MiDAS1.1 Family:
EPROM/ROM
8-bit Turbo Microcontrollers

Rev. 1.6

September 2005

- *CoreRiver Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice.*
- *Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.*
- *The CoreRiver Semiconductor products listed in this document are intended for usage in general electronics applications. These CoreRiver Semiconductor products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury.*

Table of Contents

List of Figures

List of Tables

1 Product Overviews

CoreRiver's MiDAS1.1 family is a group of fast 80C52 compatible microcontrollers without wasted clock and memory cycles. Four clocks per one machine cycle are consumed in a redesigned processor core. As a result, every 8052 instruction is executed about 3 times faster than that of traditional 80C52.

The MiDAS1.1 family offers three timer/counters, a serial port, up to 18 I/O ports, eleven-channel 10-bit ADC (Analog to Digital Converter), one-channel 8-bit PWM (Pulse Width Modulator), a Watchdog timer, and LVD (Low Voltage Detector) as peripherals.

The MiDAS1.1 family provides power saving modes to reduce power consumption in the power-critical applications and noise tolerant scheme.

To support an easy target system development, user-friendly MDS (MCU Development System) called as GENSYS and easy-to-use training system is also provided.

2 Features

- 8-bit Turbo 80C52 architecture
- 4 cycles/1 machine cycle
- Instruction level compatible with Intel 80C52
- 0/4Kbyte Mask ROM
- 0/4Kbyte Erasable Programmable ROM (EPROM)
- 128byte RAM
- Extended 2.4V to 5.5V supply voltage
- Operating Frequency
  - ✓ Max. 20MHz @ 4.5 ~ 5.5V
  - ✓ Max. 10MHz @ 2.4 ~ 3.3V
- -20 °C to 85 °C operating temperature
- Fully Programmable 18 I/O pins
  - ✓ Pull-up controlled by S/W
  - ✓ Open-drain Output
  - ✓ Push-pull Output
- Low Voltage Detector (LVD)
- 16-bit Programmable Watchdog Timer (WDT)
- Two 16-bit Timer/Counters
- Full-Duplex UART
  - ✓ Automatic address recognition
- 1-channel 8-bit high speed Pulse Width Modulator (PWM)
- 12-channel 10-bit Analog to Digital Converter (ADC)
  - ✓ Max. 100K SPS (samples per second) @ 8MHz
  - ✓ Programmable input clock frequency
- 10 interrupt sources including 4 external
  - ✓ Timer 0/1, UART, ADC, PWM, WDT, and four external
  - ✓ Two-level interrupt priority
- Reset scheme
  - ✓ On-chip power-on-reset (POR)
  - ✓ External reset
  - ✓ Low voltage detector reset
  - ✓ Optional Watchdog timer reset

- Power consumption
  - ✓ Active current: Max 10mA @ 5V, 20MHz
  - ✓ Stop current: Max 1uA
- ESD protection up to 2,000V
- Latch-up protection up to ±200mA
- Package : 8/16/20-SPDIP, 8/16/20-SOIC, and 16-TSSOP

3 Block Diagram

Figure 3-1 shows the functional block diagram of MiDAS1.1 Family. The CPU fetches instructions from the program memory (ROM or EPROM) and executes them. Data are processed and read from or written to data memory (RAM) or integrated peripherals via special function registers(SFRs). For data processing inside the CPU, the arithmetic and logic unit (ALU) is used.

Data processing as well as reading and writing is supported by a set of registers. Except for the program counter (PC) and the four general-purpose register banks, these registers are mapped to the special function register area. The program counter resides inside the CPU, and the four register banks are mapped to the internal RAM.

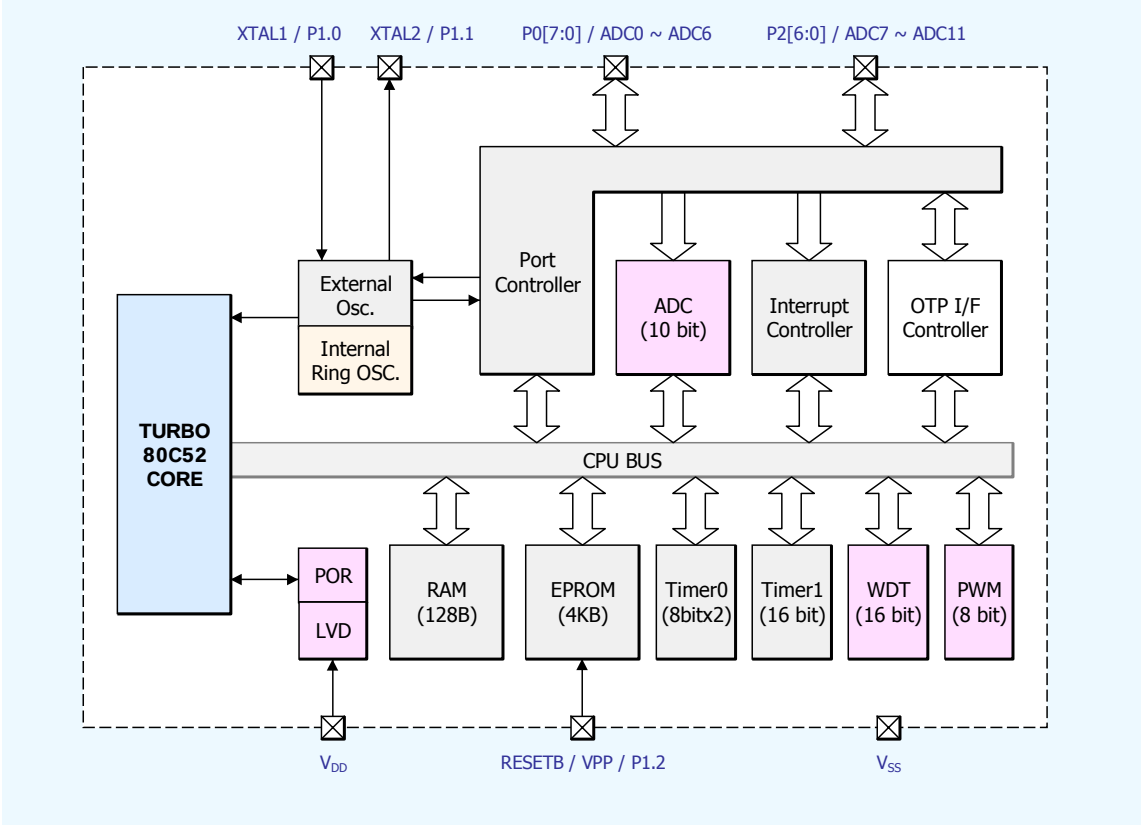


Figure 3-1 Overall Block Diagram

4 Pin Configurations

The MiDAS1.1 family supports a variety of package, e.g. 8/16/20-SPDIP, 8/16/20-SOIC and 16-TSSOP. The detailed pin configuration is shown in Figure 4-1.

Figure 4-1

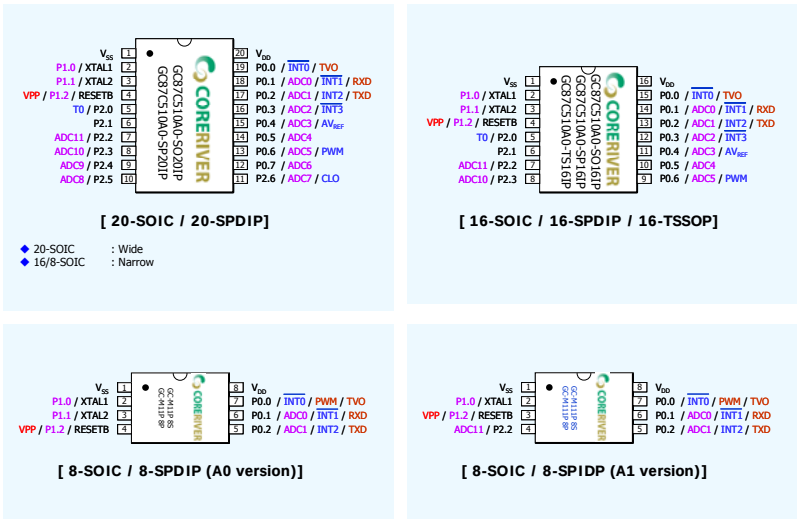


Figure 4-1 Pin Configuration

5 Pin Description

Table 5-1 Pin Description

Symbol	Direction	Description	Share Pins
V_{DD}	Input	Voltage power source	-
V_{SS}	Input	Voltage power ground	-
RESETB / VPP / P1.2	Input/Output	• External reset input signal (Default) • Bit Programmable - Open-drain output	VPP (11.5V)
XTAL1 / P1.0	Input/Output	• Crystal input/output (Default) • Bit Programmable with Schmitt trigger - Optional Pull-up control Enable - Open-drain output - Push-pull output	Crystal input
XTAL2 / P1.1			Crystal output (only A0 version)
P0[7:0]	Input/Output	• Bit Programmable with Schmitt trigger - Optional Pull-up control Enable - Open-drain output - Push-pull output (Default)	RX, TX, TVO, INT0B ~ INT3, ADC0 ~ ADC6, PWM
P2[6:0]	Input/Output	• Bit Programmable with Schmitt trigger - Optional Pull-up control Enable - Open-drain output - Push-pull output (default)	ADC7 ~ ADC11 CLO AV_{REF}

6 Functional Description

6.1 CPU Description

6.1.1 Memory Organization

MiDAS1.1 family has separate address spaces for program and data memory. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by an 8-bit CPU.

Program Memory can only be read, not written to. There can be up to 4K bytes of Program Memory.

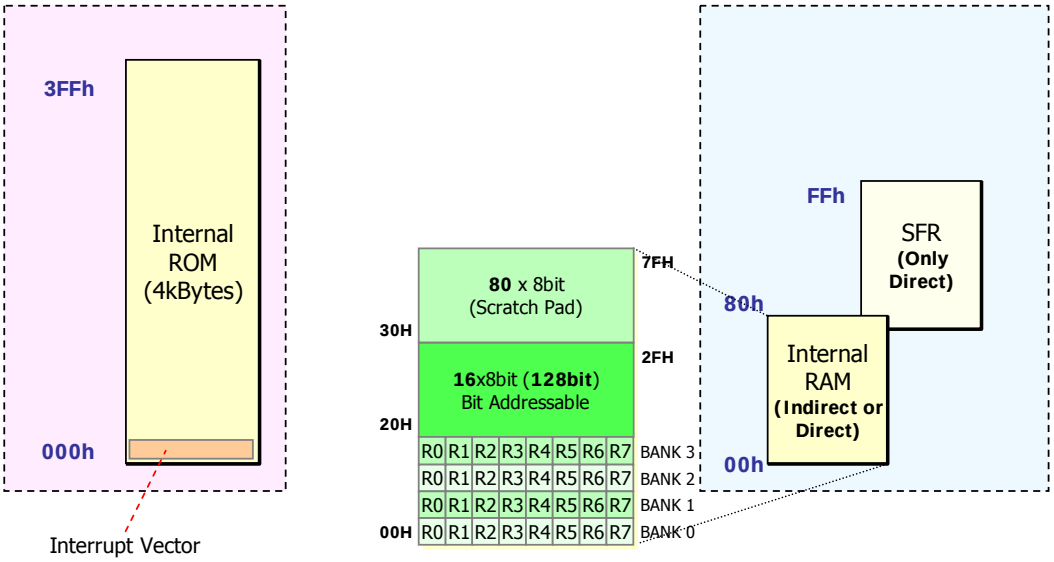


Figure 6-1 Memory Organization

6.1.1.1 Program Memory

The left part of Figure 6-1 shows a map of the Program Memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 6-1, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0 (INT0B), for example, is assigned to location 0003h. If External Interrupt 0 is going to be used, its service routine must begin at location 0003h. If the interrupt is not going to be used, its service

location is available as general purpose Program Memory.

The interrupt service locations are spaced at 8-byte intervals: 0003h for External Interrupt 0, 000Bh for Timer 0, 0013h for External Interrupt 1, 001Bh for Timer 1 and etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

Program Memory addresses are always 16bits wide, even though the actual amount of Program Memory used may be less than 4K bytes.

Program Memory can only be read, not written to. There can be up to 4K bytes of Program Memory.

6.1.1.2 Data Memory

The internal data memory address space is divided into three basic, physical separate and distinct blocks as shown in the right part of Figure 6-1:

- The lower 128byte of internal data RAM located to the address range 00h – 7Fh.
- The 128byte area for special function register (SFR) located to the address range 80h – FFh.

The special function registers (SFRs) are accessible only by direct addressing mode (address is part of the instruction).

6.1.1.2.1 Bit-addressable Memory Locations

In the lower internal data RAM, a bit-addressable area of 128 freely programmable, directly addressable bits are located at byte addresses 20h – 2Fh. Bit 0 of the data byte at 20h has bit address 00H, bit 7 of the data byte at 2Fh has bit address 7Fh.

In the special function register area, all SFRs, which are located at addresses with address bit 0-2 equal 0 (addresses 80h, 88h, 90h, …, F0h, F8h), are bit-addressable SFRs.

6.1.1.2.2 Register Banks

The lower 32 locations of the lower internal data RAM are assigned to four banks with eight general purpose registers (GPRs) each. Only one of these banks can be enabled at a time to be used as general purpose registers.

6.1.2 Special Function Registers (SFRs) Map and Description

The MiDAS1.1 family uses Special Function Registers (SFRs) to control and monitor peripherals and their modes.

The SFRs are located at 80h-FFh and are accessed by direct addressing only. Some of the SFRs are bit addressable. This is very useful in cases where one wishes to modify a particular bit without changing the others. The SFRs that are bit addressable are those whose addresses end in 0h or 8h. The list of SFRs is as follows. Refer to Appendix B for more details about SFRs.

Table 6-1 SFR map (* = bit addressable SFR)

F8h	* **EIP**								FFh
F0h	* B				**P0DIR**	**P1DIR**	**P2DIR**		F7h
E8h	* **EIE**						**ADCR**	**ADCON**	EFh
E0h	* ACC	**ADCSELH**	**ADCSEL**	**ALTSEL**	**P0SEL**	**P1SEL**	**P2SEL**		E7h
D8h	* **WDCON**				**PWMCON**		**PWMD**		DFh
D0h	* PSW				**P0TYPE**	**P1TYPE**	**P2TYPE**		D7h
C8h									CFh
C0h					**PMR**	**STATUS**			C7h
B8h	* IP						**OSCICN**		BFh
B0h									B7h
A8h	* IE								AFh
A0h	* P2								A7h
98h	* SCON	SBUF							9Fh
90h	* P1	**EXIF**							97h
88h	* TCON	TMOD	TL0	TL1	TH0	TH1			8Fh
80h	* P0	SP	DPL	DPH				PCON	87h

Note: Newly added SFRs at MiDAS1.1 family are in **bold style**.

Table 6-2 Summary of SFRs (* : undefined value)

Register Name	Symbol	Address	Reset Value	Bit Addressable
Port 0	P0	80h	11111111	Yes
Stack Pointer	SP	81h	00000111	No
Data Pointer Low Byte	DPL	82h	00000000	No
Data Pointer High Byte	DPH	83h	00000000	No
Power Control	PCON	87h	0**10000	No
Timer/Counter 0/1 Control	TCON	88h	00000000	Yes
Timer/Counter 0 Mode Control	TMOD	89h	****0000	No
Timer/Counter 0 Low Byte	TL0	8Ah	00000000	No
Timer/Counter 1 Low Byte	TL1	8Bh	00000000	No
Timer/Counter 0 High Byte	TH0	8Ch	00000000	No
Timer/Counter 1 High Byte	TH1	8Dh	00000000	No
Port 1	P1	90h	*****111	Yes
Added External Interrupt & LVD Control	**EXIF**	91h	**000101	No
Serial Control	SCON	98h	***0**00	No
Serial Buffer	SBUF	99h	00000000	No
Port 2	P2	A0h	*1111111	Yes
Interrupt Enable control	IE	A8h	00*00000	Yes
Interrupt Priority	IP	B8h	10*00000	Yes
Internal Ring Oscillator Control	**OSCICN**	BEh	*****100	No
Power Management Control	**PMR**	C4h	****0***	No
Crystal Status	**STATUS**	C5h	***0****	No
Program Status Word	PSW	D0h	00000000	Yes
Port 0 Type Control	**P0TYPE**	D4h	11111111	No
Port 1 Type Control	**P1TYPE**	D5h	*****111	No
Port 2 Type Control	**P2TYPE**	D6h	*1111111	No
Watchdog Timer Control	**WDCON**	D8h	11010000	Yes
PWM Control	**PWMCON**	DCh	0000*000	No
PWM Duty Data	**PWM0D**	DEh	00000000	No
Accumulator	ACC	E0h	00000000	Yes
ADC Channel Input Enable High	**ADCSELH**	E1h	11111111	No
ADC Channel Input Enable Low & Mux Selection	**ADCSEL**	E2h	11111111	No
Alternative function Selection	**ALTSEL**	E3h	000000**	No
Port 0 Pull-up Control	**P0SEL**	E4h	00000000	No
Port 1 Pull-up Control	**P1SEL**	E5h	******11	No
Port 2 Pull-up Control	**P2SEL**	E6h	*0000000	No
Extended Interrupt Enable	**EIE**	E8h	**00**00	Yes

ADC Result High	**ADCR**	EEh	00000000	No
ADC Control & ADC Result Low	**ADCON**	EFh	00100000	No
B Register	B	F0h	00000000	Yes
Port 0 Input/Output Control	**P0DIR**	F4h	11111111	No
Port 1 Input/Output Control	**P1DIR**	F5h	*****111	No
Port 2 Input/Output Control	**P2DIR**	F6h	*1111111	No
Extended Interrupt Priority	**EIP**	F8h	**00**00	Yes

Caution: Don't touch * bits. Updating these bits will cause the malfunctions. Especially following bits of SFR should not be updated.

6.1.3 Instruction Set Summary

The instruction set of the MiDAS1.1 family is compatible with that of the well-known 80C52.

The MiDAS1.1 family has a special internal architecture and different timing to the standard 80C52. This means the relative duration of the individual instructions, i.e. the number of clock cycles per instructions, is different to that of 80C52.

Refer to **Appendix A** for more details about instruction set.

Table 6-3 Summary of Instruction Set

Type	Instruction	Description
Arithmetic	ADD	Addition
	ADDC	Addition with Carry
	SUBB	Subtraction with Borrow
	INC	Increment
	DEC	Decrement
	MUL	Multiply
	DIV	Divide
	DA	Decimal Adjust
Logical	ANL	Logical AND
	ORL	Logical OR
	XRL	Logical Exclusive OR
	CLR	Clear
	CPL	Complement
	RL	Rotate Left
	RLC	Rotate Left with Carry
	RR	Rotate Right

	RRC	Rotate Right with Carry
	SWAP	Swap Nibbles
Data Transfer	MOV	Move Data
	MOVC	Move Code
	PUSH	Push
	POP	Pop
	XCH	Exchange
	XCHD	Exchange Lo-digit
Boolean	CLR	Clear Bit
	SETB	Set Bit
	CPL	Complement Bt
	ANL	AND Bi
	ORL	OR Bit
	MOV	Move Bit
	JC	Jump if Carry is set
	JNC	Jump if Carry is not set
	JB	Jump if Bit is set
	JNB	Jump if Bit is not set
	JBC	Jump if Bit is set & Clear
Branch	ACALL	Absolute Call
	LCALL	Long Call
	RET	Return from Subroutine
	RETI	Return from Interrupt
	AJMP	Absolute Jump
	LJMP	Long Jump
	SJMP	Short Jump
	JMP	Jump with DPTR
	JZ	Jump if ACC is zero
	JNZ	Jump if ACC is not zero
	CJNE	Compare and Jump if not equal
	DJNZ	Decrement and Jump if not zero
	NOP	No Operation

6.1.3.1 Addressing Modes

The operands used in the instructions can be addressed in different modes. The MiDAS1.1 family uses six different addressing modes to this end:

- Immediate
- Direct
- Indirect
- Index
- Register
- Register-specific

6.1.3.1.1 Immediate Addressing Mode

Immediate addressing mode means that constants can be loaded to registers. The constant is part of the instruction in program memory.

