI/PA4 is configured as T0CKI/PA4

ENPA7B: PA7/ RESETB select bit
1 = PA7/ RESETB is configured as RESETB
0 = PA7/ RESETB is configured as PA7

Special Function Registers

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
05h	PORTA(2)	PA7	PA6	PA5	PA4	b3	b2	b1	PA0	xxxx 000x	uuuu 000u
06h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx	uuuu uuuu
50h	OPTION	PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
55h	TRISA(3)	WKRST	b6	b5	b4	b3	b2	b1	b0	0111 1111	0111 1111
56h	TRISB	b7	b6	b5	b4	b3	b2	b1	b0	1111 1111	1111 1111
5Fh	PCON	PYW	PYH	PYC	PYD	-	-	PXC	PXD	001x --1x	001u --1u
65h	PAPU	-	-	-	-	-	-	-	PA0H	---- ---0	---- ---0

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.
Note 2: PORTA[3:1] is general purpose read/write bits.
Note 3: TRISA[3:1] is general purpose read/write bits.

◆ **PORTA (05h)**

7	6	5	4	3	2	1	0
PA7	PA6	PA5	PA4	b3	b2	b1	PA0

PA[7:4] , PA0: PA[7:4] and PA0 is the value of the I/O port A.
0: Output mode
1: Input mode
b[3:1] : General purpose read/write bits.

◆ **PORTB (06h)**

7	6	5	4	3	2	1	0
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

PORTB is the value of the I/O port B.

◆ **OPTION (50h)**

7	6	5	4	3	2	1	0
PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

PBPUB: Port B pull-up enable bit
0: Port B pull-up are enabled.
1: Port B pull-up are disabled

◆ **TRISA (55h)**

7	6	5	4	3	2	1	0
WKRST	b6	b5	b4	b3	b2	b1	b0

WKRST: PA7/ RESETB wake-up enable bit
0: Disable
1: Enable

b[6:4] , b[0]: PA[6:4] and PA0 input/output mode selection
0: Output mode
1: Input mode
b[3:1] : General purpose read/write bits.

◆ **TRISB (56h)**

7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

Port B input/output mode selection. '1' is input mode and '0' is output mode.

◆ **PCON (5Fh)**

7	6	5	4	3	2	1	0
PYW	PYH	PYC	PYD	-	-	PXC	PXD

PYW: PY wake-up enable bit
0: Disable
1: Enable
PYH: PY pull-up enable bit
0: PY pull-up is disabled.
1: PY pull-up is enabled
PYC: PY input/output mode selection
0: Output mode
1: Input mode
PYD: The value of the I/O port PY
PXC: PX output enable bit
0: Enable
1: Disable
PXD: The value of port PX

◆ **PAPU (65h)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	PA0H

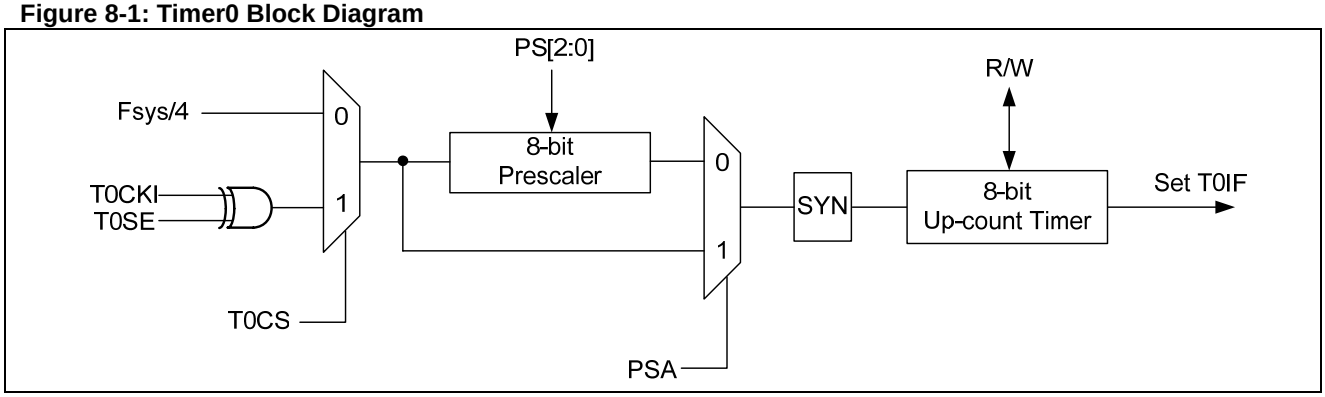
PA0H: PA0 pull-up enable bit
0: PA0 pull-up is disabled.
1: PA0 pull-up is enabled

8. On-Chip Peripherals

8.1. Timer 0

The Timer0 is an 8-bit up-count counter. It is readable and writable. The Timer0 has internal or external clock source. Besides, an 8-bit readable prescaler is provided to Timer0 if PSA (OPTION.3) is 0. Note that when changing prescaler from the WDT to the Timer0 or vice versa, CLRW instruction should be executed before switching the prescaler. When selecting clock source input from T0CKI, Timer0 will increment either on every rising or falling edge of T0CKI according to the setting of T0SE (OPTION.4). A synchronization circuit is used in Timer0. Any write to the TMR0 Register will cause a two instruction cycle inhibit. After that, Timer0 continues to increase from the new value. When T0CKI is used for Timer0 without prescaler, it is necessary for T0CKI to be high for at least 2Tsys and low for at least 2Tsys to meet the sampling requirement.