Examples:

`MOV R3, #0FFh`	R3 is loaded with constant value 0FFh
`ORL PSW, #00000101b`	Logical OR operation with PSW and 0000 0101b
`MOV DPTR, #1243h`	Load data pointer with constant value 1234h

6.1.3.1.2 Direct Addressing Mode

In direct addressing mode, the operand is specified by an 8-bit address field, which is part of the instruction.

Note: The SFRs can be accessed only in this mode. And the lower 128 bytes of the RAM can be addressed in this mode, too.

Examples:

`MOV A, TCON`	TCON is the 8-bit address of the SFR TCON.
`MOV R4, variable`	Variable is the 8-bit address of a memory location in the lower part of the RAM.

6.1.3.1.3 Indirect Addressing Mode

In indirect addressing mode, the operand is specified by an 8-bit that is stored in a register. And the content of either R0 or R1 (in the selected register bank) is used.

Note: The lower 128 bytes of the RAM can be addressed by using 8-bit addresses.

Examples:

`MOV A, @R0`	RAM is addressed by contents of R0 (8-bit address)

6.1.3.1.4 Index Addressing Mode

Only program memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in program memory. A 16-bit base register (either DPTR or the PC) points to the base of the table, and the accumulator is set up with the table entry number. The address of the table entry in program memory is formed by adding the accumulator data to the base pointer. Another type of indexed addressing is used in the “case jump” instruction. In this case, the destination address of a jump instruction is calculated as the sum of the base pointer and the accumulator data.

Examples:

`MOVC A, @A + DPTR`	Address is the sum of DPTR and accumulator.
`MOVC A, @A + PC`	Address is the sum of PC and accumulator
`JMP @A + DPTR`	Jump to sum of accumulator and DPTR

6.1.3.1.5 Register Addressing Mode

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits (RS0 and RS1) in the PSW.

6.1.3.1.6 Register-Specific Addressing Mode

Some instructions are specific to a certain register. For example, some instructions always operate on the accumulator, or data pointer, etc., so no address byte is needed to point to it.

6.1.4 CPU Timing

The CPU timing for the MiDAS1.1 family is an important aspect, especially for those who wish to use software instructions to generate timing delays. Also, it provided the user with an insight into the timing differences between the MiDAS1.1 family and the standard 80C52 as shown in Figure 6-2.

In the MiDAS1.1 family, each machine cycle is four clock periods long. Each clock period is designated a state. Thus each machine cycle is made up of four states, S1, S2, S3 and S4 in that order. Due to the reduced time for each instruction execution, both the clock edges are used for internal timing. Hence it is important that the duty cycle of the clock be as close to 50% as possible to avoid timing conflicts. Since the MiDAS1.1 family fetches one opcode per machine cycle, in most of the instructions, the number of machine cycles needed to execute the instruction is equal to the number of bytes in the instruction.

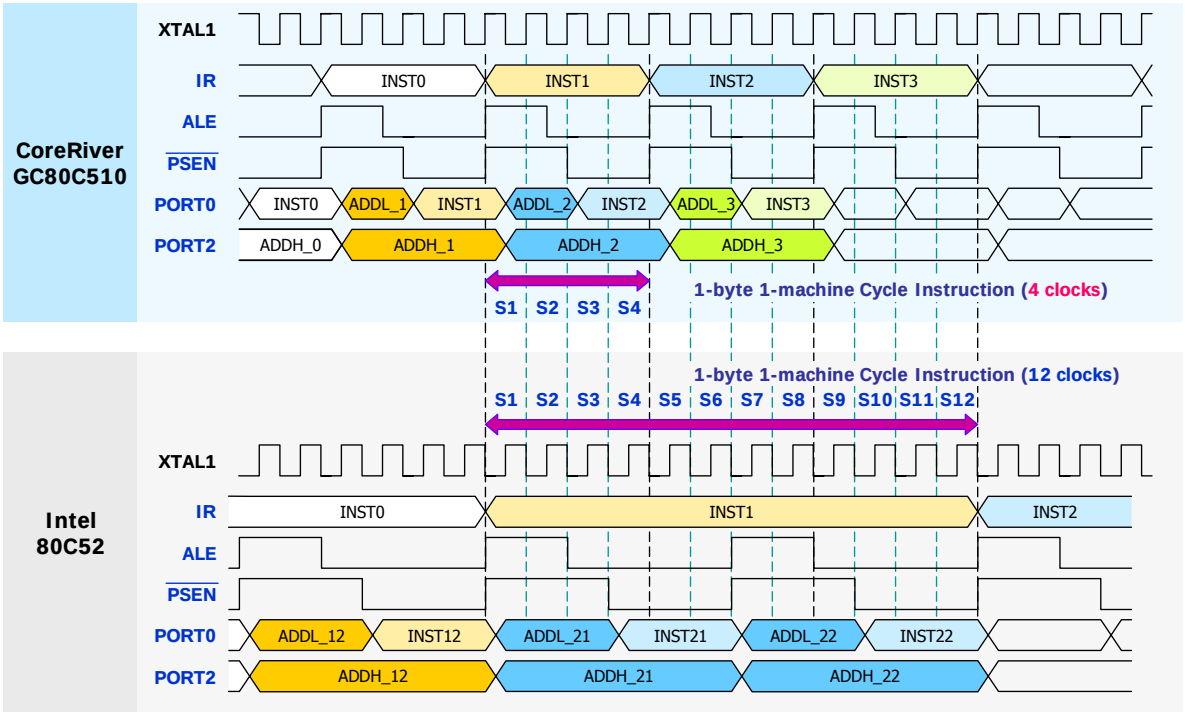


Figure 6-2 Comparative Timing of the MiDAS1.1 family and Intel 80C52

6.2 Peripheral Description

6.2.1 I/O Ports

The MiDAS1.1 Family has the port 0 (8 bits), port 1 (3 bits), and port 2 (7 bits). The port configuration is to control the registers: input/output direction control (PXDIR), output type control to determine the push-pull or open-drain output type (PXTYPE), and pull-up control (PXSEL).

The ADC (Analog-to-Digital Converter) is 12 channels and 10-bit resolution. The port 0 has 7-channel ADC input and port 2 has 5-channel ADC input.

The ADC configuration for port is to control the register: ADCSEL, ADCSELH, ADCON, and ADCR.

After reset, the port 0 and port 2 are configured as general I/O ports which is input mode with push-pull output type and pull-up register is ON. And ADC input channels of port 0 and port 2 are disabled. The port 1 is configured as crystal input/output and reset signal input.

6.2.1.1 PORT 0

Port 0 has the following configuration modes in according to P0DIR, P0SEL, P0TYPE, ALTSEL, ADCSEL, ADCON, and ADCSELH register.

- Input / Output mode controlled by P0DIR (Default = FFh, Input mode)
- Push-pull / Open-drain output type controlled by P0TYPE (Default = 00h, Push-pull output type)
- Pull-up ON / OFF controlled by P0SEL (Default = FFh, Pull-up ON)
- ADC input enable & digital input disable, or ADC input disable & digital input enable by ADCSEL and ADCSELH (Default = FFh is ADC input disable & digital input enable.)
- AV_{REF} (ADC reference voltage) using P0[4] enabled by ADCON

After reset, port 0 is configured as input mode with push-pull output type and pull-up is turn-on. Also, ADC input channels are disabled. Note that the port 0[7:1] (= P0[7:1]) is configurable as an ADC input pins. The ADC0 ~ ADC6 input channels of the port 0 are selected by ADCSEL and ADCSELH register.

[How to control the ADC input channel enable / disable for P0]

If an ADC input of port 0 is enabled, port 0 is automatically configured as ADC input channel without any effect of port 0 state. Namely, ADC input channel is valid by ADC selection registers (ADCSEL, ADCSELH) even if port 0 is operating on the output mode (bit value of P0DIR = 0). If ADC input channel is disabled, port configuration will be back to the state before entering the ADC input channel enable mode. To select the ADC channel input, set the bit of the ADCSEL or ADCSELH register to “0” (0 = ADC channel bit enable, 1 = disable (Default)). In more detail about ADC, refer to “**6.2.7 ADC**” part.

[How to control the P0 input / output mode]

The initial value of P0DIR (Port 0 Input/Output Control) register is FFh. This means that the mode of P0 is input mode. To configure the bit of the P0 to output mode, set to the bit of the P0DIR register to “0” (0 = Output, 1 = Input (Default)).

[How to control the P0 pull-up ON/OFF]

The initial value of P0SEL (Port 0 Pull-up Control) register is 00h. This means that the pull-up of P0 is ON. To disable the pull-up resistor control, set to the bit of the P0SEL register to “1” (0 = Pull-up ON (Default), 1=Pull-up OFF).

[How to control the P0 output type]

The initial value of P0TYPE (Port 0 Type Control) register is 00h. This means that the output type of P0 is push-pull output type. To configure the bit output type of the P0 to open-drain output, set the bit of the P0TYPE register to “1” (0 = Push-pull output (Default), 1 = Open-drain output).

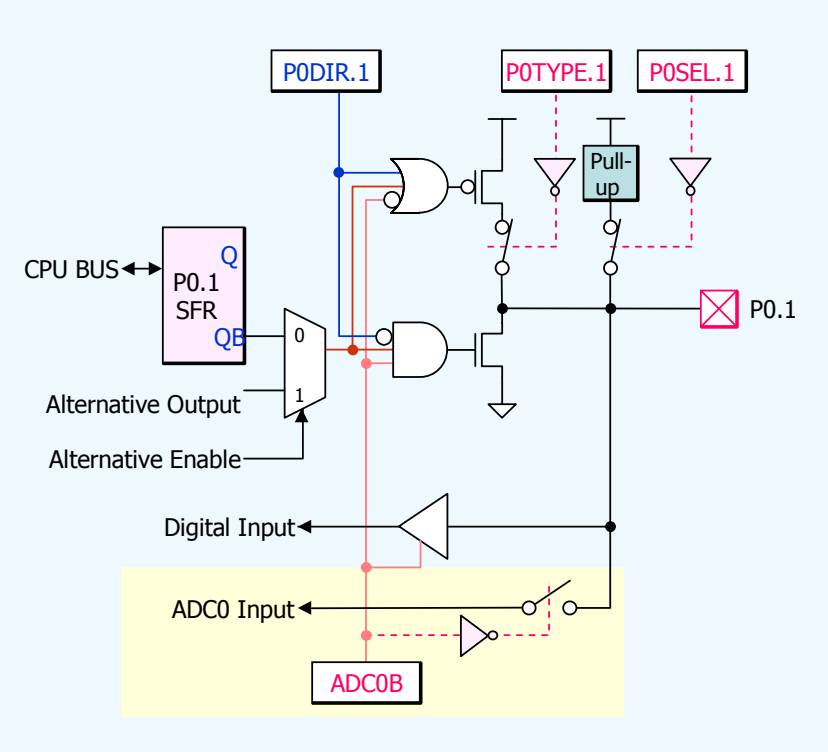


Figure 6-3 Configuration of PORT 0

Note: Read-Modify-Write instructions do not read port pin but SFR register.

[About the alternative pins and description]

In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs.

Port 0	P0[7]	P0[6]	P0[6]	P0[4]	P0[3]	P0[2]	P0[1]	P0[0]
	ADC6	ADC5 PWM	ADC4	ADC3 AV_{REF}	ADC2 INT3B	ADC1 INT2 TXD	ADC0 INT1B RXD	INT0B TVO PWM

Alternative Pin	Description
ADC6 ~ ADC0	ADC Analog Input 6 ~ 0
AV_{REF}	ADC Voltage Reference Input.
PWM	PWM Waveform Output. (Refer to ALTSEL register)
INT3B	External Interrupt 3. A falling edge on this pin will cause an external interrupt 3 if enabled.
INT2	External Interrupt 2. A rising edge on this pin will cause an external interrupt 2 if enabled.
INT1B	External Interrupt 1. A falling edge or low level on this pin will cause an external interrupt 1 if enabled.
INT0B	External Interrupt 0. A falling edge or low level on this pin will cause an external interrupt 0 if enabled.
TXD	Serial Port Transmit. This pin transmits the serial port data in serial port modes 1 and emits the synchronizing clock in serial port mode 0. (Refer to ALTSEL register)
RXD	Serial Port Receive. This pin receives the serial port data in serial port modes 1 and is a bi-directional data transfer pin in serial port mode 0.
TVO	Timer 0 Overflow Output. (Refer to ALTSEL register)

[How to control the TVO enable / disable]

If user wants to use the P0[0] bit is configured as TVO (Timer 0 overflow clock output to P0[0]), set the ALTSEL[3] to “1”. Then TVO waveform will be output to P0[0].

[How to control the PWM waveform output to P0[0] or P0[6] enable / disable]

If user wants to use the P0[0] bit as PWM waveform output to P0[0], set the PWM00 (ALTSEL[4]) to “1”. Then PWM waveform will be output to P0[0].

If user wants to use the P0[6] bit as PWM waveform output to P0[6], set the P0SEL (PWMCON[0]) to "1". The PWM waveform will be output to P0[6].

6.2.1.1.1 **P0** (80h) : Port 0 Register

Bit No.	7	6	5	4	3	2	1	0
	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register functions as a general purpose I/O port. The initial value of P0 is FFh.

6.2.1.1.2 **P0TYPE** (D4h) : Port 0 Type Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0TYPE.7	P0TYPE.6	P0TYPE.5	P0TYPE.4	P0TYPE.3	P0TYPE.2	P0TYPE.1	P0TYPE.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register configures the P0 output type to the push-pull or open-drain. The initial P0 output type is configured to push-pull output type.

0 = Push-pull output type. (Default)

1 = Open-drain output.

6.2.1.1.3 **P0SEL** (E4h) : Port 0 Pull-up Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0SEL.7	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register is enables/disabled the internal pull-up in P0.

0 = Pull-up ON. The bit is configured to pull-up ON. (Default)

1 = Pull-up OFF. The bit is configured to pull-up OFF.

6.2.1.1.4 **P0DIR** (F4h) : Port 0 Input/Output Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0DIR.7	P0DIR.6	P0DIR.5	P0DIR.4	P0DIR.3	P0DIR.2	P0DIR.1	P0DIR.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register is configured the P0 to input mode or output mode. The initial P0 mode is configured to input mode.

0 = Output mode. The bit is configured to output.

1 = Input mode. The bit is configured to input. (Default)

6.2.1.1.5 **ALTSEL** (E3h) : Alternative Function Selection Register

Bit No.	7	6	5	4	3	2	1	0
	IOXEN	IORSTEN	CLO	PWM00	TVO	TX	-	-
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)		

Symbol	Description
PWM00	PWM Waveform Output Enable to P0[0]. 0 = Disable the PWM waveform output to P0[0]. (Default) 1 = Enable the PWM waveform output to P0[0].
TVO	Timer0 Overflow Output Enable to P0[0]. 0 = Disable the Timer0 overflow output to P0[0]. (Default) 1 = Enable the Timer0 overflow output to P0[0].
TX	UART TX Data Output Enable to P0[2]. (TXD : TX Data) 0 = Disable the TX data output to P0[2]. (Default) 1 = Enable the TX data output to P0[2]. Note that the port directional mode (input/output) of P0[2] for TXD is don't care. But the directional mode (input/output) of P0[1] for RXD must set up to input mode, When user is setting to UART communication.
-	Reserved

6.2.1.1.6 **ADCSELH** (E1h) ADC Channel Selection High Register

Bit No.	7	6	5	4	3	2	1	0
	ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

Symbol	Description	
ADC4B ADC5B ADC6B	ADC Channel Selection Low. These bits are configured to enable/disable the ADC channel inputs (ADC[11:4]). P0[5] ~ P0[7] : ADC4 ~ ADC6 input.	
	ADC Channel	Result
	ADC4B	1 = ADC4 input channel is disabled & digital input is enabled at P0[5]. (Default) 0 = ADC4 input channel is enabled & digital input is disabled.
	ADC5B	1 = ADC5 input channel is disabled & digital input is enabled at P0[6]. (Default) 0 = ADC5 input channel is enabled & digital input is disabled.
	ADC6B	1 = ADC6 input channel is disabled & digital input is enabled at P0[7]. (Default) 0 = ADC6 input channel is enabled & digital input is disabled.

6.2.1.1.7 **ADCSEL** (E2h): ADC Channel Selection Low & MUX Selection Register

Bit No.	7	6	5	4	3	2	1	0
	ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

* R/W (1) means that user can read and writer this flag bit. The initial value of flag bit is “1”.

<table>
<tr><th>Symbol</th><th colspan="6">Description</th></tr>
<tr><td rowspan="5">ADC0B
ADC1B
ADC2B
ADC3B</td><td colspan="6">ADC Channel Selection Low.
These bits are configured to enable/disable the ADC channel inputs (ADC[3:0]). P0[1] ~ P0[4] : ADC0 ~ ADC3 input. (Default = 1)</td></tr>
<tr><td>ADC Channel</td><td colspan="5">Result</td></tr>
<tr><td>ADC0B</td><td colspan="5">1 = ADC0 input channel is disabled & digital input is enabled at P0[1].
0 = ADC0 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC1B</td><td colspan="5">1 = ADC1 input channel is disabled & digital input is enabled at P0[2].
0 = ADC1 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC2B</td><td colspan="5">1 = ADC2 input channel is disabled & digital input is enabled at P0[3].
0 = ADC2 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC3B</td><td colspan="5">1 = ADC3 input channel is disabled & digital input is enabled at P0[4].
0 = ADC3 input channel is enabled & digital input is disabled.</td></tr>
<tr><td rowspan="11">CH0
CH1
CH2
CH3</td><td colspan="6">ADC MUX Selection.
Theses bits is configured to select the ADC input. The initial value of CH[3:0] is 0xFh (4’b1111). Ch, Dh, Eh and Fh means that all ADC inputs are disabled.</td></tr>
<tr><td>CH3</td><td>CH2</td><td>CH1</td><td>CH0</td><td colspan="2">Result</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">0x0h : ADC0 is selected</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td colspan="2">0x1h : ADC1 is selected</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="2">0x2h : ADC2 is selected</td></tr>
<tr><td colspan="4">…</td><td colspan="2"></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td colspan="2">0xAh : ADC10s selected</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td colspan="2">0xBh : ADC11is selected</td></tr>
<tr><td colspan="4">1100 (0xCh) ~ 1111 (0xFh)</td><td colspan="2">No ADC channel is not selected (Default Value = 0xFh)</td></tr>
</table>

6.2.1.2 PORT 1

Port 1 has the following configuration modes in according to P1DIR, P1SEL, P1TYPE, ALTSEL, and PMR register.

- Input / Output mode controlled by P1DIR (Default = ******11, Input mode)
- General I/O enable by ALTSEL[7] (IOXEN, Default (= 0) is not general I/O for P1[0] and P1[1]), ALTSEL[6] (IORSTEN, Default (=0) is not general I/O for P1[2]), and PMR[3] (XTOFF, Default (=0) does not disable the internal amplifier for external crystal).
- Push-pull / Open-drain output type of P1[0] and P1[1] controlled by P1TYPE (Default = ******00, Push-pull output type). The P1[2] is only the open-drain output type.
- Pull-up ON / OFF controlled by P1SEL (Default = ******00, Pull-up OFF)

After reset, the port 1 configuration is that P1[0] and P1[1] is crystal input/output (XTAL1/XTAL2) and P1[2] is a chip reset signal input (RESETB). The pull-up of P[0] and P[1] is OFF and P[2] has no pull-up. In addition, Port 1 can be configured as the following modes using P1DIR, P1SEL, P1TYPE, ALTSEL register.

[How to control the XTAL1 (P1[0]) and XTAL2 (P1[1]) as general I/O pins]

If you want to use XTAL1 (P1[0]) and XTAL2 (P1[1]) as general I/O pins, you should set ALTSEL[7] (IOXEN flag) to “1”. Then, P1[0] and P1[1] are configured as general I/O pins. And you should set PMR[3] (XTOFF flag) to “1”.