Figure 8-1: Timer0 Block Diagram



Note: The prescaler may be used by either the Timer0 module or the WDT(Figure 8-2), but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT and vice versa.

Special Function Registers

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
01h	TMR0	Timer 0 counter value								xxxx xxxx	uuuu uuuu
50h	OPTION	PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
5Dh	PSCCNT	Prescaler counter value								0000 0000	0000 0000

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.

- **TMR0 (01h)**

7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

TMR0 contains the value in the counter of the Timer 0.

- **OPTION (50h)**

7	6	5	4	3	2	1	0
PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

PS2-0: Prescaler divide ratio

PS[2:0]	TMR0	WDT
000	2	1
001	4	2
010	8	4
011	16	8
100	32	16
101	64	32
110	128	64
111	256	128

PSA: Prescaler assignment bit
0: Prescaler is assigned to Timer 0
1: Prescaler is assigned to WDT

T0SE: Timer 0 source edge select bit
0: Increment on rising edge of T0CKI pin
1: Increment on falling edge of T0CKI pin

T0CS: Timer 0 clock source select bit
0: Internal instruction cycle clock
1: T0CKI pin

◆ **PSCCNT (5Dh)**

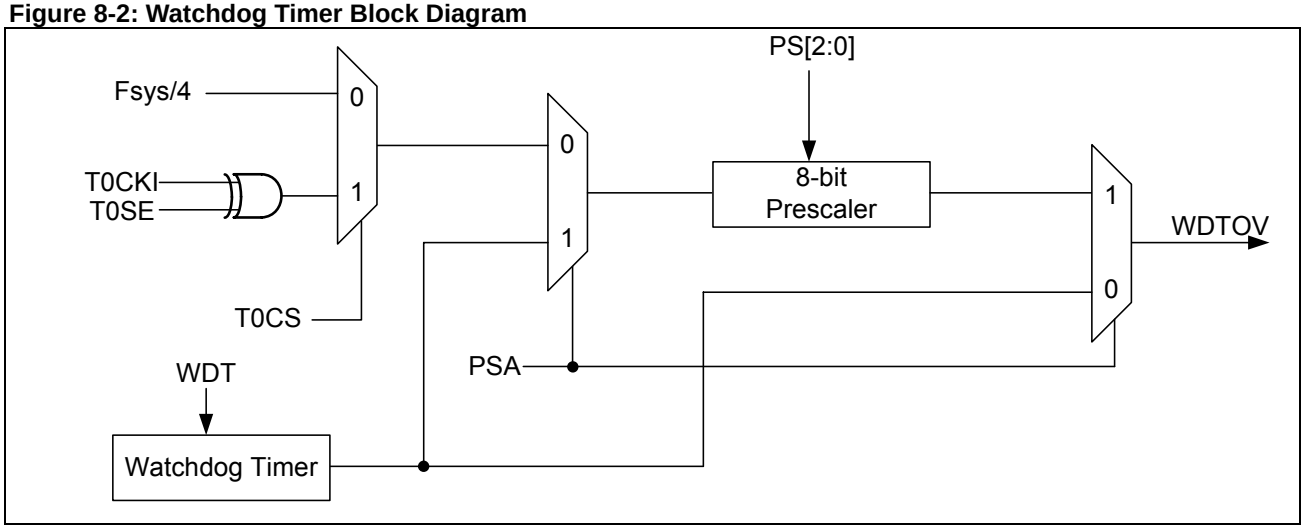
7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

PSCCNT is a read-only register which contains the value in the counter of the prescaler. The prescaler is programmable for Timer0 or watchdog timer.

8.2. Watchdog Timer

Watchdog Timer is enabled when bit 5 of Configuration 0 is set. The period of time-out (WDTOV) is 29ms (with no prescaler). If the prescaler is assigned to the Watchdog Timer, it causes a longer time-out period depending on the setting on PS[2:0] of OPTION register. Note that when changing prescaler from the WDT to the Timer0 or vice versa, CLRW instruction should be executed before switching the prescaler. A reset is generated when Watchdog Timer is overflow. The TO_ bit in the STATUS register will be cleared upon a Watchdog Timer time-out. The CLRW and HALT instructions clear Watchdog Timer.

Figure 8-2: Watchdog Timer Block Diagram



◆ **Configuration 0**

						9			6	5		3			
–	–	–	–	–	–	S16M	–	–	OST	WDT		DIV			

WDT: Watchdog Timer enable bit
1 = Enable
0 = Disable

◆ **Configuration 1**

				11	10	9		7		5	0
–	–	–	–	ENPA0	ENPA7B	DRT	–	WUP	–	OC	

WUP: Watchdog timer wakes CPU up under HALT mode
1 = CPU continues to execute next instruction after Watchdog Timer is overflow
0 = CPU is reset after Watchdog Timer is overflow

Special Function Registers

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
50h	OPTION	PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.

◆ **OPTION (50h)**

7	6	5	4	3	2	1	0
PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

PS2-0: Prescaler divide ratio

PS[2:0]	TMR0	WDT
000	2	1
001	4	2
010	8	4
011	16	8
100	32	16
101	64	32
110	128	64
111	256	128

PSA: Prescaler assignment bit
0: Prescaler is assigned to Timer 0
1: Prescaler is assigned to WDT

T0SE: Timer 0 source edge select bit
0: Increment on rising edge of T0CKI pin
1: Increment on falling edge of T0CKI pin

T0CS: Timer 0 clock source select bit
0: Internal instruction cycle clock
1: T0CKI pin

9. Special Features

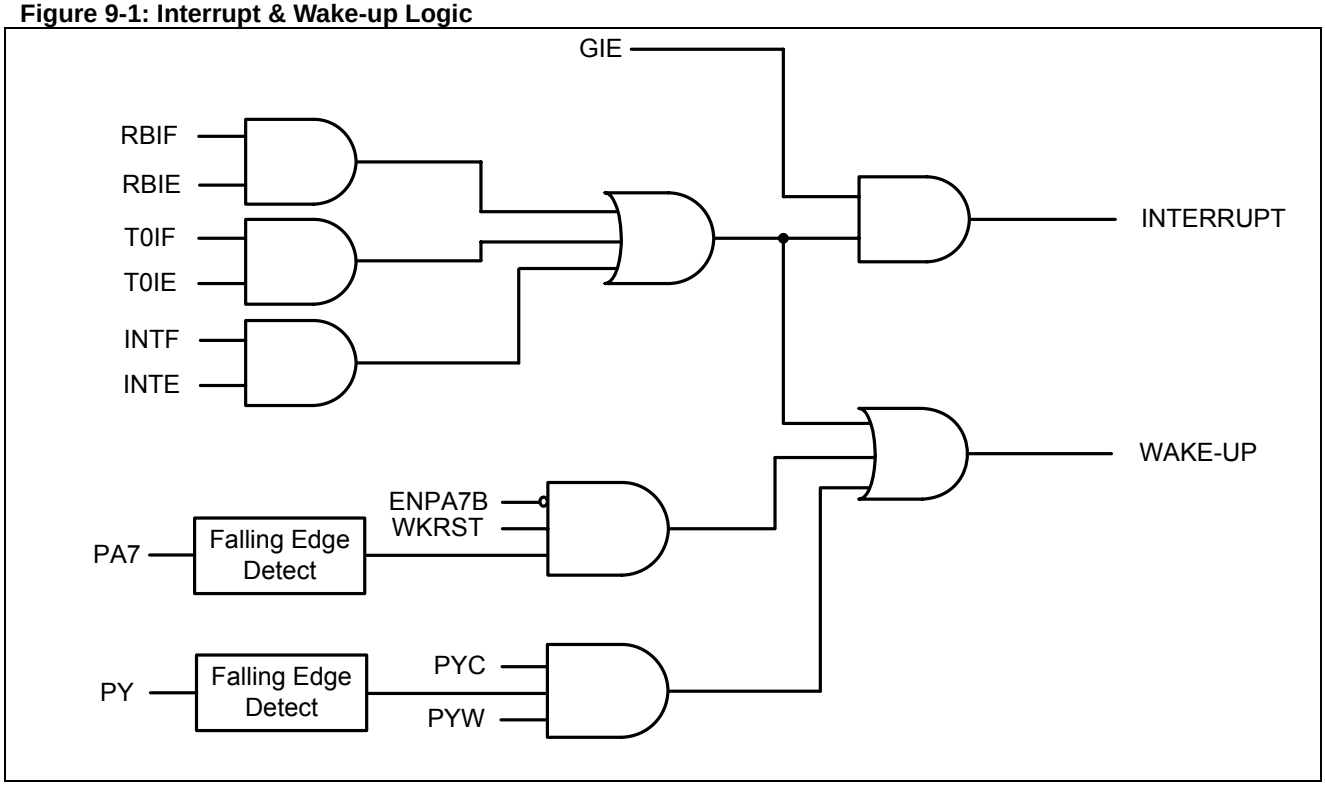
9.1. Interrupts

In interrupt mode there are three sources of interrupt: external INT, interrupt from PORTB, and interrupt from time-out of Timer 0. When the global interrupt enable bit (GIE) is set and an interrupt condition occurs, its corresponding flag in register file INTCON will be set to '1'. If its corresponding interrupt enable bit in register file INTCON is set, it requests to go into interrupt service routine. After it enters Interrupt Service Routine (ISR), the hardware clears GIE and will not generate any interrupt request even though there is another interrupt to prevent recursive interrupts. Within ISR, user needs to judge from INTCON to decide which kind of interrupt and then interrupt flag needs to be cleared by program. If multiple interrupt sources request simultaneously, the priority should be determined by program. When the first ISR is complete and returns from RETI, it enters ISR again if there is another interrupt request until all interrupts are taking care.

9.2. Wake-up

By executing a HALT instruction, CPU enters power-down mode. In power-down mode there are six sources that can wake CPU from HALT: external reset input on RESETB, time-out of Watch-dog timer, external INT, interrupt from PORTB, a high-to-low transition on PY, and a high-to-low transition on PA7. External reset or time-out of Watch-dog timer will cause a reset. The other interrupt sources will wake CPU from HALT if the corresponding wake-up enable bit is set. Wake-up is regardless of the state of the global interrupt enable bit (GIE).

Figure 9-1: Interrupt & Wake-up Logic



◆ **Configuration 1**

11	10	9	7	5	0
– – – – ENPA0	ENPA7B	DRT	–	WUP –	OC

ENPA7B: PA7/ RESETB select bit
1 = PA7/ RESETB is configured as RESETB
0 = PA7/ RESETB is configured as PA7

Special Function Registers

Address	Label	B7	B6	B5	B4	B3	B2	B1	B0	Value on POR	Value on other reset
08h	INTCON	GIE	-	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 0000	0000 0000
55h	TRISA(3)	WKRST	b6	b5	b4	b3	b2	b1	b0	0111 1111	0111 1111

Abbreviations: x = unknown, u = unchanged, p = value depends on conditions, - = unimplemented.
Note 3: TRISA[3:1] is general purpose read/write bits.