In other words, to use the general I/O pins, you should set P1[0] and P1[1] as general I/O pins and disable the internal amplifier for crystal between these two pins. Refer to the below figure.

[How to control the RESETB (P1[2]) as I/O pin]

If you want to use RESETB as general I/O pin, you should set ALTSEL[6] (IORSTEN flag) to “1”. Then, P1[2] is configured as general I/O pin. Note that retain “1” at P1[2] pin until this RESETB pin is configured as general I/O pin. If you do not that, MCU remains at reset mode.

[How to control the P1 input / output mode]

The initial value of P1DIR (Port 1 Input/Output Control) register is *****111. This means that the mode of P1[0], P1[1], and P1[2] is input mode. To configure the bit of the P1 to output mode, set to the bit of the P1DIR register to “0” (0 = Output, 1 = Input (Default)).

[How to control the P1 pull-up ON/OFF]

The initial value of P0SEL (Port 1 Pull-up Control) register is ******11. This means that the pull-up of

P1[0] and P1[1] is OFF. To enable the pull-up resistor control, set to the bit of the P1SEL register to “0” (0 = Pull-up ON, 1=Pull-up OFF (Default)). Note that P1[2] has no pull-up.

[How to control the P1 output type]

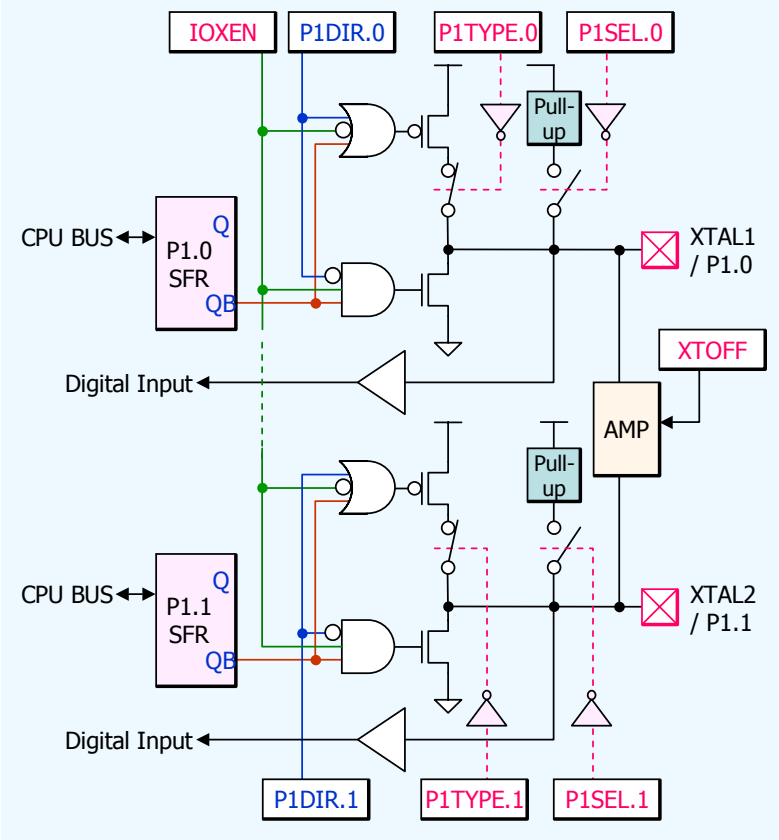
The initial value of P1TYPE (Port 1 Type Control) register is ******00h. This means that the output type of P1[0] and P1[1] is push-pull output type. To configure the bit output type of the P1[0] and P1[1] to open-drain output, set the bit of the P1TYPE register to “1” (0 = Push-pull output (Default), 1 = Open-drain output). Note that P1[2] is only open-drain output type. If user wants to use the P1[2] pin as output pin, this pin may need the external pull-up resistor.

[About the alternative pins and description]

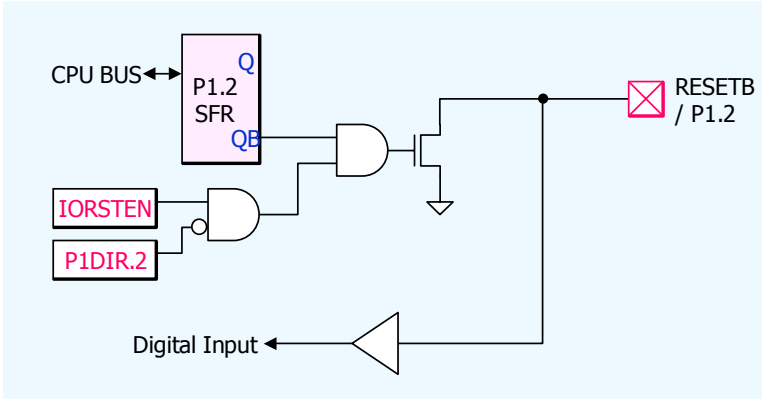
In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs

Port 1	-	-	-	-	-	P1[2]	P1[1]	P1[0]
						RESETB	XTAL2	XTAL1

Alternative Pin	Description
XTAL1	External Crystal Oscillator Input. (Refer to ALTSEL)
XTAL2	External Crystal Oscillator Output. (Refer to ALTSEL)
RESETB	Inverted Reset Input. Active Low. (Refer to ALTSEL)



(a) P1.0 (XTAL1) and P1.1 (XTAL2)



(b) P1.2 (RESETB)

Figure 6-4 Configuration of PORT1

6.2.1.2.1 **P1** (90h) : Port 1 Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	P0.2	P0.1	P0.0
						R/W(1)	R/W(1)	R/W(1)

This register functions as a general purpose I/O port. The initial value of P0 is FFh.

6.2.1.2.2 **P1TYPE** (D5h) : Port 1 Type Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	P1TYPE.1	P1TYPE.0
							R/W(0)	R/W(0)

This register configures the P0 output type to the push-pull or open-drain. The initial P1 output type is configured to push-pull output type. Note that the P1[2] pin is only the open-drain output type.

0 = Push-pull output type. (Default)

1 = Open-drain output.

6.2.1.2.3 **P1SEL** (E5h) : Port 1 Pull-up Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	P1SEL.1	P1SEL.0
							R/W(1)	R/W(1)

This register is enables/disabled the internal pull-up in P1. Note that the P1[2] has no pull-up.

0 = Pull-up ON. The bit is configured to pull-up ON.

1 = Pull-up OFF. The bit is configured to pull-up OFF. (Default)

6.2.1.2.4 **P1DIR** (F5h) : Port 1 Input/Output Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
						R/W(1)	R/W(1)	R/W(1)

This register is configured the P1 to input mode or output mode. The initial P1 mode is configured to input mode.

0 = Output mode. The bit is configured to output.

1 = Input mode. The bit is configured to input. (Default)

6.2.1.2.5 **ALTSEL** (E3h) : Alternative Function Selection Register

Bit No.	7	6	5	4	3	2	1	0
	IOXEN	IORSTEN	CLO	PWM00	TVO	TX	-	-
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)		

Symbol	Description
IOXEN	I/O bits from XTAL1/XTAL2 Enable. (XTAL1 : Input, XTAL2 : Output) 0 = The P1[1] and P1[0] bits are configured as XTAL Input/Output (Default) 1 = The P1[1] and P1[0] bits are configured as general I/O pins. Note that you must set XTOFF bit (PMR.3) to "1" to turn off the internal amplifier.
IORSTEN	I/O bits from RESETB Enable. 0 = The P1[2] bit is configured as RESETB input. (Default) 1 = The P1[2] bit is configured as general I/O pin.
-	Reserved

6.2.1.2.6 **PMR** (C4h) : Power Management Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	XTOFF	-	-	-
					R/W(0)			

Symbol	Description
XTOFF	Internal Amplifier Disable for External Crystal Oscillator. This bit controls the enable/disable the internal amplifier for external crystal oscillator. 0 = Internal amplifier is enabled. (Default) 1 = Internal amplifier is disabled. Set the XTOFF bit to "1" when user wants to use P1[0] and P1[2] as general I/O pins. Don't set XTOFF bit to "1" when XT/RG bit (EXIF.3) is "1". (XT/RG = "1" : The external clock is selected as system clock. Refer to EXIF register.)

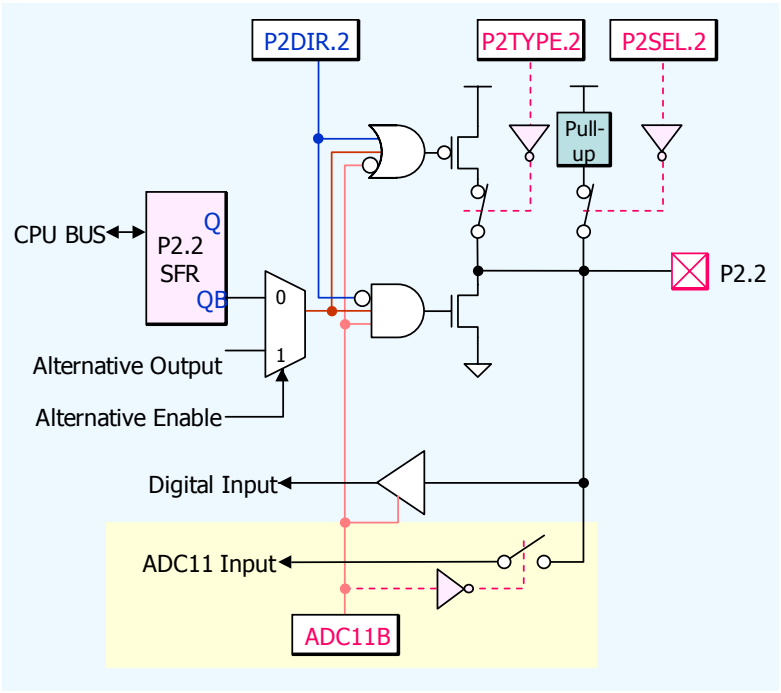


Figure 6-5 Configuration of PORT2

6.2.1.3 PORT 2

Port 2 has the following configuration modes in according to P2DIR, P2SEL, P2TYPE, ALTSEL, ADCON, ADCSEL, and ADCSELH register.

- Input / Output mode controlled by P2DIR (Default = *1111111, Input mode)
- Push-pull / Open-drain output type controlled by P2TYPE (Default = *0000000h, Push-pull output type)
- Pull-up ON / OFF controlled by P0SEL (Default = *1111111h, Pull-up ON)
- ADC input enable & digital input disable, or ADC input disable & digital input enable by ADCSEL and ADCSELH (Default = FFh is ADC input disable & digital input enable.)

After reset, port 2 is configured as input mode with push-pull output type and pull-up is turn-on. Also, ADC input channels are disabled. Note that the port 2[6:2] (= P2[6:2]) is configurable as an ADC input pins. The ADC7 ~ ADC11 input channels of the port 2 are selected by ADCSEL and ADCLSELH register.

[How to control the ADC input channel enable / disable for P2]

If an ADC input of port 2 is enabled, port 2 is automatically configured as ADC input channel without any effect of port 2 state. Namely, ADC input channel is valid by ADC selection registers (ADCSEL, ADCSELH) even if port 2 is operating on the output mode (bit value of P2DIR = 0). If ADC input channel is disabled, port configuration will be back to the state before entering the ADC input channel enable mode. To select the ADC channel input, set the bit of the ADCSEL or ADCSELH register to “0” (0 = ADC channel bit enable, 1 = disable (Default)). In more detail about ADC, refer to “**6.2.7 ADC**” part.

[How to control the P2 input / output mode]

The initial value of P2DIR (Port 2 Input/Output Control) register is *1111111. This means that the mode of P2 is input mode. To configure the bit of the P2 to output mode, set to the bit of the P2DIR register to “0” (0 = Output, 1 = Input (Default)).

[How to control the P2 pull-up ON/OFF]

The initial value of P2SEL (Port 2 Pull-up Control) register is *0000000h. This means that the pull-up of P2 is ON. To disable the pull-up resistor control, set to the bit of the P2SEL register to “1” (0 = Pull-up ON (Default), 1=Pull-up OFF).

[How to control the P2 output type]

The initial value of P2TYPE (Port 2 Type Control) register is *0000000h. This means that the output type of P2 is push-pull output type. To configure the bit output type of the P2 to open-drain output, set the bit of the P2TYPE register to “1” (0 = Push-pull output (Default), 1 = Open-drain output).

[How to control the Timer0 Counter from P2[0]]

If user wants to use the timer0 as counter from external input pin (P2[6]; T0 input pin), set the C/T bit (TMOD[2]) to “1”. A “1” to “0” transition on this T0 input pin will increment the timer0 counter.

The default value of C/T bit (TMOD[2]) is “0”, timer0 timer uses the system clock for timing. When the value of C/T bit is 1, timer0 counter uses the external T0 input transition for counter. In more detail, refer to TMOD register.

[About the alternative pins and description]

In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs.

Port 2	-	P2[6]	P2[5]	P2[4]	P2[3]	P2[2]	P2[1]	P2[0]
		ADC7 CLO	ADC8	ADC9	ADC10	ADC11		T0

Alternative Pin	Description
ADC11 ~ ADC7	ADC Analog Input 11 ~ 7
CLO	System Clock Output to P2[6]. (Refer to ALTSEL register)
T0	Timer0 Input or T0 Input. (Refer to TMOD register) A “1” to “0” transition on this T0 input pin will increment the timer0 counter.

6.2.1.3.1 **P2** (A0h) : Port 2 Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register functions as a general purpose I/O port. The initial value of P2 is *1111111.

6.2.1.3.2 **P2TYPE** (D6h) : Port 2 Type Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register configures the P2 output type to the push-pull or open-drain. The initial P2 output type is configured to push-pull output type.

0 = Push-pull output type. (Default)

1 = Open-drain output.

6.2.1.3.3 **P2SEL** (E6h) : Port 2 Pull-up Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register is enables/disabled the internal pull-up in P2.

0 = Pull-up ON. The bit is configured to pull-up ON. (Default)

1 = Pull-up OFF. The bit is configured to pull-up OFF.

6.2.1.3.4 **P2DIR** (F6h) : Port 2 Input/Output Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register is configured the P2 to input mode or output mode. The initial P2 mode is configured to input mode.

0 = Output mode. The bit is configured to output.

1 = Input mode. The bit is configured to input. (Default)

6.2.1.3.5 **ADCSELH** (E1h) ADC Channel Selection High Register

Bit No.	7	6	5	4	3	2	1	0
	ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

<table>
<tr><th>Symbol</th><th colspan="2">Description</th></tr>
<tr><td rowspan="6">ADC7B
ADC8B
ADC9B
ADC10B
ADC11B</td><td colspan="2">ADC Channel Selection Low. These bits are configured to enable/disable the ADC channel inputs (ADC[11:4]). P2[2] ~ P2[6] : ADC7 ~ ADC11 input.</td></tr>
<tr><td>ADC Channel</td><td>Result</td></tr>
<tr><td>ADC7B</td><td>1 = ADC7 input channel is disabled & digital input is enabled at P2[6]. (Default)
0 = ADC7 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC8B</td><td>1 = ADC8 input channel is disabled & digital input is enabled at P2[5]. (Default)
0 = ADC8 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>ADC11B</td><td>1 = ADC11 input channel is disabled & digital input is enabled at P2[2]. (Default)
0 = ADC11 input channel is enabled & digital input is disabled.</td></tr>
</table>

6.2.1.3.6 **ADCSEL** (E2h): ADC Channel Selection Low & MUX Selection Register

Bit No.	7	6	5	4	3	2	1	0
	ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

* R/W (1) means that user can read and writer this flag bit. The initial value of flag bit is “1”.

Symbol	Description				
CH0 CH1 CH2 CH3	ADC MUX Selection. Theses bits is configured to select the ADC input. The initial value of CH[3:0] is 0xFh (4’b1111). Ch, Dh, Eh and Fh means that all ADC inputs are disabled.				
	CH3	CH2	CH1	CH0	Result
	0	0	0	0	0x0h : ADC0 is selected
	0	0	0	1	0x1h : ADC1 is selected
	0	0	1	0	0x2h : ADC2 is selected
	...				
	1	0	1	0	0xAh : ADC10s selected
	1	0	1	1	0xBh : ADC11is selected
	1100 (0xCh) ~ 1111 (0xFh)				No ADC channel is not selected (Default Value = 0xFh)

6.2.1.3.7 **ALTSEL** (E3h) : Alternative Function Selection Register

Bit No.	7	6	5	4	3	2	1	0
	IOXEN	IORSTEN	CLO	PWM00	TVO	TX	-	-
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)		

Symbol	Description
CLO	System Clock Output Enable to P2[6]. 0 = Disable the System Clock Output to P2[6]. (Default) 1 = Enable the System Clock Output to P2[6].
-	Reserved

6.2.1.4 Read-Modify-Write Instructions

The normal read instructions will read the pin state without regard to the output data latch. The only exception is the read-modify-write category of instructions. They are listed as follows:

Table 6-4 Read-Modify-Write Instructions

Instruction	Description
ANL	Logical AND
ORL	Logical OR
XRL	Logical Exclusive OR (XOR)
JBC	Branch if bit is set then clear bit
CPL	Complement bit
INC	Increment
DEC	Decrement
DJNZ	Decrement and branch if not zero
MOV PX.Y, C	Move the carry bit to bit Y of Port X
CLR PX.Y	Clear bit Y of Port X
SETB PX.Y	Set bit Y of Port X

The read-modify-write instructions read the state of the latch, and then write back the result to the latch. Thus the operation takes place using the value that was originally written to the SFR, without regard to the pin state. The last three instructions listed above are read-modify-write because they read the entire port latch, then write back the changed value. In this case, only one bit will be changed as specified by the instruction.

6.2.1.5 Alternative Functions

Every port pin has an optional special function. These functions are individually selectable. They can also be turned on and off dynamically according to the application. The alternative function for each port pin is described briefly below. More information about each alternative function is available in the section dealing with that function.

Table 6-5 Alternative Functions

Port	Alternative Functions
P0[0]	(INT0B) External Interrupt 0 falling edge active (TVO) Timer 0 Overflow Clock Output
P0[1]	(ADC0) ADC Input Channel 0 (INT1B) External Interrupt 1 falling edge active
P0[2]	(ADC1) ADC Input Channel 1 (INT2) External Interrupt 2 rising edge active
P0[3]	(ADC2) ADC Input Channel 2 (INT3B) External Interrupt 3 falling edge active
P0[4]	(ADC3) ADC Input Channel 3 (AVref) ADC Reference Voltage Input
P0[5]	(ADC4) ADC Input Channel 4
P0[6]	(ADC5) ADC Input Channel 5 (PWM) PWM Output
P0[7]	(ADC6) ADC Input Channel 6
P1[0]	(XTAL1) Crystal Oscillator Input
P1[1]	(XTAL2) Crystal Oscillator Output
P1[2]	(RESETB) RESET Input –Active Low
P2[0]	(T0) Timer 0 Input
P2[1]	
P2[2]	(ADC11) ADC Input Channel 11
P2[3]	(ADC10) ADC Input Channel 10
P2[4]	(ADC9) ADC Input Channel 9
P2[5]	(ADC8) ADC Input Channel 8
P2[6]	(ADC7) ADC Input Channel 7 (CLO) System Clock Output

6.2.2 LVD (Low Voltage Detector)

6.2.2.1 Power–On Reset (POR)

Members of the MiDAS1.1 family incorporate an internal voltage reference which holds the CPU in the power–on reset state while V_{DD} is out of tolerance. Once V_{DD} has risen above the threshold (V_{RST} = 2.3V), the microcontroller will restart the oscillation of the external crystal and count 65,536 clock cycles. The processor will then begin software execution at location 0000h.

This helps the system maintain reliable operation by only permitting processor operation when voltage is in a known good state. The processor will exit the reset condition automatically once the above conditions are met. This happens automatically, needing no external components or action. Execution begins at the standard reset vector address of 0000h.

Software can determine that a Power–On Reset has occurred using the Power–On Reset flag (POR in PCON[4]). Since all resets cause a vector to location 0000h, the POR flag allows software to acknowledge that power failure was the reason for a reset. Software should clear the POR bit after reading it. When a reset occurs, software will be able to determine if a power cycle was the cause.