◆ **INTCON (08h)**

7	6	5	4	3	2	1	0
GIE	-	T0IE	INTE	RBIE	T0IF	INTF	RBIF

RBIF: PB7:PB4 port change interrupt flag bit
0: None of the PB7:PB4 pins have changed state
1: At least one of the PB7:PB4 pins changed state, clear by software
INTF: PB0/INT external interrupt flag bit
0: PB0/INT external interrupt does not occur
1: PB0/INT external interrupt occurs, clear by software
T0IF: Timer 0 overflow interrupt flag bit
0: Timer 0 does not overflow
1: Timer 0 has overflowed, clear by software
RBIE: PB7:PB4 interrupt enable bit
0: Disable interrupt
1: Enable interrupt
INTE: PB0/INT external interrupt enable bit
0: Disable PB0/INT external interrupt
1: Enable PB0/INT external interrupt
T0IE: Timer 0 overflow interrupt enable bit
0: Disable Timer 0 overflow interrupt
1: Enable Timer 0 overflow interrupt
GIE: Global interrupt enable bit
0: Disable all interrupt
1: Enable interrupt function

◆ **TRISA (55h)**

7	6	5	4	3	2	1	0
WKRST	b6	b5	b4	b3	b2	b1	b0

WKRST: PA7/ RESETB wake-up enable bit
0: Disable
1: Enable
b[6:4] , b[0]: PA[6:4] and PA0 input/output mode selection
0: Output mode
1: Input mode
b[3:1] : General purpose read/write bits.

9.3. Programming PIN

Table 9.1 Parallel mode programming information

Programming information		
Writer connector		Parts: M151A
Name	Description	Name
Power and clock Pins		
VDD	3V	VDD
GND	Ground	VSS
VPP	7.5V	PA7/ RESETB
CLK	CLK input	PA6
Communication Pins		
Sync. CLK	Used to sync. data I/O	PA0/T0CKI/PA4
Command/Data Bus 0	I/O pin	PB0
Command/Data Bus 1	I/O pin	PB1
Command/Data Bus 2	I/O pin	PB2
Command/Data Bus 3	I/O pin	PB3
Command/Data Bus 4	I/O pin	PB4
Command/Data Bus 5	I/O pin	PB5
Command/Data Bus 6	I/O pin	PB6
Command/Data Bus 7	I/O pin	PB7

Table 9.2 Serial mode programming information

Programming information		
Writer connector		Parts: M151A
Name	Description	Name
Power and clock Pins		
VDD	3V	VDD
GND	Ground	VSS
VPP	7.5V	PA7/ RESETB
CLK	CLK input	PA6
Communication Pins		
Sync. CLK	Used to sync. data I/O	PA0/T0CKI/PA4
Serial Command/Data Bus	I/O pin	PB7

10. Instruction Set (P9 CPU Core)

XXX dest, src

dest= 0~127 or A

src= 0~127 or A

XXX A, #k

dest= A

Table 10-1: Instruction Set

Instruction	Flag	Cycles	Description
Arithmetic			
ADC A, r	C, H, Z	1	r+A+C→ A
ADC r, A	C, H, Z	1	r+A+C→ r
ADD A, r	C, H, Z	1	r+A→ A
ADD r, A	C, H, Z	1	r+A→ r
ADD A, #k	C, H, Z	1	k + A → A
DAA A, r	C	1	r DAA → A
DAA r, r	C	1	r DAA → r
SBC A, r	C, H, Z	1	r – A –(~C) → A
SBC r, A	C, H, Z	1	r – A –(~C) → r
SUB A, r	C, H, Z	1	r – A → A
SUB r, A	C, H, Z	1	r – A → r
SUB A, #k	C, H, Z	1	k – A → A

Instruction	Flag	Cycles	Description
Logic Operation			
ANL A, r	Z	1	r&A→ A
ANL r, A	Z	1	r&A→ r
ANL A, #k	Z	1	k & A → A
CPL A, r	Z	1	~r → A
CPL r, r	Z	1	~r → r
ORL A, r	Z	1	r | A→ A
ORL r, A	Z	1	r | A→ r
ORL A, #k	Z	1	k | A → A
XRL A, r	Z	1	r ⊕ A → A
XRL r, A	Z	1	r ⊕ A → r
XRL A, #k	Z	1	k ⊕ A → A

Instruction	Flag	Cycles	Description
Rotate & Shift			
RLC A, r	C	1	{r[6:0], C} → A, r[7] → C
RLC r, r	C	1	{r[6:0], C} → r, r[7] → C
RRC A, r	C	1	{C, r[7:1]} → A, r[0] → C
RRC r, r	C	1	{C, r[7:1]} → r, r[0] → C

Instruction	Flag	Cycles	Description
Bit Operation			
RMBn r		1	0 → r[n], n=0~7
SMBn r		1	1 → r[n], n=0~7

Instruction	Flag	Cycles	Description
Data Move			
MOV r, A	(Z)	1	A → r If r=ACC, Z is updated.
MOV A, r	Z	1	r → A
MOV A, #k		1	k → A
MOVC		2	ROM code (low byte) of {PCH[0], A} → A ROM code (high byte) of {PCH[0], A} → TBDH
PEEK r	Z	1	CR[r] → A
POKE r	(Z)	1	A → CR[r] If r=ACC, Z is updated.

Instruction		Flag	Cycles	Description
Increment & Decrement				
DEC	A, r	Z	1	r -1 → A
DEC	r, r	Z	1	r -1 → r
INC	A, r	Z	1	r +1 → A
INC	r, r	Z	1	r +1 → r

Instruction		Flag	Cycles	Description
Branch				
DSZ	A, r		1 or 2	r -1 → A, skip if 0
DSZ	r, r		1 or 2	r -1 → r, skip if 0
ISZ	A, r		1 or 2	r +1 → A, skip if 0
ISZ	r, r		1 or 2	r +1 → r, skip if 0
JMP	k		2	k → PC
JSR	k		2	PC + 1 → STACK[top], k → PC
RET			2	STACK[top] → PC
RET	#k		2	k →A, STACK[top] →PC
RETI			2	1 →GIE, STACK[top] →PC
SBRn	r		1 or 2	Skip if r[n] = 0
SBSn	r		1 or 2	Skip if r[n] ≠ 0
SE	r		1 or 2	Skip if A = r
SE	#k		1 or 2	Skip if A= k
SJMP	k		2	k[8:0] → PC
SJSR	k		2	PC + 1 → STACK[top], {0, k[7:0]} → PC

Instruction		Flag	Cycles	Description
Miscellaneous				
CLR	A	Z	1	0 → A
CLR	r	Z	1	0 → r
CLRW		TO_, PD_	1	0 → WDT, 0 → WDT prescaler 1 → TO_, 1 → PD_
HALT		TO_, PD_	1	0 → WDT, 0 → WDT prescaler 1 → TO_, 0 → PD_
NOP			1	No operation
SWAP	A, r		1	{r[3:0], r[7:4]} → A
SWAP	r, r		1	{r[3:0], r[7:4]} → r
TEST	r	Z	1	r → r

10.1. Instruction Description

ADC	Add A and Reg and Carry
Format:	ADC A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r + A + C -> A
Flag Affected:	C, H, Z
Description:	Add A with register r and C. Result is stored back in A.

ADC	Add A and Reg and Carry
Format:	ADC r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r + A + C -> r
Flag Affected:	C, H, Z
Description:	Add A with register r and C. Result is stored back in r.

ADD	Add A and Reg
Format:	ADD A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r + A -> A
Flag Affected:	C, H, Z
Description:	Add A with register r. Result is stored back in A.

ADD	Add A and Reg
Format:	ADD r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r + A -> r
Flag Affected:	C, H, Z
Description:	Add A with register r. Result is stored back in r.

ADD	Add A and literal
Format:	ADD A, #k
Cycles:	1
Operands:	0 ≤ k ≤ 255
Operations:	k + A -> A
Flag Affected:	C, H, Z
Description:	Add A with 8-bit literal ‘k’. Result is stored back in A.

ANL	Logic AND A with Reg
Format:	ANL A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r & A -> A
Flag Affected:	Z
Description:	Logic AND A with register r. Result is stored back in A.

ANL	Logic AND A with Reg
Format:	ANL r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r & A -> r
Flag Affected:	Z
Description:	Logic AND A with register r. Result is stored back in r.

ANL	Logic AND A with literal
Format:	ANL A, #k
Cycles:	1
Operands:	0 ≤ k ≤ 255
Operations:	k & A -> A
Flag Affected:	Z
Description:	Logic AND A with 8-bit literal ‘k’. Result is stored back in A.

CLR	Clear register
Format:	CLR A
Cycles:	1
Operands:	A
Operations:	0 -> A, 1 -> Z
Flag Affected:	Z
Description:	Register A is clear and Z is set.

CLR	Clear register
Format:	CLR r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	0 -> r, 1 -> Z
Flag Affected:	Z
Description:	Register r is clear and Z is set.

CLRW	Clear watchdog timer
Format:	CLRW
Cycles:	1
Operands:	None
Operations:	0 -> WDT, 0 -> WDT prescaler 1 -> TO_ ->PD_
Flag Affected:	TO_, PD_
Description:	Clear watchdog time and prescaler of the WDT. Flags TO_ and PD_ are set.

CPL	Complement
Format:	CPL A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	~r -> A
Flag Affected:	Z
Description:	Complement r. Result is stored back in A.

CPL	Complement
Format:	CPL r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	~r -> r
Flag Affected:	Z
Description:	Complement r. Result is stored back in r.

DAA	Decimal-adjust after addition
Format:	DAA A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	If r[3:0] > 9 or H = 1 r[3:0] + 6 -> A[3:0] else r[3:0] -> A[3:0] If r[7:4] > 9 or C = 1 r[7:4] + 6 -> A[7:4] else r[7:4] -> A[7:4]
Flag Affected:	C
Description:	Adjust data in register r from hexadecimal to decimal. Result is stored back in A.

DAA	Decimal-adjust after addition
Format:	DAA r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	If r[3:0] > 9 or H = 1 r[3:0] + 6 -> r[3:0] else r[3:0] -> r[3:0] If r[7:4] > 9 or C = 1 r[7:4] + 6 -> r[7:4] else r[7:4] -> r[7:4]
Flag Affected:	C
Description:	Adjust data in register r from hexadecimal to decimal. Result is stored back in r.

DEC	Decrement
Format:	DEC A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r – 1 -> A
Flag Affected:	Z
Description:	Decrement r. Result is stored back in A.