6.2.2.2 Power-Fail Reset

When power fails (below V_{RST}), the power monitor will invoke the reset state again. This reset condition will remain while power is below the threshold (V_{RST}). When power returns above the reset threshold (V_{RST}), a full power–on reset will be performed. Thus a brown-out that causes V_{DD} to drop below V_{RST} appears the same as a power up.

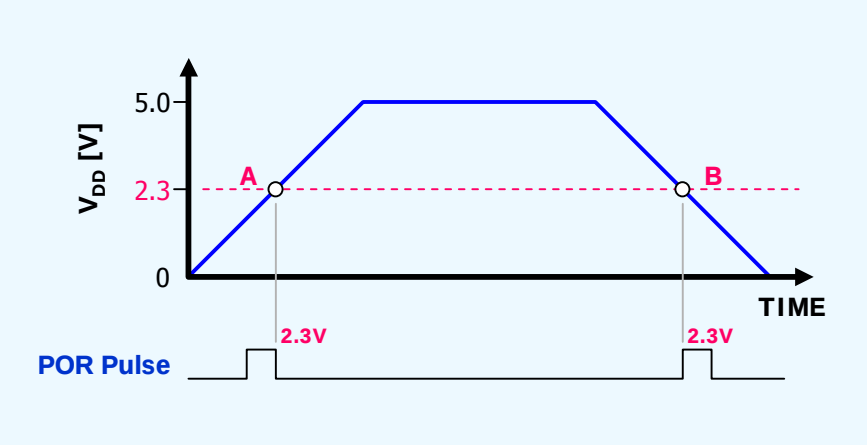


Figure 6-6 Power-On Reset/Power-fail Reset and Power-fail interrupt (V_{RST}=2.3V)

[Related SFRs are: EXIF (External Interrupt Flag) and PCON (Power Control) Registers]

- **EXIF**[0] : (BGS) Band-gap Select (Default = 1).
  If BGS=0, band-gap block (LVD) will do not run in power-down mode, but run during normal mode. It will support the significant power savings in power-down mode.
- **PCON**[4] : (POF) Power Off Fail. After POR (Power-On Reset), POF bit will be set to "1".
  For all resets except of POR, the value of POF is not changed if user sets it bit to "0".
- **PCON**[1] : (PD) Power-down (Stop) Mode Enable (Default = 0).
  If user sets this bit to "1", it causes the MCU to go into the power-down mode.

Power-On/Fail reset and interrupt signal is generated when V_{DD} goes up and down as shown in Figure 6-6.

LVD block is alive after power-up. LVD block is off only when BGS = 0 and power-down mode.

If the use of the power–fail features are desired in power-down mode (PCON[1] = 1), the BGS bit (EXIF[0]), may be used. When BGS bit is set to logic 1 by software, the band–gap reference and associated power monitor circuits will remain active in power-down mode. The price of this feature is higher power supply current requirements. In power-down mode with the band–gap disabled, the processor draws less than 1 μA. With the band–gap enabled, it draws approximately 180 μA.

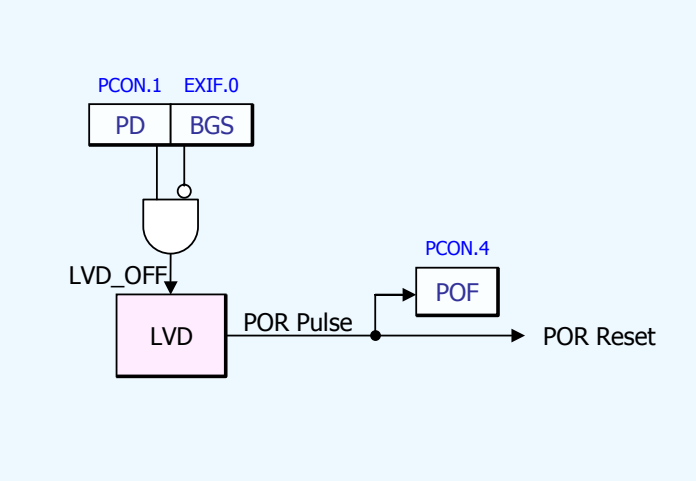


Figure 6-7 LVD Block Diagram

6.2.2.3 **PCON** (87h) : Power Control Register

Bit No.	7	6	5	4	3	2	1	0
	SMOD1	-	-	POF	GF1	GF0	PD	IDL
	R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
-	Reserved
POF	Power Off Flag. When POR (Power-On Reset), this bit will be set to "1" by H/W. For all resets except of POR, the value of POF is not changed if user sets it bit to "0".
PD	Power-down (Stop) Mode Enable. Setting this bi causes the MCU (MiDAS1.1 family) to go into the power-down mode. In this mode, all the clocks are stopped and program execution is frozen. 0 (Default), 1 = Power-down mode enable.

6.2.2.4 **EXIF** (91h) : External Interrupt Flag Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	IE3	IE2	XT/RG	RGMD	RGSL	BGS
			R/W(0)	R/W(0)	R/W(1)	R(1)	R/W(0)	R/W(1)

Symbol	Description
-	Reserved
BGS	Band-Gap Select. This bit enables/disables the band-gap reference during power-down mode. Disabling the band-gap reference provides significant power savings in power-down mode, but sacrifices the ability to perform power-fail reset while stopped. The state of bit will be undefined on devices which do not incorporate a band-gap reference. 0 = The band-gap reference (LVD) will do not run in power-down mode, but will run during normal operation. 1 = The band-gap reference (LVD) will run in power-down mode. (Default)

6.2.3 WDT (Watchdog Timer)

The Watchdog timer is a free-running timer which can be programmed by user to serve as a system monitor, a time-base generator, or an event timer. And it can detect the malfunction of program due to external noise or other causes. It is basically a set of dividers that divide the system clock. The divider output is selectable and determines the time-out interval. When the time-out occurs (counter overflow), a flag is set, which can cause an interrupt if enabled, and a system reset can also be caused if it is enabled. The interrupt will occur if the individual interrupt enable and the global enable are set. The interrupt and reset functions are independent of each other and may be used separately or together depending on the user software. Figure 6-8 shows the block diagram for Watchdog timer.

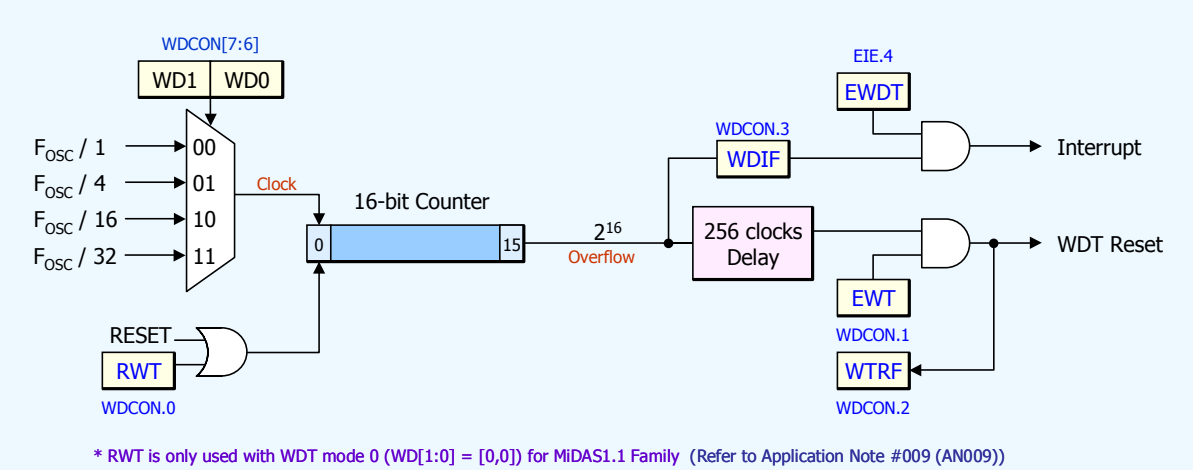


Figure 6-8 Block Diagram for Watchdog Timer

Note: RWT is only used with WDT mode 0 (WD[1:0] = 2'b00) for MiDAS1.1 Family.

The Watchdog timer should first be restarted by using RWT (WDCON.0). This ensures that the timer starts from a known state. This bit is self-clearing, i.e. after writing a 1 to this bit, the hardware will automatically clear this bit. Not that WDT counter restart (RWT = 1) must be executed in WDT mode 0 (WD1 and WD0 are both "0").

The Watchdog timer will now count clock cycles. The time-out interval is selected by the two bits WD1 and WD0 (CKCON.7 and CKCON.6). When the selected time-out occurs, the Watchdog interrupt flag WDIF (WDCON.3) is set. If the Watchdog reset EWT (WDCON.1) is enabled, then 256 clocks after the time-out, if there is no RWT, a system reset due to Watchdog timer will occur. This will last for thirty clock cycles, and the Watchdog timer reset flag WTRF (WDCON.2) will be set. This indicates to the software that the Watchdog was the cause of the reset.

When used as a simple timer, the reset and interrupt functions are disabled. The timer will set the WDIF flag each time the timer completes the selected time interval. The WDIF flag is polled to detect a time-out and the RWT allows software to restart the timer. The Watchdog timer can also be used as a very long timer. The interrupt feature is enabled in this case. Every time the time-out occurs an interrupt will occur if the global interrupt enable EA is set.

The main use of the Watchdog timer is as a system monitor. This is important in real-time control applications. In case of some power glitches or electro-magnetic interference, the processor may begin to execute errant code. If this is left unchecked, the entire system may crash. Using the Watchdog timer interrupt during software development will allow the user to select ideal Watchdog reset locations. The code is first written without the Watchdog interrupt or reset. Then the Watchdog interrupt is enabled to identify code locations where interrupt occurs. The user can now insert instructions to reset the watchdog timer which will allow the code to run without any watchdog timer interrupts. Now the Watchdog timer reset is enabled and the watchdog interrupt may be disabled. If any errant code is executed now, then the watchdog timer reset instructions will not be executed at the required instants and watchdog reset will occur.

The watchdog time-out selection will result in different time-out values depending on the clock speed. The watchdog timer will generate a reset after 256 clocks from after the time-out being occurred.

Table 6-6 Time-out Values for the Watchdog Timer

Mode	WD1	WD0	Interrupt Time-Out (@4MHz)		Reset Time-Out (@4MHz)	
0	0	0	1 X 2^{16} clocks	16.38 ms	1 X 2^{16} + 256 clocks	16.45 ms
1	0	1	4 X 2^{16} clocks	65.54 ms	4 X 2^{16} + 256 clocks	65.60 ms
2	1	0	16 X 2^{16} clocks	262.14 ms	16 X 2^{16} + 256 clocks	262.21 ms
3	1	1	132X 2^{16} clocks	524.29 ms	32 X 2^{16} + 256 clocks	524.35 ms

The Watchdog timer will be disabled by a power-on/fail reset. The Watchdog timer reset does not disable the watchdog timer, but will restart it. In general, software should restart the timer to put it into a known state.

The default Watchdog time-out is WDT mode 3 (32 x 2^{16} clocks), which is the shortest time-out period.

6.2.3.1 **WDCON** (D8h) : Watchdog Control Register

Bit No.	7	6	5	4	3	2	1	0
	WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
	R/W(1)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

<table>
<tr><th>Symbol</th><th colspan="5">Description</th></tr>
<tr><td rowspan="6">WD1
WD0</td><td colspan="5">Watchdog Timer Mode Select</td></tr>
<tr><td>Mode</td><td>WD1</td><td>WD0</td><td>Interrupt Time-Out</td><td>Reset Time-Out</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1 X 2^{16} clocks</td><td>1 X 2^{16} + 256 clocks</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>4 X 2^{16} clocks</td><td>4 X 2^{16} + 256 clocks</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>16 X 2^{16} clocks</td><td>16 X 2^{16} + 256 clocks</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>132X 2^{16} clocks</td><td>32 X 2^{16} + 256 clocks</td></tr>
<tr><td rowspan="7">WDIF</td><td colspan="5">Watchdog Interrupt Flag.
This bit, in conjunction with the Watchdog timer interrupt enable bit EWDT (EIE.4) and Enable Watchdog Timer Reset bit EWT (WDCON.1) indicates if a watchdog timer event has occurred and what action will be taken. This bit must be cleared in software before exiting the interrupt service routine, or another interrupt will be generated. Setting this bit in software will generate a watchdog interrupt if enabled.
(EWT → WDCON.1, EWDT → EIE.4, WDIF → WDCON.3)</td></tr>
<tr><td>EWT</td><td>EWDT</td><td>WDIF</td><td colspan="2">Description</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td colspan="2">No WDT (Watchdog timer) event has occurred.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.
WDT reset will occur after 512 clocks if RWT is not strobed.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.
WDT reset will occur after 512 clocks if RWT is not set.</td></tr>
<tr><td>WTRF</td><td colspan="5">Watchdog Timer Reset Flag.
When set, this bit indicates that a watchdog timer reset has occurred. It is typically interrogated to determine if a reset was caused by watchdog timer reset. It is cleared by a power-on-reset, but otherwise must be cleared by software before the next reset of any kind or software may erroneously determine that a watchdog timer reset has occurred. Setting this bit in software will not generate a watchdog timer reset. If the EWT bit is cleared, the watchdog timer will have no effect on this bit.</td></tr>
</table>

EWT	Enable Watchdog Timer Reset. This bit enables/disables the ability of the watchdog timer to reset the device. This bit has no effect on the ability of the watchdog timer to generate a watchdog interrupt. The time-out period of the watchdog timer is controlled by the Watchdog Timer Mode Select bits (CKCON.7-6). Clearing these bits will disable the ability of the watchdog timer to generate a reset, but have no affect on the timer itself, or its ability to generate a watchdog timer interrupt. 0 = A time-out of the watchdog timer will not cause the device to reset. 1 = A time-out of the watchdog timer will cause the device to reset.
RWT	Restart Watchdog Timer. This bit serves as the strobe for the Watchdog function. During the time-out period, software must set the RWT bit if the Watchdog is enabled. Failing to set the RWT will cause a reset when the time-out has elapsed. There is no need to set the RWT bit to a 0 because it is self-clearing. Note that RWT flag bit must be set to "1" at only WDT mode 0. After watchdog timer is restarted, WDT mode can be changed to mode 3 or any mode.

6.2.4 Timer 0/1

The MiDAS1.1 family has two 16-bit programmable timer/counters.

6.2.4.1 Timer/Counter 0

The Timer/Counter 0 has two 8-bit registers which form the 16-bit counting register. They are the upper 8-bit register TH0 and the lower 8-bit register TL0.

When configured as a “Timer”, the timer counts clock cycles. The timer clock can be programmed to be thought of as 1/12 of the system clock. In the “Counter” mode, the register is incremented on the falling edge of the external input pin, T0 in case of Timer 0. The T0 input is sampled at S3 state of 12-clock system. If the sampled value is high in one machine cycle and low in the next, then a valid high-to-low transition on the pin is recognized and the count register is incremented. Since it takes 24 clocks (12-clock system) to recognize a negative transition on the pin, the maximum rate at which counting will take place is 1/24 of the master clock frequency. In either the “Timer” or “Counter” mode, the count register will be updated at S2 state. Therefore, in the “Timer” mode, the recognized negative transition on pin T0 can cause the count register value to be updated only in the machine cycle following the one in which the negative edge was detected.

The “Timer” or “Counter” function is selected by the “C/T” bit in the TMOD SFR. The Timer/Counter 0 has one selection bit for its own; bit 2 of TMOD selects the function for Timer/Counter 0. In addition the Timer/Counter 0 can be set to operate in any one of four possible modes. The mode selection is done by bits M0 and M1 in the TMOD SFR.

6.2.4.1.1 Time-base

The MiDAS1.1 family gives the user one time base for the timer. The timers can be programmed to operate like the standard 80C52 family, counting at the rate of 1/12 of the clock speed (F_{OSC}/12). This will ensure that timing loops on the MiDAS1.1 family and the standard 80C52 family can be matched. This is the default mode of operation of the MiDAS1.1 timers.

6.2.4.1.2 Mode 0

In Mode 0, the timer/counter 0 operates as an 8-bit counter with a divide-by-32 prescaler. In this mode, we have a 13-bit timer/counter. The 13-bit counter consists of 8 bits of TH0 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored.

The negative edge of the clock increments the counts in the TL0 register. When the fifth bit in TL0 moves from 1 to 0, then the count in the TH0 register is incremented. When the count in TH0 moves from FFh to 00h, the overflow flag TF0 in TCON SFR is set. The counted input is enabled only if TR0 is set and either GATE=0 or INT0B=1. When C/T is cleared to 0, then it will count clock cycles, and if C/T

is set to 1, then it will count 1 to 0 transitions on T0 (P3.4) for Timer 0.

When the 13-bit counter reaches 1FFFh, the count will cause it to roll-over to 0000H. The timer overflow flag TF0 of the relevant timer is set and if enabled an interrupts will occur.

$$\text{Time}[\text{sec}] = \left(\frac{F_{OSC}}{12} \right)^{-1} \times 2^{13}$$

6.2.4.1.3 Mode 1

Mode 1 is similar to Mode 0 except that the counting register forms a 16-bit counter, rather than a 13-bit counter. This means that all the bits of TH0 and TL0 are used. Roll-over occurs when the timer changes from FFFFh to 0000h. The timer overflow flag TF0 of the relevant timer is set and if enabled an interrupt will occur. The selection of the time-base in the timer mode is similar to that in Mode 0. The gate function operates similarly to that in Mode 0.

$$\text{Time}[\text{sec}] = \left(\frac{F_{OSC}}{12} \right)^{-1} \times 2^{16}$$

6.2.4.1.4 Mode 2

In Mode 2, the Timer/Counter 0 is in the auto reload mode. In this mode, TL0 operates as an 8-bit counter, while TH0 holds the reload value. When the TL0 register overflows from FFH to 00H, the TF0 bit in TCON is set and TL0 is reloaded with the contents of TH0, and the counting process continues from reloaded value. The reload operation leaves the contents of the TH0 register unchanged. Counting is enabled by the TR0 bit and proper setting of GATE and INT0B pins. As in the other two modes 0 and 1, Mode 2 allows counting of pulses on pin T0.

$$\text{Time}[\text{sec}] = \left(\frac{F_{OSC}}{12} \right)^{-1} \times \left(2^{8} - \text{TH0} \right)$$

6.2.4.1.5 Mode 3

In the Mode 3, Timer/Counter 0 configures TL0 and TH0 as two separate 8-bit counters in this mode. TL0 uses the Timer/Counter 0 control bits C/T, GATE, TR0, INT0B and TF0. The TL0 can be used to count clock cycles (clock/12) or 1-to-0 transitions on pin T0 as determined by C/T (TMOD.2). TH0 is forces as a clock cycle counter (clock/12) and takes over the use of TR1 and TF1 from Timer/Counter 1. Mode 3 is used in cases where an extra 8-bit timer is needed. With Timer 0 in Mode 3, Timer 1 can still be used in Modes 2. It can be used as a baud rate generator for the UART.

Figure 6-9 illustrates the various supported modes in Timer/Counter 0/1.

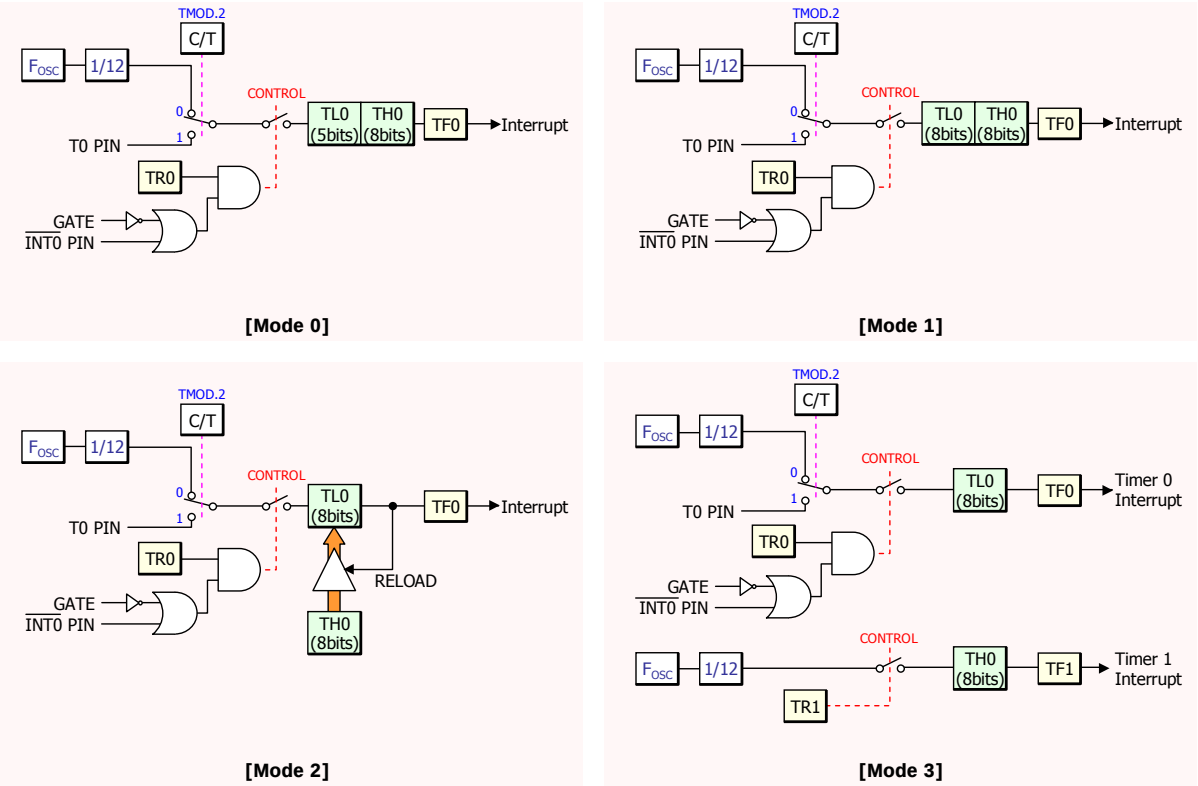


Figure 6-9 Timer/Counter 0 in Mode 0/1/2/3

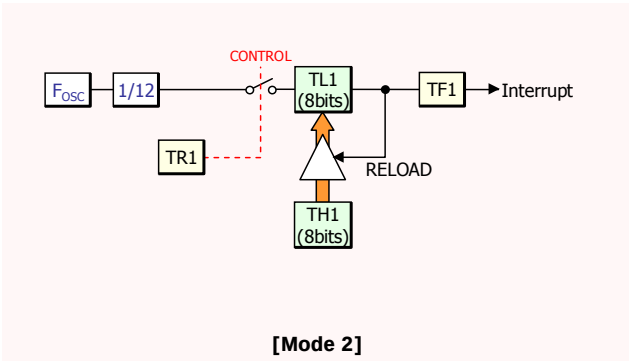


Figure 6-10 Timer/Counter 1 in Mode 2

6.2.4.2 Timer/Counter 1

The Timer/Counter 1 has two 8-bit registers, TH1 and TL1. (Similarly Timer/Counter 1 has two 8-bit registers, TH1 and TL1. They can be configured to operate only as timers counting machine cycles and configured only as a “Timer”, the timer counts clock cycles. The timer clock can be programmed to be thought of as 1/12 of the system clock.

6.2.4.2.1 Time-base

The MiDAS1.1 family gives the user one time base for the timer. The timers can be programmed to operate like the standard 80C52 family, counting at the rate of 1/12 of the clock speed. This will ensure that timing loops on the MiDAS1.1 family and the standard 80C52 family can be matched. This is the default mode of operation of the MiDAS1.1 timers. The Timer/Counter 1 is supported with only Mode 2.

6.2.4.2.2 Mode2

In Mode 2, the Timer/Counter 1 is in the auto reload mode. In this mode, TL1 operates as an 8-bit counter, while TH1 holds the reload value. When the TL1 register overflows from FFH to 00H, the TF1 bit in TCON is set and TL1 is reloaded with the contents of TH1, and the counting process continues from reloaded value. The reload operation leaves the contents of the TH1 register unchanged.

$$\text{Time}[\text{sec}] = \left(\frac{\text{F}_{\text{OSC}}}{12} \right)^{-1} \times \left(2^8 - \text{TH1} \right)$$

6.2.4.3 **TCON** (88h) : Timer/Counter 0/1 Control Register

Bit No.	7	6	5	4	3	2	1	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
TF1	Timer 1 Overflow Flag. This bit indicates when Timer 1 overflows its maximum count as defined by the current mode. This bit can be cleared by software. 0 = No Timer/Counter 1 overflow has been detected. 1 = Timer/Counter 1 has overflowed with its maximum count
TR1	Timer 1 Run Enable.
TF0	Timer 0 Overflow Flag. This bit indicates when Timer 0 overflows its maximum count as defined by the current mode. This bit can be cleared by software. 0 = No Timer/Counter 0 overflow has been detected. 1 = Timer/Counter 0 has overflowed with its maximum count
TR0	Timer 0 Run Enable.

6.2.4.4 **TMOD** (89h) : Timer/Counter 0 Mode Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	GATE	C/T	M1	M0
					R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description			
-	Reserved			
GATE	Timer 0 Gate Control. This bit enables/disables the ability of Timer 0 to increment. 0 = Timer 0 will clock when TR0=1, regardless of the states of INT0B. (Default) 1 = Timer 0 will clock only when TR0=1 and INT0B=1.			
C/T	Timer 0 Counter/Timer Select. 0 = Timer. Timer 0 is incremented by internal clocks (system clock). (Default) 1 = Counter. Timer 0 is incremented by pulsed on T0 pin when TR0=1.			
M1, M0				
	Mode	M1	M0	Description
	0	0	0	8 bits with 5-bit prescale
	1	0	1	16 bits
	2	1	0	8 bits with auto-reload
	3	1	1	Two 8-bit timer/counter

6.2.4.5 **TL0** (8Ah) : Timer/Counter 0 Low Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the least significant byte of Timer 0.

6.2.4.6 **TL1** (8Bh) : Timer/Counter 1 Low Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the least significant byte of Timer 1.

6.2.4.7 **TH0** (8Ch) : Timer/Counter 0 High Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the most significant byte of Timer 0.

6.2.4.8 **TH1** (8Dh) : Timer/Counter 1 High Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the most significant byte of Timer 1.

6.2.5 UART (Universal Asynchronous Rx/Tx)

UART in the MiDAS1.1 family is a full duplex port. The UART ports are capable of asynchronous communication. In the asynchronous mode, full duplex operation is available. This means that it can simultaneously transmit and receive data. The transmit register and the receive buffer are both addressed as SBUF register. However any write to SBUF will be to the transmit register, while a read from SBUF register will be from the receive buffer. The UART port can operate in only Mode 1 as described below.

6.2.5.1 Mode 1

In Mode 1, the full duplex asynchronous mode is used. Serial communication frames are made up of 10 bits transmitted on TXD and received on RXD. The 10 bits consist of a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receiving, the stop bit goes into RB8 in the SCON. The baud rate in this mode is variable. The serial baud rates can be programmed to be 1/16 or 1/32 of the Timer 1 overflow. Since the Timer 1 can be set to different reload values, a wide variation in baud rates is possible.

[Transmission for UART Communication]

Transmission begins with a write to SBUF. The serial data is brought out on to TXD pin at S1 state following the first roll-over of divide by 16-bit counter. The next bit is placed on TXD pin at S1 state following the next roll-over of the divide by 16-bit counter. Thus the transmission is synchronized to the divide by 16-bit counter and not directly to the write to SBUF signal. After all 8-bit data are transmitted, the stop bit is transmitted. The TI flag is set in the S1 state after the stop bit has been put out on TXD pin. This will be at the 10^{th} roll-over of the divide by 16-bit counter after a write to SBUF.

[Reception for UART Communication]

Reception is enabled only if REN is “1”. The UART port actually starts the receiving of serial data, with the detection of a falling edge on the RXD pin. The 1-to-0 detector continuously monitors the RXD line, sampling it at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide by 16-bit counter is immediately reset. This helps to align the bit boundaries with the roll-overs of the divide by 16-bit counter.

The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three bases. The bit detector samples the RXD pin, at the 8^{th}, 9^{th}, and 10^{th} counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port. If the first bit detected after the falling edge of RXD pin is not 0, then this indicates an invalid start bit, and the reception is immediately aborted. The serial port again looks for a falling edge in the RXD line. If a valid start bit is detected, then the rest of the bits are also

detected and shifted into the SBUF.

After shifting in 8-bit data, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before the loading and setting of RI can be done.

1. RI must be 0 and
2. the received stop bit = 1

If these conditions are met and the stop bit goes to RB8, the 8-bit data go into SBUF and RI is set. Otherwise the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RXD pin.

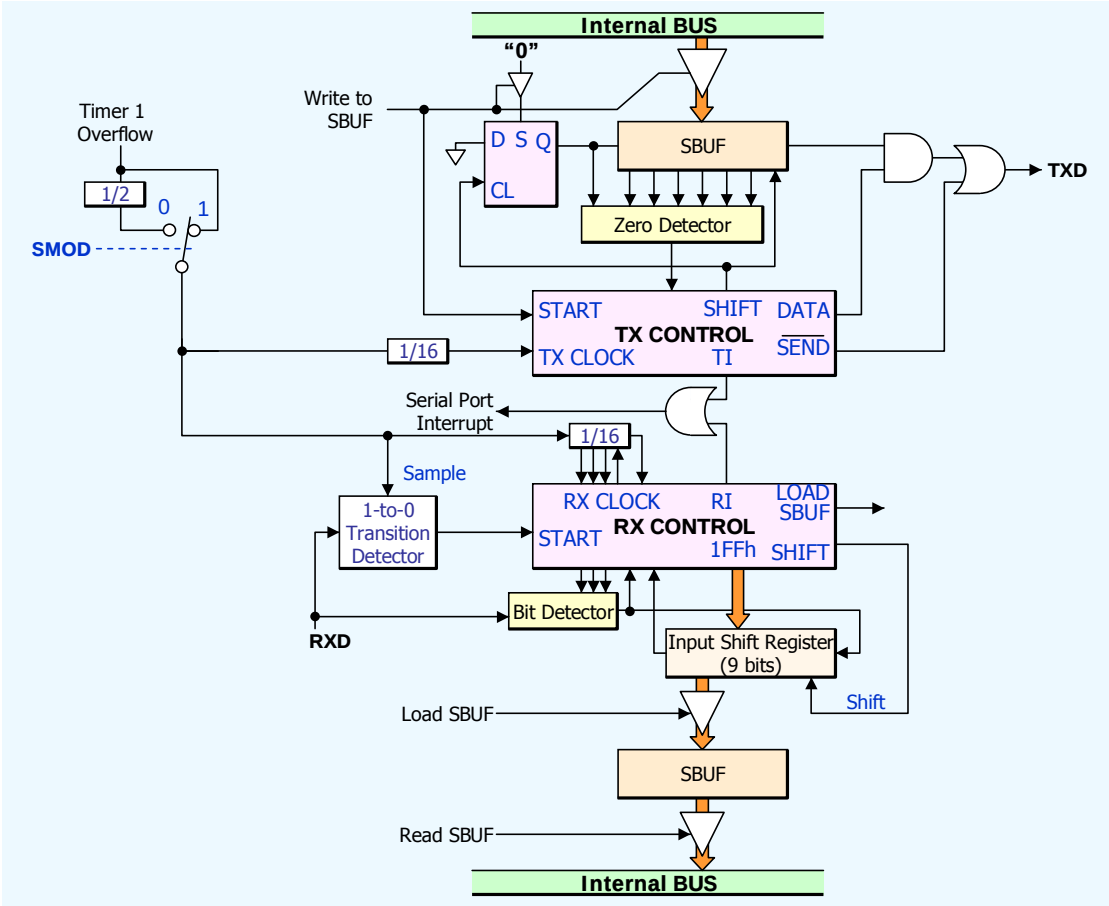


Figure 6-11 UART Mode 1

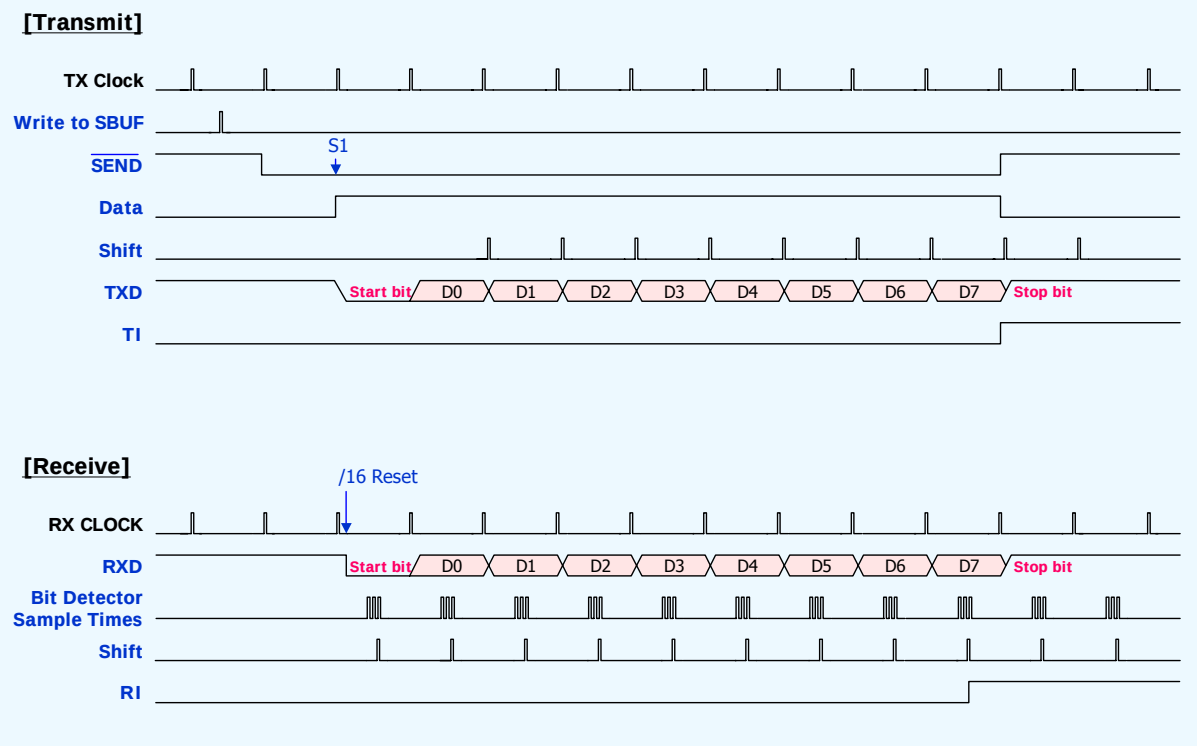


Figure 6-12 UART Mode 1 Timing

Table 6-7 Summary of Mode in UART

UART Mode	Data Size		Baudrate
Mode 1	10 bits	Start bit (0) 8 bits data Stop bit (1)	1/32 X Timer1 Overflow. (When SMOD1 = 0) 1/16 X Timer1 Overflow. (When SMOD1 = 1)

6.2.5.2 Baudrate Example

We can summarize the example using the equation of baudrate.

[Equation for Baudrate]

$$\text{Baudrate[bps]} = \frac{2^{\text{SMOD1}}}{32} \times \left(\frac{F_{OSC}}{12} \right) \times \frac{1}{2^8 - \text{TH1}} = \frac{2^{\text{SMOD1}}}{32} \times (\text{Timer1 Overflow})$$

Table 6-8 Example of Baudrate

Baudrate	UART Mode	F_{osc} [MHz]	SMOD1	Timer 1		
				C/T	Mode	Reload Value (TH1)
62.5 KHz	Mode 1	12	1	0	Mode 2, 8-bit, Auto-reload	FFh
19.2 KHz		11.0592	1	0		FDh
9.6 KHz		11.0592	0	0		FDh
4.8 KHz		11.0592	0	0		FAh
2.4 KHz		11.0592	0	0		F4h
1.2 KHz		11.0592	0	0		E8h
137.5 Hz		11.0592	0	0		1Dh
110 Hz		6	0	0		72h

6.2.5.3 PCON (87h) : Power Control Register

Bit No.	7	6	5	4	3	2	1	0
	SMOD1	-	-	POF	GF1	GF0	PD	IDL
	R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
SMOD1	Timer1 Buadrate Double in UART mode. 0 = Double baudrate disable (Default) 1 = Double baudrate enable
-	Reserved

6.2.5.4 **SCON** (98h) : Serial Port Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	REN	-	-	TI	RI
				R/W(0)			R/W(0)	R/W(0)

Symbol	Description
-	Reserved
REN	Serial Reception Enable. 0 = Serial reception disable. (Default) 1 = Serial reception enable.
TI	Transmission Interrupt Flag. This bit indicates that a byte of data in the serial port buffer (SBUF) has been completely shifted out. Must be cleared by software.
RI	Reception Interrupt Flag. This bit indicates that a byte of data has been received in the serial port buffer (SBUF). Must be cleared by software.

6.2.5.5 **SBUF** (99h) : Serial Data Buffer Register

Bit No.	7	6	5	4	3	2	1	0
	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Data for serial port is read from or written to this location. The buffers for serial transmission and serial reception are separate registers, but both are addressed at the same location.

6.2.6 PWM (Pulse Width Modulator)

The MiDAS1.1 family has one-channel 8-bit PWM circuit and enables two outputs. The operation of circuits in PWM is controlled by each control register; PWMCON.

The PWM counter can be used as 8-bit incrementing counters. Under operation, the counters increment until overflow occurs. To enable the PWM waveform output to P0.6, you set PWMCON[7] to “1”. And to enable the PWM waveform output to P0.0, you set ALTSEL[4] to “1”. To start the counter and enable the PWM block, you set PWMCON[0] to "1". If the counter is stopped, it retains its current count value; when re-started, it resumes counting from the retained count value. When there is a need to clear the counter you set PWMCON[0] to "0". To the value of the counter, set the clear bit, PWMCON[1] to “1”.

You can select a clock for the PWM counter by set PWMCON[6:4]. Clocks which you can select are $F_{OSC}/128$ $F_{OSC}/64$, $F_{OSC}/32$, $F_{OSC}/16$, $F_{OSC}/8$, $F_{OSC}/4$, $F_{OSC}/2$, and $F_{OSC}/1$.

6.2.6.1 Block Diagram

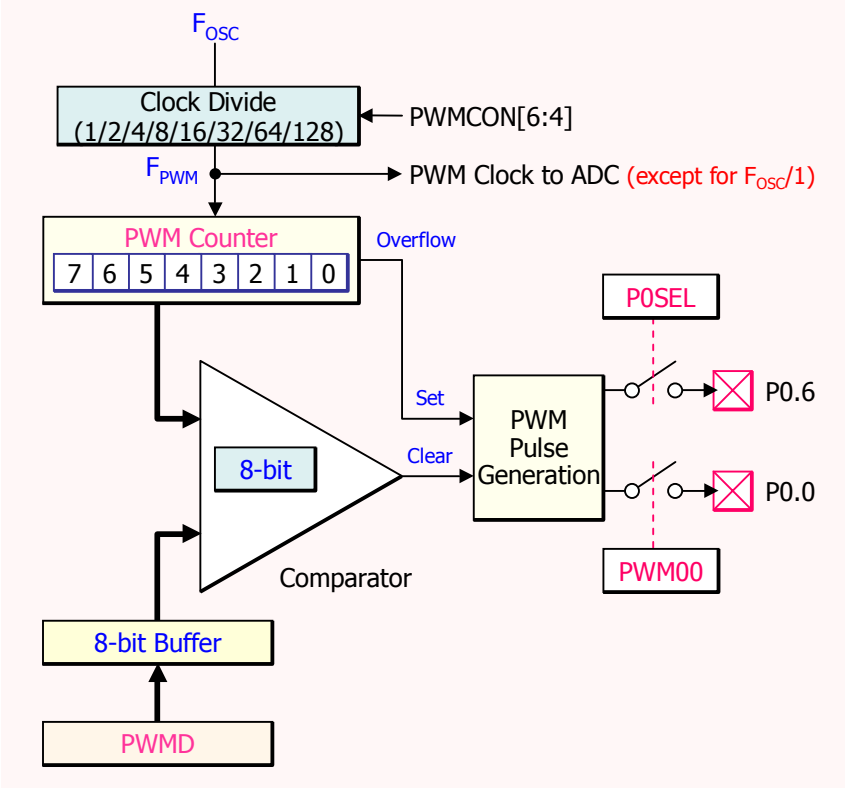


Figure 6-13 Functional block diagram

6.2.6.2 PWM components

The 8-bit PWM circuits have the following components:

- 8-bit comparator and extension cycle circuit
- 8-bit reference data register (PWMD)
- PWM output pins (PWM waveform output to P0.6 by PWMCON[7] or P0.0 by ALTSEL[4])

6.2.6.3 PWM counter

PWM counter operates in 8-bit counter mode. In this mode, the counter increments using all 8 bits in the counter and decides on PWM frequency compared to the PWMD. The mode is used in low frequency application simply.