DEC	Decrement
Format:	DEC r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r – 1 -> r
Flag Affected:	Z
Description:	Decrement r. Result is stored back in r.

DSZ	Decrement, skip if 0
Format:	DSZ A, r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	r – 1 -> A, skip if 0
Flag Affected:	None
Description:	Decrement r. Result is stored back in A. If result is 0, skip next instruction by executing a NOP.

DSZ	Decrement, skip if 0
Format:	DSZ r, r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	r – 1 -> r, skip if 0
Flag Affected:	None
Description:	Decrement r. Result is stored back in r. If result is 0, skip next instruction by executing a NOP.

HALT	Standby
Format:	HALT
Cycles:	1
Operands:	None
Operations:	0 -> WDT, 0 -> WDT prescaler 1 -> TO_, 0 -> PD_
Flag Affected:	TO_, PD_
Description:	Clear watchdog time and prescaler of the WDT. Flags TO_ and PD_ are set. Then the device is put into power-down mode and the oscillator stopped.

INC	Increment
Format:	INC A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r + 1 -> A
Flag Affected:	Z
Description:	Increment r. Result is stored back in A.

INC	Increment
Format:	INC r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r + 1 -> r
Flag Affected:	Z
Description:	Increment r. Result is stored back in r.

ISZ	Increment, skip if 0
Format:	ISZ A, r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	r + 1 -> A, skip if 0
Flag Affected:	None
Description:	Increment r. Result is stored back in A. If result is 0, skip next instruction by executing a NOP.

ISZ	Increment, skip if 0
Format:	ISZ r, r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	r + 1 -> r, skip if 0
Flag Affected:	None
Description:	Increment r. Result is stored back in r. If result is 0, skip next instruction by executing a NOP.

JMP	Unconditional branch
Format:	JMP k
Cycles:	2
Operands:	0 ≤ k ≤ 1023
Operations:	K -> PC
Flag Affected:	None
Description:	JMP is an unconditional branch. The immediate address k is loaded into PC.

JSR	Call subroutine
Format:	JSR k
Cycles:	2
Operands:	0 ≤ k ≤ 1023
Operations:	PC + 1 -> STACK[top], K -> PC.
Flag Affected:	None
Description:	Call subroutine. Return address (PC+1) is pushed onto the STACK. The immediate address k is loaded into PC.

MOV	Move between Register and A
Format:	MOV r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	A -> r
Flag Affected:	(Z)
Description:	Move A to r. If r=ACC, Z flag is updated.

MOV	Move between Register and A
Format:	MOV A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r -> A
Flag Affected:	Z
Description:	Move r to A.

MOV	Move literal to A
Format:	MOV A, #k
Cycles:	1
Operands:	0 ≤ k ≤ 255
Operations:	k -> A
Flag Affected:	None
Description:	Move 8-bit literal ‘k’ to A.

MOVC	Move the ROM code to A and TBDH
Format:	MOVC
Cycles:	2
Operands:	None
Operations:	ROM code (low byte) of {PCH[0], A} -> A ROM code (high byte) of {PCH[0], A} -> TBDH
Flag Affected:	None
Description:	The low byte of ROM code addressed by A is moved to A and the high byte is to TBDH.

NOP	No Operation
Format:	NOP
Cycles:	1
Operands:	None
Operations:	None
Flag Affected:	None
Description:	No operation.

ORL	Logic OR A with Reg
Format:	ORL A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r | A -> A
Flag Affected:	Z
Description:	Logic OR A with register r. Result is stored back in A.

ORL	Logic OR A with Reg
Format:	ORL r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r | A -> r
Flag Affected:	Z
Description:	Logic OR A with register r. Result is stored back in r.

ORL	Logic OR A with literal
Format:	ORL A, # k
Cycles:	1
Operands:	0 ≤ k ≤ 127
Operations:	k | A -> A
Flag Affected:	Z
Description:	Logic OR A with 8-bit literal ‘k’. Result is stored back in A.

PEEK	Read value from CM
Format:	PEEK r
Cycles:	1
Operands:	0 ≤ r ≤ 63
Operations:	CR[r] -> A
Flag Affected:	Z
Description:	Read value from control register and store back in A.

POKE	Write A to CM
Format:	POKE r
Cycles:	1
Operands:	0 ≤ r ≤ 63
Operations:	A -> CR[r]
Flag Affected:	(Z)
Description:	Write A to control register. If r=ACC, Z flag is updated.

RET	Return from subroutine
Format:	RET
Cycles:	2
Operands:	None
Operations:	STACK[top] -> PC
Flag Affected:	None
Description:	Return from subroutine. The top of the stack is poped and loaded into PC.

RET	Return with literal in A
Format:	RET #k
Cycles:	2
Operands:	0 ≤ k ≤ 255
Operations:	K -> A, STACK[top] -> PC
Flag Affected:	None
Description:	Return from interrupt. 8-bit literal ‘k’ is stored in A. The top of the stack is poped and loaded into PC.

RETI	Return from interrupt
Format:	RETI
Cycles:	2
Operands:	None
Operations:	1 -> GIE, STACK[top] -> PC
Flag Affected:	None
Description:	Return from interrupt. Set GIE flag. The top of the stack is poped and loaded into PC.

RLC	Rotate left through carry
Format:	RLC A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	{r[6:0], C} -> A, r[7] -> C
Flag Affected:	C
Description:	Rotate register r one bit from right to left through carry flag. The LSB of r is replaced by C flag and the MSB value is moved to C flag. Result is stored back in A.