6.2.6.4 PWM frequency (F_{PWM})

The timing characteristics of PWM output are based on the prescaled F_{OSC} clock frequency. The PWM counter clock value is determined by the setting of PWMCON[6:4] (Prescaled Cock Selection for PWM). Table 6-9 shows the clock rate by PWMCON[6:4].

Note: PWM clock (F_{PWM}) to ADC should not be set to F_{OSC}/1.

PWMCON[6:4]			PWM Clock Rate
PS2_P0	PS1_P0	PS0_P0	(F_{PWM})
0	0	0	F_{OSC} / 1
0	0	1	F_{OSC} / 2
0	1	0	F_{OSC} / 4
0	1	1	F_{OSC} / 8
1	0	0	F_{OSC} / 16
1	0	1	F_{OSC} / 32
1	1	0	F_{OSC} / 64
1	1	1	F_{OSC} / 128

Table 6-9 PWM clock rate by PWMCON[6:4]

6.2.6.5 PWM Function Description

The PWM output signal toggles to Low level whenever the 8bits of counter matches the reference data register (PWMD). If the values in the PWMD register are not zero, an overflow of the 8bits of counter causes the PWM output to toggle from Low level to High level. In this way, the reference value written to the reference data register determines the module's base duty cycle.

6.2.6.6 PWM Timing Diagram

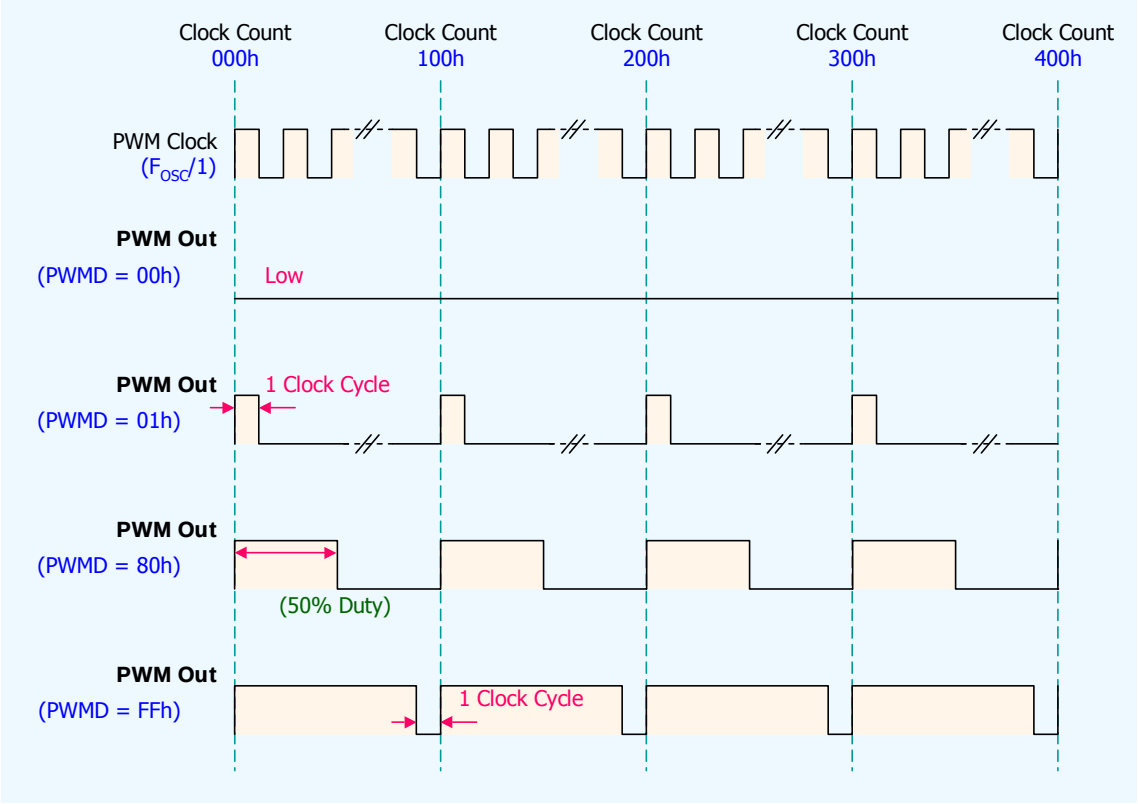


Figure 6-14 PWM basic timing diagram

6.2.6.7 **PWMCON** (DCh) : PWM Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0SEL	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

Symbol	Description			
P0SEL	PWM Waveform Output Enable to P0[6]. 0 = Disable the PWM waveform output to P[6]. (Default) 1 = Enable the PWM waveform output to P[6].			
PS2_P0 PS1_P0 PS0_P0	Prescaled Clock Selection. Note that PWM clock (F_{PWM}) to ADC should not be set to F_{OSC}/1.			
	PS2_P0	PS1_P0	PS0_P0	PWM Clock Rate (F_{PWM})
	0	0	0	F_{OSC} / 1
	0	0	1	F_{OSC} / 2
	0	1	0	F_{OSC} / 4
	0	1	1	F_{OSC} / 8
	1	0	0	F_{OSC} / 16
	1	0	1	F_{OSC} / 32
	1	1	0	F_{OSC} / 64
	1	1	1	F_{OSC} / 128
-	Reserved.			
PWMF	PWM Interrupt Flag. This flag bit is cleared by software.			
CLR_P0	Counter Reset Enable. This flag bit is cleared by hardware.			
RUN_P0	Counter Start Enable.			

6.2.6.8 **PWMD** (DEh) : PWM Control Register

Bit No.	7	6	5	4	3	2	1	0
	PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

PWMD duty data register, located in DEh, determines the duty of the output value generated by each 8-bit PWM circuit. To program the required PWM output, you load the appropriate initialization values into the 8-bit data register (PWMD).

6.2.6.9 **ALTSEL** (E3h) : Alternative Function Selection Register

Bit No.	7	6	5	4	3	2	1	0
	IOXEN	IORSTEN	CLO	PWM00	TVO	TX	-	-
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
PWM00	PWM Waveform Output Enable to P0[0]. 0 = Disable the PWM waveform output to P[0]. (Default) 1 = Enable the PWM waveform output to P[0].
-	Reserved.

6.2.7 ADC (Analog to Digital Converter)

The 10-bit A/D converter (ADC) module uses successive approximation logic to convert analog levels entering at one of the twelve input channels to equivalent 10-bit digital values. The analog input level must lie between the V_{DD} (or AV_{REF}) and V_{SS} values.

The A/D converter has the following components:

- Analog comparator with successive approximation logic
- D/A converter logic
- ADC control register (ADCON)
- Twelve multiplexed analog data input pins (ADC0–ADC11)
- 10-bit A/D conversion data output register (ADCR[7:0], ADCON[1:0])

To initiate an analog-to-digital conversion procedure, you write the channel selection data in the A/D converter control register ADCON to select one of the twelve analog input pins (ADCn, n = 0–11) and set the conversion enable and start bit, ADCON[7] and ADCON[6]. The read-write ADCON register is located at address EFh.

During a normal conversion, ADC logic initially sets the successive approximation register to 000h This register is then updated automatically during each conversion step. The successive approximation block performs 10-bit conversions for one input channel at a time. You can dynamically select different channels by manipulating the analog input selection bits in ADCSEL[7:4] and ADCSELH[7:0], and the channel selection bits in ADCSEL[3:0].

To run the A/D conversion, you should set the enable bit (ADCON[7]) and start bit (ADCON[6]). When a conversion is completed, the end-of-conversion (AD_END and ADCF) bits are automatically set to “1”. AD_END bit is ADCON[5] and ADCF (ADCON[4]) is used for interrupt flag. Conversion result is moved into the ADCR[7:0] and ADCON[1:0] register where it can be read. The A/D converter then enters an idle state.

Remember to read the contents of ADCR and ADCON before another conversion starts. Otherwise, the previous result will be overwritten by the next conversion result.

Note : Because the ADC does not use sample-and-hold circuitry, it is important that any fluctuations in the analog level at the ADC0–ADC11 input pins during a conversion procedure be kept to an absolute minimum. Any change in the input level, perhaps due to circuit noise, will invalidate the result.

The ADC module's input pins are alternatively used as digital input in P0[7:1] and P2[6:2].

When ADC channel input pins are used for ADC, they must be configured as analog input. So, you should turn off the pull-up resistor on analog input pins. To turn off the pull-up resistor, set the pull-up resistor control bits that are P0SEL[7:0] and P2SEL[6:2] to “1’.

6.2.7.1 Reference Voltage Levels

In the ADC function block (Figure 6-15), the analog input voltage level is compared to the reference voltage. The default setting for reference voltage level is V_{DD}. And the analog input voltage level must remain within the range of V_{SS} to V_{DD}.

If the analog input voltage level is compared to the ADC reference voltage level (AV_{REF}) from P0[4], you set the bit in ADCON[3] to “1”. The analog input voltage level must remain within the range of V_{SS} to AV_{REF}. Note that AV_{REF} is the voltage level 2V ~ V_{DD}.

Different reference voltage levels are generated internally along the resistor tree during the analog conversion process for each conversion step. The reference voltage level for the first bit conversion is always 1/2 ADC reference voltage level (1/2 V_{DD} or 1/2 AV_{REF}).

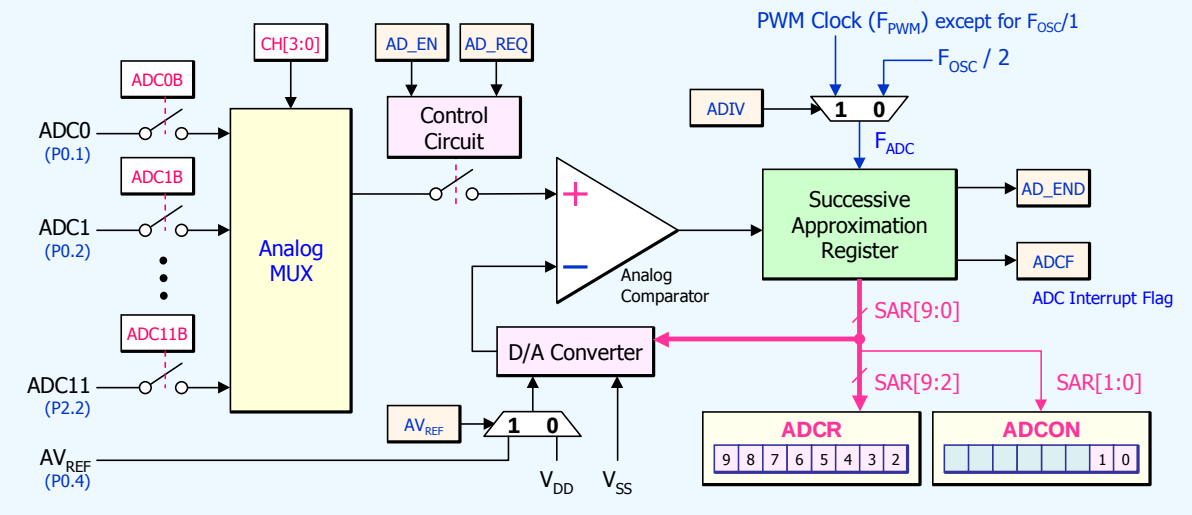


Figure 6-15 ADC Circuit Diagram

6.2.7.2 Conversion Timing

The A/D conversion process requires total 96 clocks.

- Setup time : 8 clocks
- Conversion time : 10 bits x 8 clocks = 80 clocks
- Hold time : 8 clocks

Conversion timing for each frequency is presented in Table 6-10.

Table 6-10 Example of Conversion Timing versus Frequency

F_{OSC}	Divide (ADIV = 0)	F_{ADC}	T_{ADC} ($1/F_{ADC}$)	1 Sample Conversion Time
20 MHz @5V	F_{OSC} / 2	10 MHz	100 ns	9.6 us
10 MHz @5V	F_{OSC} / 2	5 MHz	200 ns	19.2 us
10 MHz @5V	F_{OSC} / 2	5 MHz	200 ns	19.2 us
5 MHz @5V	F_{OSC} / 2	2.5MHz	400 ns	38.4 us

6.2.7.3 Internal A/D Conversion Procedure

Following sequences is the way to run ADC conversion as described in Figure 6-16.

1. Analog input must remain between the range of V_{SS} and reference voltage level (V_{DD} or AV_{REF}).
2. Configure the analog input pins to analog input mode by setting the bits of ADCSEL[7:4] and ADCSELH[7:9] to “1”.
3. Turn off pull-up resistors by setting P0SEL[7:1] and P2SEL[2:6] to “1”.
4. Before the conversion operation starts, you must first select one of the twelve input pins (ADC0–ADC11) by writing the appropriate value to the ADCSEL[3:0].
5. To enable ADC conversion, set ADCON[7] to “1”.
6. To start ADC conversion, set ADCON[6] to “1’.
7. When conversion has been completed, the AD_END will be set to “0” and ADCF flag to “1”..
8. Check AD_END or ADCF flag to verify that the conversion was finished.
9. The converted digital value is loaded to the output register, ADCR[7:0] and ADCON[1:0], then the ADC module enters an idle state.
10. The digital conversion result can now be read from the ADCR and ADCON register.

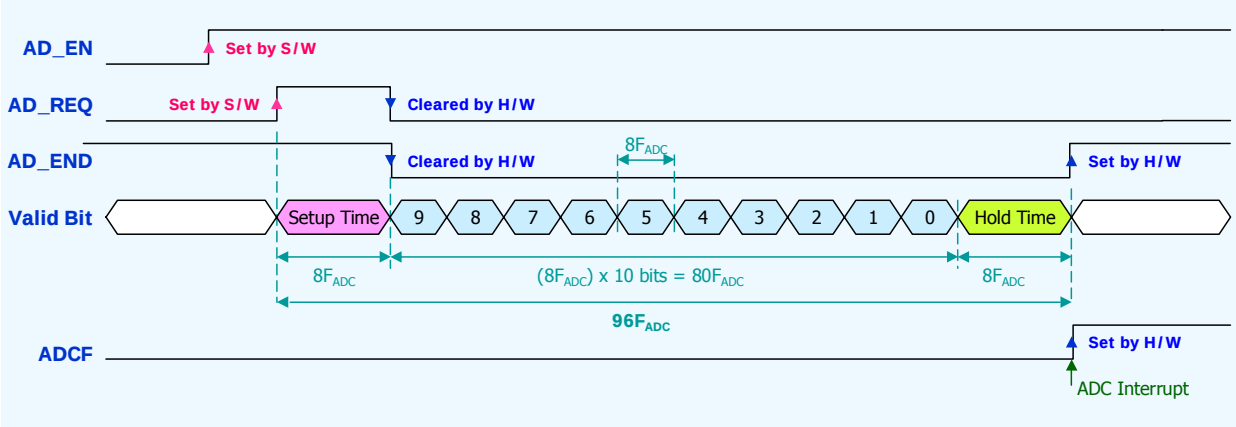


Figure 6-16 A/D Converter Timing Diagram

Recommended A/D converter circuit for more high accuracy is described in Figure 6-17.

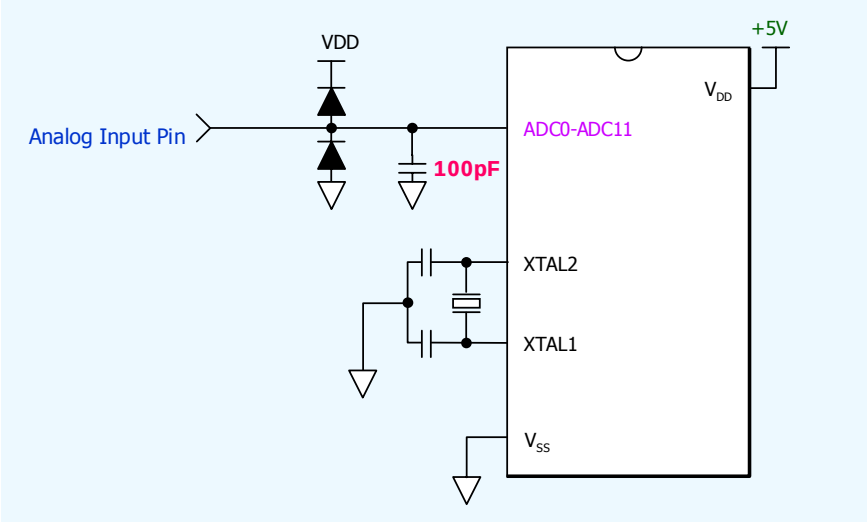


Figure 6-17 Recommended A/D Converter Circuit for more high accuracy

6.2.7.4 **ADCSELH** (E1h) : ADC Channel Selection High Register

Bit No.	7	6	5	4	3	2	1	0
	ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

Symbol	Description	
ADC4B ADC5B ADC6B ADC7B ADC9B ADC10B ADC11B	ADC Channel Selection Low. These bits are configured to enable/disable the ADC channel inputs (ADC[11:4]). P0[5] ~ P0[7] : ADC4 ~ ADC6 input.	
	ADC Channel	Result
	ADC4B	1 = ADC4 input channel is disabled & digital input is enabled at P0[5]. (Default) 0 = ADC4 input channel is enabled & digital input is disabled.
	ADC5B	1 = ADC5 input channel is disabled & digital input is enabled at P0[6]. (Default) 0 = ADC5 input channel is enabled & digital input is disabled.
	ADC6B	1 = ADC6 input channel is disabled & digital input is enabled at P0[7]. (Default) 0 = ADC6 input channel is enabled & digital input is disabled.
	ADC7B	1 = ADC7 input channel is disabled & digital input is enabled at P2[6]. (Default) 0 = ADC7 input channel is enabled & digital input is disabled.
	ADC8B	1 = ADC8 input channel is disabled & digital input is enabled at P2[5]. (Default) 0 = ADC8 input channel is enabled & digital input is disabled.
	ADC9B	1 = ADC9 input channel is disabled & digital input is enabled at P2[4]. (Default) 0 = ADC7 input channel is enabled & digital input is disabled.
	ADC10B	1 = ADC10 input channel is disabled & digital input is enabled at P2[3]. (Default) 0 = ADC10 input channel is enabled & digital input is disabled.
	ADC11B	1 = ADC11 input channel is disabled & digital input is enabled at P2[2]. (Default) 0 = ADC11 input channel is enabled & digital input is disabled.

6.2.7.5 **ADCSEL** (E2h) : ADC Channel Selection Low & MUX Selection Register

Bit No.	7	6	5	4	3	2	1	0
	ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

<table>
<tr><th>Symbol</th><th colspan="6">Description</th></tr>
<tr><td rowspan="6">ADC0B
ADC1B
ADC2B
ADC3B</td><td colspan="6">ADC Channel Selection Low.
These bits are configured to enable/disable the ADC channel inputs (ADC[3:0]). P0[1] ~ P0[4] : ADC0 ~ ADC3 input. (Default = 1)</td></tr>
<tr><td>ADC Channel</td><td colspan="5">Result</td></tr>
<tr><td>ADC0B</td><td colspan="5">1 = ADC0 input channel is disabled & digital input is enabled at P0[1].
0 = ADC0 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC1B</td><td colspan="5">1 = ADC1 input channel is disabled & digital input is enabled at P0[2].
0 = ADC1 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC2B</td><td colspan="5">1 = ADC2 input channel is disabled & digital input is enabled at P0[3].
0 = ADC2 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC3B</td><td colspan="5">1 = ADC3 input channel is disabled & digital input is enabled at P0[4].
0 = ADC3 input channel is enabled & digital input is disabled.</td></tr>
<tr><td rowspan="8">CH0
CH1
CH2
CH3</td><td colspan="6">ADC MUX Selection.
Theses bits is configured to select the ADC input. The initial value of CH[3:0] is 0xFh (4’b1111). Ch, Dh, Eh and Fh means that all ADC inputs are disabled.</td></tr>
<tr><td>CH3</td><td>CH2</td><td>CH1</td><td>CH0</td><td colspan="2">Result</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">0x0h : ADC0 is selected</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td colspan="2">0x1h : ADC1 is selected</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="2">0x2h : ADC2 is selected</td></tr>
<tr><td colspan="4">…</td><td colspan="2"></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td colspan="2">0xAh : ADC10 is selected</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td colspan="2">0xBh : ADC11 is selected</td></tr>
<tr><td colspan="4">1100 (0xCh) ~ 1111 (0xFh)</td><td colspan="2">No ADC channel is not selected (Default Value = 0xFh)</td></tr>
</table>

6.2.7.6 **ADCR** (EEh) : ADC Result High Register

Bit No.	7	6	5	4	3	2	1	0
	SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

The MSB (ADC result value [9:2]) of ADC conversion result (total 10 bits) is stored into ADCR. LSB of conversion result will be stored into ADCON[1:0].

6.2.7.7 **ADCON** (EFh) : ADC Control & ADC Result Low Register

Bit No.	7	6	5	4	3	2	1	0
	AD_EN	AD_REQ	AD_END	ADCF	AVREF	ADIV	SAR1	SAR0
	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
AD_EN	ADC Ready Enable.
AD_REQ	ADC Start Enable. Cleared by H/W when AD_END goes to "1" from "0".
AD_END	Current ADC Status. Set by hardware when ADC conversion has been finished. Note that user should not use the AD_END but ADCF for finish of ADC conversion. 0 = ADC is running now. 1 = ADC is idle state. ADC is stopped. (Default)
ADCF	ADC Interrupt Flag. This bit must be cleared by software.
AVREF	ADC Reference Voltage Input Enable from P0[4]. 0 = ADC reference voltage from V_{DD} (System Power). (Default) 1 = ADC reference voltage from P0[4].
ADIV	ADC Input Clock Select. 0 = System Clock (F_{OSC}) / 2. (Default) 1 = PWM Input Clock (F_{PWM}).
SAR1 SAR0	ADC Result Value [1:0]. (Low bits of ADC result value)

6.2.8 Interrupt

The MiDAS1.1 family has a two priority level interrupt structure with 10 interrupt sources. Each of the interrupt sources has an individual priority bit, flag, interrupt vector and enable bit. In addition, the interrupts can be globally enabled or disabled.

6.2.8.1 Interrupt Sources

The two external interrupts INT0B and INT1B can be either edge triggered or level triggered, depending on bits IT0 and IT1 of the TCON register. The bits IE0 and IE1 in the TCON register are the flags which are checked to generate the interrupt. Another two external interrupts of INT2 and INT3B are only edge triggered type. INT2 is positive edge triggered type but INT3B is negative edge triggered type. In the negative edge triggered mode, the external interrupts are sampled in every machine cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IE0 or IE1 in TCON or IE3 in EXIF[5] is set. In the positive edge triggered mode, a low-to-high transition is detected. The flag bit requests the interrupt. Since the external interrupts are sampled every machine cycle, they have to be held high or low for at least one complete machine cycle. If IEx in TCON is selected to level triggered mode (ITx=0), then the requesting source has to hold the pin low till the interrupt is serviced. The IE0 and IE1 will not be cleared by the hardware on entering the service routine in this mode. If the interrupt continues to be held low even after the service routine is completed, then the processor may acknowledge another interrupt request from the same source.

If IEx in TCON is selected to edge triggered mode (ITx=1), the IEx bit is automatically cleared when the service routine is called. But, note that the external interrupts INT2 and INT3B are edge triggered only. The individual interrupt flag corresponding to external interrupt 2 and 3 must be cleared manually by software.

The Timer 0 and 1 interrupts are generated by the TF0 and TF1 flags. These flags are set by the overflow in the Timer 0 and 1. The TF0 and TF1 flags are automatically cleared by the hardware when the timer interrupt is serviced.

The Watchdog timer can be used as a system monitor or a simple timer. In either case, when the time-out count is reached, the Watchdog timer interrupt flag WDIF (WDCON.3) is set. If the interrupt is enabled by the enable bit EIE.4, then an interrupt occurs.

The UART can generate interrupts on reception or transmission. There are two interrupt sources from the UART, which are obtained by the RI and TI bits in the SCON. These bits are not automatically cleared by the hardware, and the user must clear these bits using software.

PWM block will generate an interrupt and the PWM interrupt flag PWMF is set whenever PWM output signal goes “0” to “1” if PWM interrupt enable bit (EIE.5) is “1’. This PWMF are not automatically cleared by the hardware, and the user must clear these bits using software.

ADC interrupt will be generated when ADC conversion is finished. ADCF flag will be set to “1” and AD_END flag will be set to “1”. ADCF flag must be cleared by S/W in ADC interrupt routine.

All the bits that generate interrupts can be set or reset by hardware, and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing a bit in the IE register. The IE register also has a global enable/disable bit EA which can be cleared to disable all the interrupts.

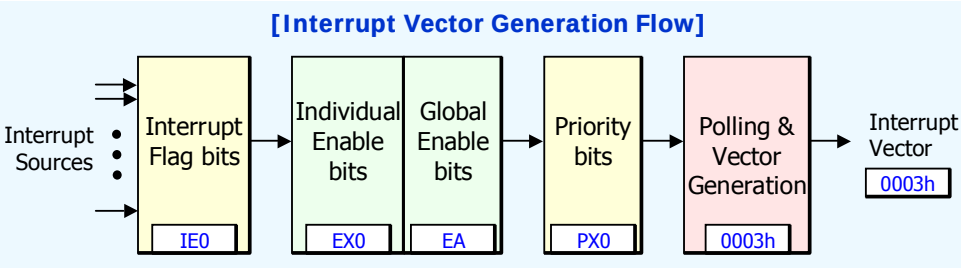


Figure 6-18 Interrupt Vector Generation Flow

6.2.8.2 Priority Level Structure

There are two priority levels for the interrupts. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there exists a pre-defined hierarchy amongst the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as shown below; the interrupts are numbered starting from the highest priority to the lowest.

Table 6-11 Priority Structure of Interrupts

Hierarchy	Sources	Vector Address	Priority Level
1 (Highest)	INT0B	0003h	2 Levels
2	TF0	000Bh	2 Levels
3	INT1	0013h	2 Levels
4	TF1	001Bh	2 Levels
5	RI + TI	0023h	2 Levels
8	ADC	003Bh	2 Levels
9	INT2	0043h	2 Levels
10	INT3B	004Bh	2 Levels
13	WDT	0063h	2 Levels
14 (Lowest)	PWM	006Bh	2 Levels

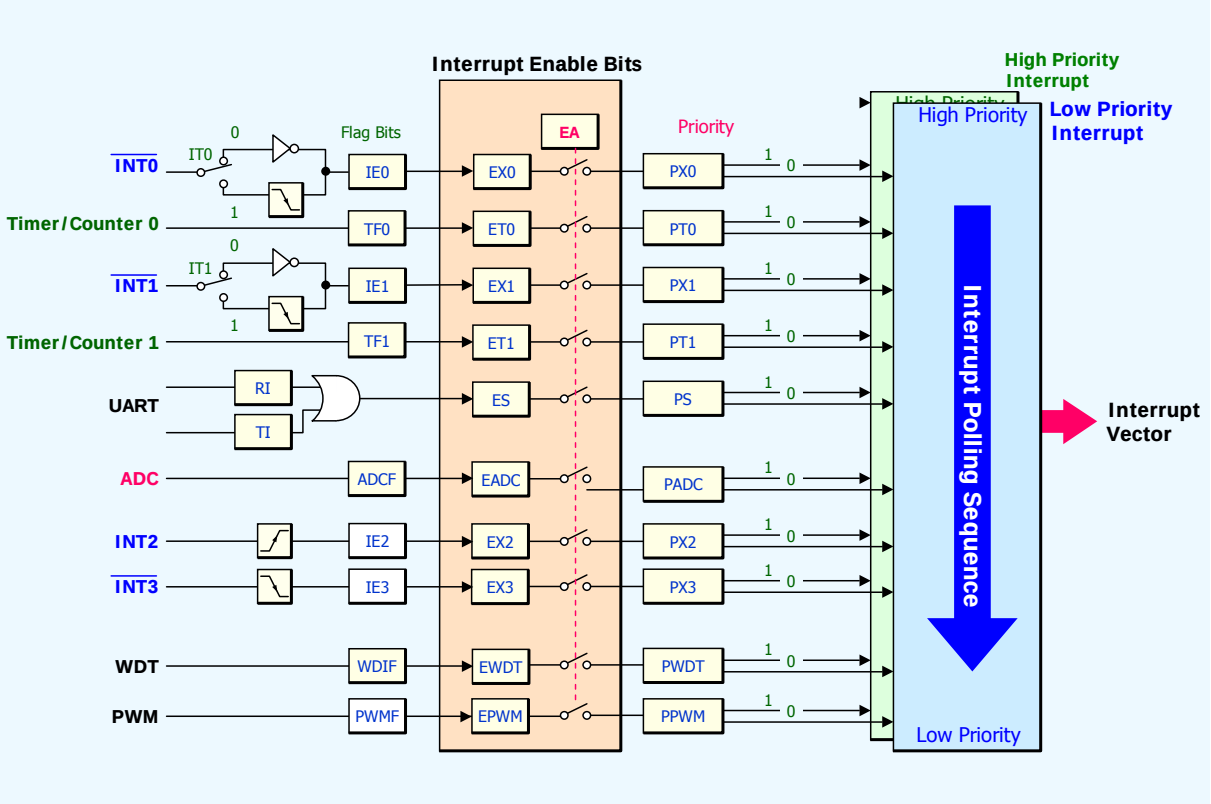


Figure 6-19 Hierarchy of Interrupt Priority

The interrupt flags are sampled every machine cycle. In the same machine cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are

1. An interrupt of equal or higher priority is not currently being services.
2. The current polling cycle is the last machine cycle of the instruction currently being executed.
3. The current instruction does not involve a write to IP, IE, EIP, EIE, IPH or EXIF and is not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every machine cycle, with the interrupts sampled in the same machine cycle. Then note that if an interrupt flag is active but not being responded to for one of the above conditions, and is not still active when the blocking condition is removed, the denied interrupt will not be serviced. This means that active interrupts are not remembered; every polling cycle is new.

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. This may or may not clear the flag which caused the interrupt. In case of timer interrupts, the TF0 or TF1 flags are cleared by hardware whenever the processor vectors to the appropriate timer service routine. In case of external interrupt, INT0B and INT1B, the flags are cleared only if they are edge triggered. In case of UART interrupts, the flags are not cleared by hardware. Watchdog timer interrupt flag WDIF have to be cleared by software. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the stack, but does not save the PSW. The PC is reloaded with the vector address of that interrupt which caused the LCALL.

Execution continues from the vector address until an RETI instruction is executed. On execution of the RETI instruction the processor pops the stack and loads the PC with the contents at the top of the stack. The user must take care that the status of the stack is restored to what is after the hardware LCALL, if the execution is to return to the interrupted program. The processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC. Note that a RET instruction would perform exactly the same process as a RETI instruction, but it would not inform the interrupt controller that the interrupt service routine is completed, and would leave the controller still thinking that the service routine is underway.

6.2.8.3 Interrupt Response Time

The response time for each interrupt source depends on several factors, such as the nature of the interrupt and the instruction underway. In the case of external interrupts INT0B to INT3B, they are sampled at S3 state of every machine cycle and then their corresponding interrupt flags IEx will be set or cleared. The Timer 0 and 1 overflow flags are set at S3 state of the machine cycle in which overflow has occurred. These flag values are polled only in the next machine cycle. If a request is active and all three conditions are met, then the hardware generated LCALL is executed. This LCALL itself takes four machine cycles to be completed. Thus there is a minimum time of five machine cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed. If the polling cycle is not the last machine cycle of the instruction being executed, then an additional delay is introduced. The maximum response time (if no other interrupt is in service) occurs if the MiDAS1.1 family is performing a write to IE, IP, EIE, EIP, IPH or EXIF and then executes a 4-machine cycle instruction. From the time an interrupt source is activated, the longest reaction time is 11 machine cycles. This includes 1 machine cycle to detect the interrupt, 2

machine cycles to complete the IE, IP, EIE, EIP, IPH or EXIF access, 4 machine cycles to complete the instruction and 4 machine cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system, the interrupt response time will always be more than 5 machine cycles and not more than 11 machine cycles. The maximum latency of 11 machine cycle is 44 clock cycles. Note that in the standard 80C52, the maximum latency is 8 machine cycles which equals 96 clock cycles. This is more than 50% reduction in terms of clock periods.

6.2.8.4 **TCON** (88h) : Timer/Counter 0/1 Control Register

Bit No.	7	6	5	4	3	2	1	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
TF1	Timer 1 Overflow Flag. This bit indicates when Timer 1 overflows its maximum count as defined by the current mode. This bit can be cleared by software. 0 = No Timer/Counter 1 overflow has been detected. 1 = Timer/Counter 1 has overflowed with its maximum count
TR1	Timer 1 Run Enable.
TF0	Timer 0 Overflow Flag. This bit indicates when Timer 0 overflows its maximum count as defined by the current mode. This bit can be cleared by software. 0 = No Timer/Counter 0 overflow has been detected. 1 = Timer/Counter 0 has overflowed with its maximum count
TR0	Timer 0 Run Enable.
IE1	External Interrupt 1 Flag. This bit is set when an edge/level type defined by IT1 flag is detected. If IT1=1, this bit will remain set until cleared in software or the start of the External Interrupt 1 service routine. If IT1=0, this bit will inversely reflect the state of the INT1B pin.
IT1	External Interrupt 1 Type Select Flag. This bit selects whether the INT1B pin will detect the edge/level triggered interrupts. 0 = INT1B is level triggered (Level detect type). (Default) 1 = INT1B is edge triggered (Edge detect type).
IE0	External Interrupt 0 Flag. This bit is set when an edge/level type defined by IT0 flag is detected. If IT0=1, this bit will remain set until cleared in software or the start of the External Interrupt 0 service routine. If IT0=0, this bit will inversely reflect the state of the INT0B pin.
IT0	External Interrupt 0 Type Select Flag. This bit selects whether the INT0B pin will detect the edge/level triggered interrupts. 0 = INT0B is level triggered (Level detect type). (Default) 1 = INT0B is edge triggered (Edge detect type).

6.2.8.5 **EXIF** (91h) : External Interrupt Flag Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	IE3	IE2	XT/RG	RGMD	RGSL	BGS
			R/W(0)	R/W(0)	R/W(1)	R(1)	R/W(0)	R/W(1)

Symbol	Description
-	Reserved.
IE3	External Interrupt 3 Flag. This bit will be set when a falling edge is detected on INT3B. This bit must be cleared manually by software. Setting this bit in software will cause an interrupt if enabled.
IE2	External Interrupt 2 Flag. This bit will be set when a rising edge is detected on INT2. This bit must be cleared manually by software. Setting this bit in software will cause an interrupt if enabled.

6.2.8.6 **IE** (A8h) : Interrupt Enable Register

Bit No.	7	6	5	4	3	2	1	0
	EA	EADC	-	ES	ET1	EX1	ET0	EX0
	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
EA	Global Interrupt Enable. This bit controls the global masking of all interrupts. 0 = Disable all individual interrupt masks. (Default) 1 = Enable all individual interrupt masks.
EADC	ADC Interrupt Enable. This bit controls the masking of the ADC interrupt. 0 = Disable the ADC interrupt. (Default) 1 = Enable the ADC interrupt.
-	Reserved.
ES	Serial Port Interrupt Enable. This bit controls the masking of the serial port interrupt. (RI + TI) 0 = Disable the serial port interrupt. (Default) 1 = Enable the serial port interrupt.
ET1	Timer1 Interrupt Enable. This bit controls the masking of the Timer1 interrupt. 0 = Disable the Timer1 interrupt. (Default) 1 = Enable the Timer1 interrupt.
EX1	External Interrupt 1 Enable. This bit controls the masking of the external interrupt 1. 0 = Disable the external interrupt 1. (Default) 1 = Enable the external interrupt 1 (INT1B).
ET0	Timer0 Interrupt Enable. 0 = Disable the Timer0 interrupt. (Default) 1 = Enable the Timer0 interrupt.
EX0	External Interrupt 0 Enable. 0 = Disable the external interrupt 0. (Default) 1 = Enable the external interrupt 0 (INT0B).

6.2.8.7 **IP** (B8h) : Interrupt Priority Register

Bit No.	7	6	5	4	3	2	1	0
	-	PADC	-	PS	PT1	PX1	PT0	PX0
	R(1)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
-	Reserved.
PADC	ADC Interrupt Priority. This bit controls the priority of the ADC interrupt up to two levels. 0 = Priority level 0 for ADC interrupt. (Default) 1 = Priority level 1 for ADC interrupt. Note : The priority level 0 means the low priority interrupt level. And, the priority level 1 means the high priority interrupt level.
PS	Serial Port (UART) Interrupt Priority. This bit controls the priority of the serial port interrupt up to two levels. 0 = Priority level 0 for serial port interrupt. (Default) 1 = Priority level 1 for serial port.
PT1	Timer1 Interrupt Priority. This bit controls the priority of the ADC interrupt up to two levels. 0 = Priority level 0 for timer1 interrupt. (Default) 1 = Priority level 1 for timer1 interrupt.
PX1	External Interrupt 1 Priority. This bit controls the priority of the external interrupt 1 up to two levels. 0 = Priority level 0 for external interrupt 1. (Default) 1 = Priority level 1 for external interrupt 1.
PT0	Timer0 Interrupt Priority. This bit controls the priority of the ADC interrupt up to two levels. 0 = Priority level 0 for timer0 interrupt. (Default) 1 = Priority level 1 for timer0 interrupt.
PX0	External Interrupt 1 Priority. This bit controls the priority of the external interrupt 0 up to two levels. 0 = Priority level 0 for external interrupt 0. (Default) 1 = Priority level 1 for external interrupt 0.

6.2.8.8 **EIE** (E8h) : Extended Interrupt Enable Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	EPWM	EWDT	-	-	EX3	EX2
			R/W(0)	R/W(0)			R/W(0)	R/W(0)

Symbol	Description
-	Reserved.
EPWM	PWM Interrupt Enable. This bit controls the masking of the PWM interrupt. 0 = Disable the PWM interrupt. (Default) 1 = Enable the PWM interrupt.
EWDT	WDT (Watchdog Timer) Enable. This bit controls the masking of the WDT interrupt. 0 = Disable the WDT interrupt. (Default) 1 = Enable the WDT interrupt.
EX3	External Interrupt 3 Enable. This bit controls the masking of the external interrupt 3. 0 = Disable the external interrupt 3. (Default) 1 = Enable the external interrupt 3 (INT3B).
EX2	External Interrupt 2 Enable. This bit controls the masking of the external interrupt 2. 0 = Disable the external interrupt 2. (Default) 1 = Enable the external interrupt 2 (INT2 pin).