RLC	Rotate left through carry
Format:	RLC r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	{r[6:0], C} -> r, r[7] -> C
Flag Affected:	C
Description:	Rotate register r one bit from right to left through carry flag. The LSB of r is replaced by C flag and the MSB value is moved to C flag. Result is stored back in r.

RMBn	Bit clear
Format:	RMBn r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	0 -> r[n]
Flag Affected:	None
Description:	Bit ‘n’ in register r is clear.

RRC	Rotate right through carry
Format:	RRC A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	{C, r[7:1]} -> A, r[0] -> C
Flag Affected:	C
Description:	Rotate register r one bit from left to right through carry flag. The MSB of r is replaced by C flag and the LSB value is moved to C flag. Result is stored back in A.

RRC	Rotate right through carry
Format:	RRC r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	{C, r[7:1]} -> r, r[0] -> C
Flag Affected:	C
Description:	Rotate register r one bit from left to right through carry flag. The MSB of r is replaced by C flag and the LSB value is moved to C flag. Result is stored back in r.

SBC	Subtract A from Reg and Carry
Format:	SBC A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r – A – (~C) -> A
Flag Affected:	C, H, Z
Description:	Subtract A from register r and C. Result is stored back in A.

SBC	Subtract A from Reg and Carry
Format:	SBC r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r – A – (~C) -> r
Flag Affected:	C, H, Z
Description:	Subtract A from register r and C. Result is stored back in r.

SBRn	Bit test, skip if zero
Format:	SBRn r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	Skip if r[n] = 0
Flag Affected:	None
Description:	If r[n] = 0, skip next instruction by executing a NOP.

SBSn	Bit test, skip if not zero
Format:	SBSn r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	Skip if r[n] ≠ 0
Flag Affected:	None
Description:	If r[n] ≠ 0, skip next instruction by executing a NOP.

SE	Skip if equal
Format:	SE r
Cycles:	1 (2)
Operands:	0 ≤ r ≤ 127
Operations:	Skip if A = r
Flag Affected:	None
Description:	If A = r, skip next instruction by executing a NOP.

SE	Skip if equal literal
Format:	SE #k
Cycles:	1 (2)
Operands:	0 ≤ k ≤ 255
Operations:	Skip if A = k
Flag Affected:	None
Description:	If A = k, skip next instruction by executing a NOP.

SJMP	Unconditional branch
Format:	SJMP k
Cycles:	2
Operands:	0 ≤ k ≤ 511
Operations:	K[8:0] -> PC
Flag Affected:	None
Description:	SJMP is an unconditional branch. The immediate address k[8:0] and PCH are loaded into PC.

SJSR	Call subroutine
Format:	SJSR k
Cycles:	2
Operands:	0 ≤ k ≤ 255
Operations:	PC + 1 -> STACK[top], {0, K[7:0]} -> PC.
Flag Affected:	None
Description:	Call subroutine. Return address (PC+1) is pushed onto the STACK. The immediate address k[7:0] and PCH are loaded into PC.

SMBn	Bit set
Format:	SMBn r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	1 -> r[n]
Flag Affected:	None
Description:	Bit ‘n’ in register r is set.

SUB	Subtract A from Reg
Format:	SUB A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r – A -> A
Flag Affected:	C, H, Z
Description:	Subtract A from register r. Result is stored back in A.

SUB	Subtract A from Reg
Format:	SUB r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r – A -> r
Flag Affected:	C, H, Z
Description:	Subtract A from register r. Result is stored back in r.

SUB	Subtract A from literal
Format:	SUB A, #k
Cycles:	1
Operands:	0 ≤ k ≤ 255
Operations:	k – A -> A
Flag Affected:	C, H, Z
Description:	Subtract A from 8-bit literal ‘k’. Result is stored back in A.

SWAP	Swap nibbles
Format:	SWAP A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	{r[3:0], r[7:4]] -> A
Flag Affected:	None
Description:	The upper and lower nibbles of r are exchange. Result is stored back in A.

SWAP	Swap nibbles
Format:	SWAP r, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	{r[3:0], r[7:4]] -> r
Flag Affected:	None
Description:	The upper and lower nibbles of r are exchange. Result is stored back in r.

TEST	Test Register
Format:	TEST r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r -> r
Flag Affected:	Z
Description:	Test register r. Z flag is affected.

XRL	Logic exclusive OR A with Reg
Format:	XRL A, r
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r ⊕ A -> A
Flag Affected:	Z
Description:	Logic exclusive OR A with register r. Result is stored back in A.

XRL	Logic exclusive OR A with literal
Format:	XRL A, #k
Cycles:	1
Operands:	0 ≤ k ≤ 255
Operations:	k ⊕ A -> A
Flag Affected:	Z
Description:	Logic exclusive OR A with 8-bit literal ‘k’. Result is stored back in A.

XRL	Logic exclusive OR A with Reg
Format:	XRL r, A
Cycles:	1
Operands:	0 ≤ r ≤ 127
Operations:	r ⊕ A -> r
Flag Affected:	Z
Description:	Logic exclusive OR A with register r. Result is stored back in r.

11. Absolute Maximum Ratings

Supply Voltage …….. -0.3V~3.6V
Input Voltage …… V_{SS}-0.3V~V_{DD}+0.3V
Storage temperature ……..… -50℃~125℃
Operation temperature ……..…-25℃~75℃

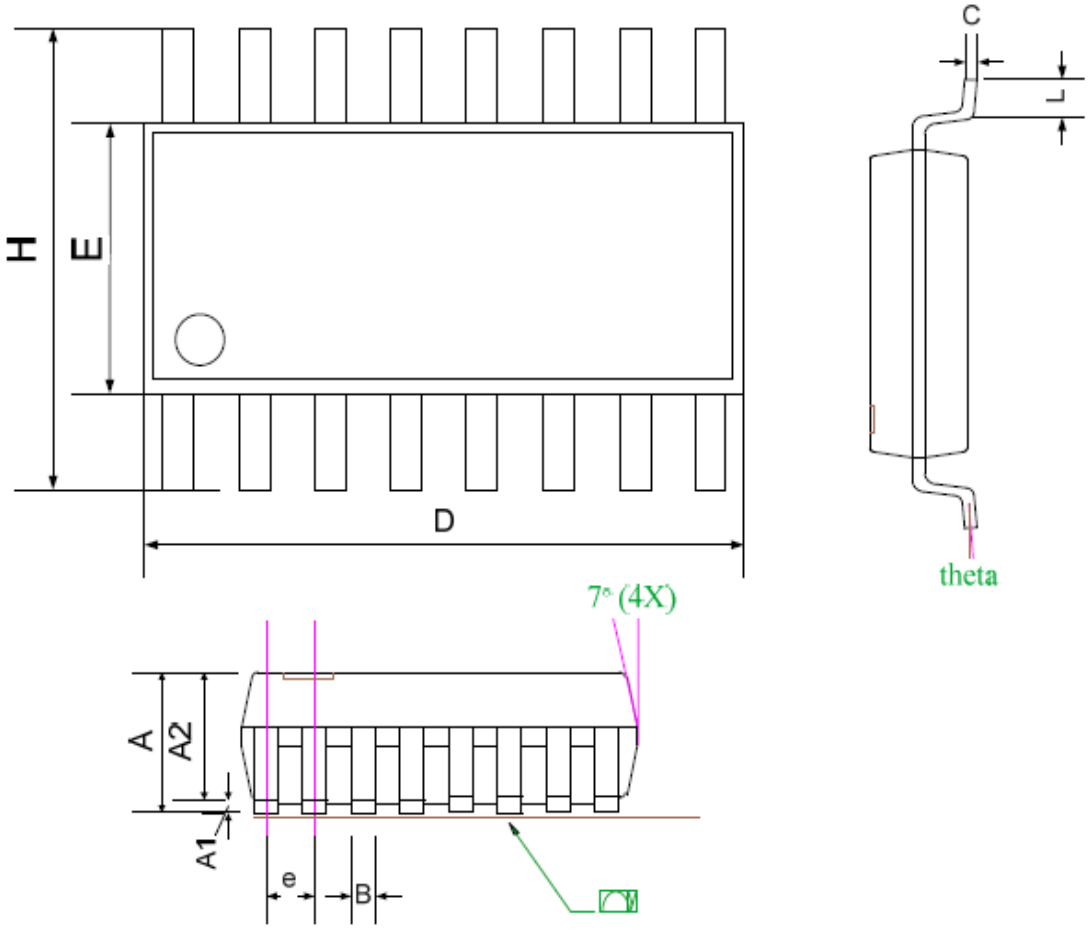
11.1. DC Characteristics

TA=25℃

Symbol	Parameter	V_{DD}	Conditions	Min.	Typ.	Max.	Unit
V_{DD1}	Operating Voltage	-	BR4M	1.8	-	3.6	V
V_{DD2}	Operating Voltage	-	BR16M	2.2	-	3.6	V
I_{STB}	Standby Current		System halt, WDT disable				
		3V		-	-	1	uA
I_{OH}	I/O Port Source Current (PA, PB, PX, PY)						
		3V	V_{OH}=2.7V		-4		mA
I_{OL1}	I/O Port Sink Current (PA, PB, PY)						
		3V	V_{OL}=0.3V		10		mA
I_{OL2}	I/O Port Sink Current (PX)	3V	V_{OL}=1.5V		320		mA
R_{PH1}	PA, PB Pull-high R		-				
		3V	-		150		KΩ
R_{PH2}	PA7/ RESETB Pull-high R		-				
		3V	-		80		KΩ
V_{IL1}	Input Low Voltage for Input Port	-	-	0	-	$0.3V_{DD}$	V
V_{IH1}	Input High Voltage for Input Port	-	-	$0.7V_{DD}$	-	V_{DD}	V
V_{IL2}	Input Low Voltage for RESETB	-	-	0	-	$0.3V_{DD}$	V
V_{IH2}	Input High Voltage for RESETB	-	-	$0.7V_{DD}$	-	V_{DD}	V

12. Package Information

16-Lead Plastic Small Outline (SOP) — 150 mil



SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.60	1.75	0.053	0.063	0.069
A1	0.10	---	0.25	0.004	---	0.010
A2	---	1.45	---	---	0.057	---
B	0.33	---	0.51	0.013	---	0.020
C	0.19	---	0.25	0.007	---	0.010
D	9.80	---	10.00	0.386	---	0.394
E	3.80	---	4.00	0.150	---	0.157
e	---	1.27	---	---	0.050	---
H	5.80	---	6.20	0.228	---	0.244
L	0.40	---	1.27	0.016	---	0.050
y	---	---	0.10	---	---	0.004
theta	0°	---	8°	0°	---	8°