6.2.8.9 **EIP** (F8h) : Extended Interrupt Priority Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	PPWM	PWDT	-	-	PX3	PX2
			R/W(0)	R/W(0)			R/W(0)	R/W(0)

Symbol	Description
-	Reserved.
PPWM	PWM Interrupt Priority. This bit controls the priority of the PWM interrupt up to two levels. 0 = Priority level 0 for PWM interrupt. (Default) 1 = Priority level 1 for PWM interrupt. Note : The priority level 0 means the low priority interrupt level. And, the priority level 1 means the high priority interrupt level.
PWDT	WDT (Watchdog Timer) Priority. This bit controls the priority of the WDT up to two levels. 0 = Priority level 0 for WDT interrupt. (Default) 1 = Priority level 1 for WDT interrupt.
PX3	External Interrupt 3 Priority. This bit controls the priority of the external interrupt 3 up to two levels. 0 = Priority level 0 for external interrupt 3. (Default) 1 = Priority level 1 for external interrupt 3.
PX2	External Interrupt 2 Priority. This bit controls the priority of the external interrupt 2 up to two levels. 0 = Priority level 0 for external interrupt 2. (Default) 1 = Priority level 1 for external interrupt 2.

6.2.8.10 **WDCON** (D8h) : Watchdog Control Register

Bit No.	7	6	5	4	3	2	1	0
	WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
	R/W(1)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

<table>
<tr><th>Symbol</th><th colspan="5">Description</th></tr>
<tr><td rowspan="6">WD1
WD0</td><td colspan="5">Watchdog Timer Mode Select</td></tr>
<tr><td>Mode</td><td>WD1</td><td>WD0</td><td>Interrupt Time-Out</td><td>Reset Time-Out</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1 X 2^{16} clocks</td><td>1 X 2^{16} + 256 clocks</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>4 X 2^{16} clocks</td><td>4 X 2^{16} + 256 clocks</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>16 X 2^{16} clocks</td><td>16 X 2^{16} + 256 clocks</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>132X 2^{16} clocks</td><td>32 X 2^{16} + 256 clocks</td></tr>
<tr><td rowspan="7">WDIF</td><td colspan="5">Watchdog Interrupt Flag.
This bit, in conjunction with the Watchdog timer interrupt enable bit EWDT (EIE.4) and Enable Watchdog Timer Reset bit EWT (WDCON.1) indicates if a watchdog timer event has occurred and what action will be taken. This bit must be cleared in software before exiting the interrupt service routine, or another interrupt will be generated. Setting this bit in software will generate a watchdog interrupt if enabled.
(EWT → WDCON.1, EWDT → EIE.4, WDIF → WDCON.3)</td></tr>
<tr><td>EWT</td><td>EWDT</td><td>WDIF</td><td colspan="2">Description</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td colspan="2">No WDT (Watchdog timer) event has occurred.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.
WDT reset will occur after 512 clocks if RWT is not strobed.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.
WDT reset will occur after 512 clocks if RWT is not set.</td></tr>
<tr><td>WTRF</td><td colspan="5">Watchdog Timer Reset Flag.
When set, this bit indicates that a watchdog timer reset has occurred. It is typically interrogated to determine if a reset was caused by watchdog timer reset. It is cleared by a power-on-reset, but otherwise must be cleared by software before the next reset of any kind or software may erroneously determine that a watchdog timer reset has occurred. Setting this bit in software will not generate a watchdog timer reset. If the EWT bit is cleared, the watchdog timer will have no effect on this bit.</td></tr>
</table>

EWT	Enable Watchdog Timer Reset. This bit enables/disables the ability of the watchdog timer to reset the device. This bit has no effect on the ability of the watchdog timer to generate a watchdog interrupt. The time-out period of the watchdog timer is controlled by the Watchdog Timer Mode Select bits (CKCON.7-6). Clearing these bits will disable the ability of the watchdog timer to generate a reset, but have no affect on the timer itself, or its ability to generate a watchdog timer interrupt. 0 = A time-out of the watchdog timer will not cause the device to reset. 1 = A time-out of the watchdog timer will cause the device to reset.
RWT	Restart Watchdog Timer. This bit serves as the strobe for the Watchdog function. During the time-out period, software must set the RWT bit if the Watchdog is enabled. Failing to set the RWT will cause a reset when the time-out has elapsed. There is no need to set the RWT bit to a 0 because it is self-clearing. Note that RWT flag bit must be set to "1" at only WDT mode 0. After watchdog timer is restarted, WDT mode can be changed to mode 3 or any mode.

6.2.8.11 **PWMCON** (DCh) : PWM Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0SEL	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

Symbol	Description			
P0SEL	PWM Waveform Output Enable to P0[6]. 0 = Disable the PWM waveform output to P[6]. (Default) 1 = Enable the PWM waveform output to P[6].			
PS2_P0 PS1_P0 PS0_P0	Prescaled Clock Selection. Note that PWM clock (F_{PWM}) to ADC should not be set to F_{OSC}/1.			
	PS2_P0	PS1_P0	PS0_P0	PWM Clock Rate (F_{PWM})
	0	0	0	F_{OSC} / 1
	0	0	1	F_{OSC} / 2
	0	1	0	F_{OSC} / 4
	0	1	1	F_{OSC} / 8
	1	0	0	F_{OSC} / 16
	1	0	1	F_{OSC} / 32
	1	1	0	F_{OSC} / 64
	1	1	1	F_{OSC} / 128
-	Reserved.			
PWMF	PWM Interrupt Flag. This flag bit is cleared by software.			
CLR_P0	Counter Reset Enable. This flag bit is cleared by hardware.			
RUN_P0	Counter Start Enable.			

6.2.9 Reset Circuit

The user has several hardware related options for placing the MiDAS1.1 family into reset state. In general, most register bits go to their reset values irrespective of the current states, but there are a few flags whose state depends on the source of reset. The user can use these flags to determine the cause of reset using software. There are three ways of putting the device into reset state. They are Power on/fail reset, External reset, and Watchdog reset as shown in Figure 6-20.

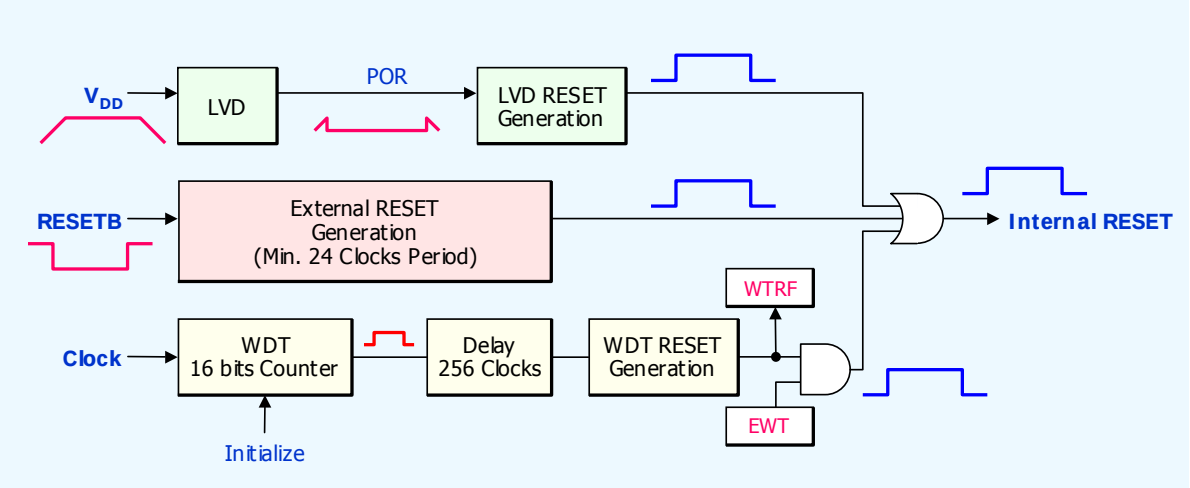


Figure 6-20 Three Reset Resources

6.2.9.1 Power On/Fail Reset

The MiDAS1.1 family incorporates a precision band-gap voltage reference to determine when V_{DD} is out of tolerance. While powering up, internal circuits will hold the device in a reset state until V_{DD} rises above the V_{RST} reset threshold (2.0V). Once V_{DD} is above this level, the oscillator will begin running. An internal reset circuit will then count 65,536 clocks to allow time for power and the oscillator to stabilize. The CPU will then exit the reset state. No external components are needed to generate a power on reset. During power-down or during a severe power glitch, as V_{DD} falls below V_{RST}, the CPU will also generate its own reset. It will hold the reset state as long as power remains below the threshold. This reset will occur automatically, needing no action from the user or from the software.

6.2.9.2 External Reset

The device continuously samples the RESETB pin at state S4 of every machine cycle. Therefore the RESET pin must be held for at least 6 machine cycles (24 clocks) to ensure detection of a valid RESET low. The reset circuitry then synchronously applies the internal reset signal. Thus the reset is a

synchronous operation and requires the clock to be running to cause an external reset.

Once the device is in reset state, it will remain so as long as RESETB is “0”. Even after RESETB is deactivated, the device will continue to be in reset state for up to three machine cycles, and then begin program execution from 0000h. There is no flag associated with the external reset condition. However, since the other two reset sources have flags, the external reset can be considered as the default reset if those two flags are cleared.

6.2.9.3 Watchdog Timer Reset

The Watchdog timer is a free running timer with programmable time-out intervals. The user can clear the watchdog timer at any time, causing it to restart the count. When the time-out interval is reached, an interrupt flag is set. If the Watchdog timer reset is enabled and the Watchdog timer is not cleared, then 256 clocks from the flag being set, the Watchdog timer will generate a reset. This reset condition is maintained by hardware for thirty clock cycles. Once the reset is removed the device will begin execution from 0000h.

6.2.9.4 **WDCON** (D8h) : Watchdog Control Register

Bit No.	7	6	5	4	3	2	1	0
	WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
	R/W(1)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

<table>
<tr><th>Symbol</th><th colspan="5">Description</th></tr>
<tr><td rowspan="6">WD1
WD0</td><td colspan="5">Watchdog Timer Mode Select</td></tr>
<tr><td>Mode</td><td>WD1</td><td>WD0</td><td>Interrupt Time-Out</td><td>Reset Time-Out</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1 X 2^{16} clocks</td><td>1 X 2^{16} + 256 clocks</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>4 X 2^{16} clocks</td><td>4 X 2^{16} + 256 clocks</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>16 X 2^{16} clocks</td><td>16 X 2^{16} + 256 clocks</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>132X 2^{16} clocks</td><td>32 X 2^{16} + 256 clocks</td></tr>
<tr><td rowspan="7">WDIF</td><td colspan="5">Watchdog Interrupt Flag.
This bit, in conjunction with the Watchdog timer interrupt enable bit EWDT (EIE.4) and Enable Watchdog Timer Reset bit EWT (WDCON.1) indicates if a watchdog timer event has occurred and what action will be taken. This bit must be cleared in software before exiting the interrupt service routine, or another interrupt will be generated. Setting this bit in software will generate a watchdog interrupt if enabled.
(EWT → WDCON.1, EWDT → EIE.4, WDIF → WDCON.3)</td></tr>
<tr><td>EWT</td><td>EWDT</td><td>WDIF</td><td colspan="2">Description</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td colspan="2">No WDT (Watchdog timer) event has occurred.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.
WDT reset will occur after 512 clocks if RWT is not strobed.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.
WDT reset will occur after 512 clocks if RWT is not set.</td></tr>
<tr><td>WTRF</td><td colspan="5">Watchdog Timer Reset Flag.
When set, this bit indicates that a watchdog timer reset has occurred. It is typically interrogated to determine if a reset was caused by watchdog timer reset. It is cleared by a power-on-reset, but otherwise must be cleared by software before the next reset of any kind or software may erroneously determine that a watchdog timer reset has occurred. Setting this bit in software will not generate a watchdog timer reset. If the EWT bit is cleared, the watchdog timer will have no effect on this bit.</td></tr>
</table>

EWT	Enable Watchdog Timer Reset. This bit enables/disables the ability of the watchdog timer to reset the device. This bit has no effect on the ability of the watchdog timer to generate a watchdog interrupt. The time-out period of the watchdog timer is controlled by the Watchdog Timer Mode Select bits (CKCON.7-6). Clearing these bits will disable the ability of the watchdog timer to generate a reset, but have no affect on the timer itself, or its ability to generate a watchdog timer interrupt. 0 = A time-out of the watchdog timer will not cause the device to reset. 1 = A time-out of the watchdog timer will cause the device to reset.
RWT	Restart Watchdog Timer. This bit serves as the strobe for the Watchdog function. During the time-out period, software must set the RWT bit if the Watchdog is enabled. Failing to set the RWT will cause a reset when the time-out has elapsed. There is no need to set the RWT bit to a 0 because it is self-clearing. Note that RWT flag bit must be set to "1" at only WDT mode 0. After watchdog timer is restarted, WDT mode can be changed to mode 3 or any mode.

6.2.10 Clock Circuit (On-chip oscillators)

The MiDAS1.1 family supports three clock configurations as shown in Figure 6-21: 1) crystal oscillator (only A0 version), 2) oscillator module and 3) internal ring oscillator.

The MiDAS1.1 family has the on-chip oscillator circuitry which consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator. The system clock of the MiDAS 1.1 family is Ring oscillator by default. User can turn off its oscillator under software control by writing a 1 to the PD bit in PCON as shown in Figure 6-22.

The value of the load capacitor of crystal oscillator is dependent on the operating frequency as shown in Figure 6-23. Table 6-12 shows the recommended load capacitor in the various frequency.

Table 6-12 Recommended Load Capacitor in Crystal Oscillator (V_{DD} = 5V)

	Crystal Oscillator [MHz]	
	~ 11.0592	20.0000
Load Capacitor (C_{LOAD})	47 pF	20 pF

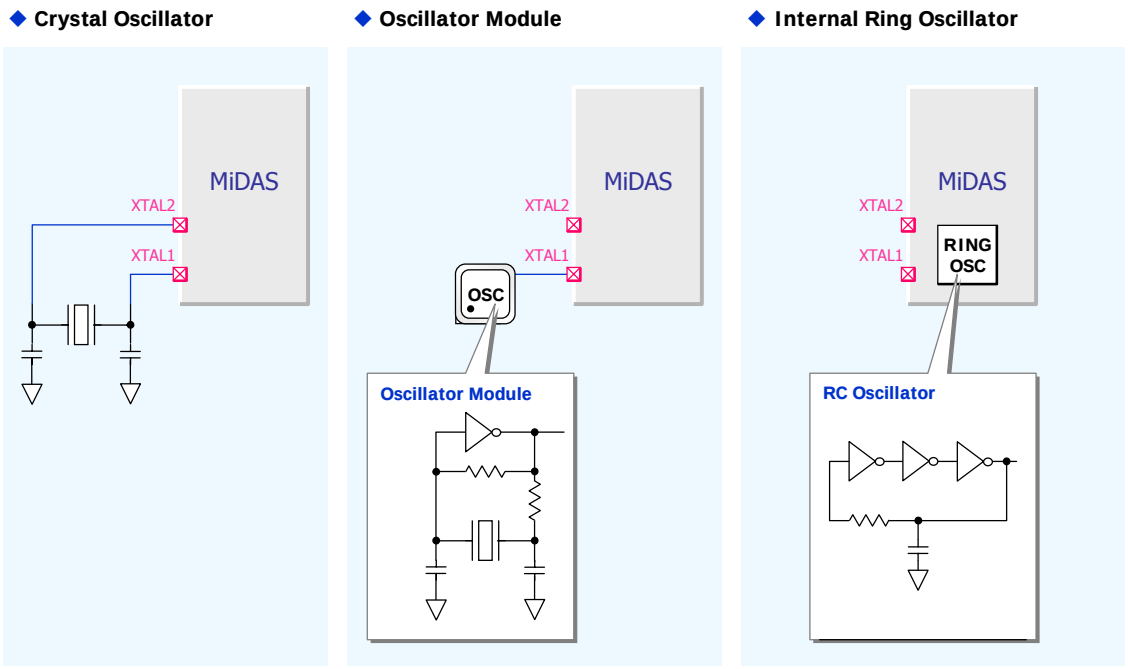


Figure 6-21 Configuration for Clock

To drive the device with an external clock source, apply the external clock signal to XTAL1 and leave XTAL2 float as shown in Figure 6-22.

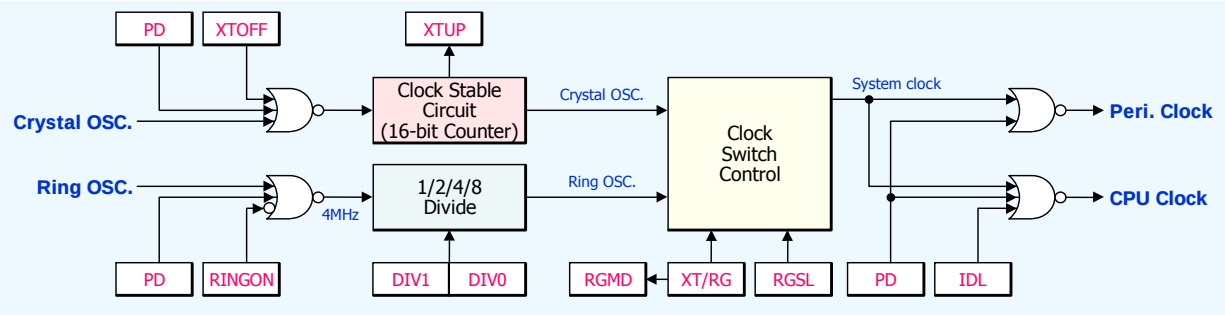


Figure 6-22 Clock Circuit

Table 6-13 System Clock Configuration

Control Flag				System Clock (F_{osc})	Status Bits	
XT/RG	XTOFF	RINGON	TRGSL		RGMD	XTUP
1	0	X	X	Crystal Oscillator	0	1
0	X	1	X	Ring Oscillator	1	0 / 1
1	0	X	0	Crystal Oscillator (During power-down wake-up)	0	0
0	X	1	1	Ring Oscillator (During power-down wake-up)	1	0

An external oscillator may encounter as much as a 100pF load at XTAL1 when it starts up. This is due to interaction between the amplifier and its feedback capacitance. Once the external signal meets the V_{IL} and V_{IH} specifications the capacitance will not exceed 20pF.

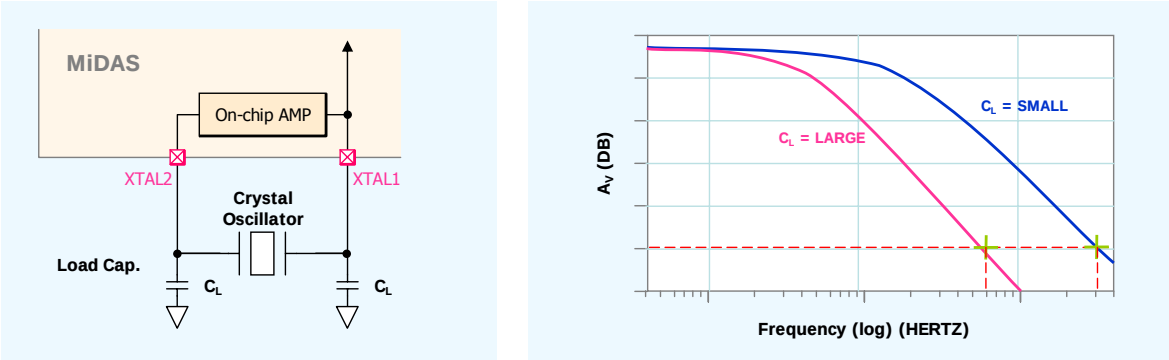


Figure 6-23 The Load Capacitor versus Operating Frequency

6.2.10.1 **EXIF** (91h) : External Interrupt Flag Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	IE3	IE2	XT/RG	RGMD	RGSL	BGS
			R/W(0)	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(1)

Symbol	Description
-	Reserved.
XT/RG	System Clock Selection. This bit is set to "0" after a power-on reset, and unchanged by all other resets. Note : Don't clear RINGON (OSCICN.2) flag when F_{OSC} is Ring oscillator (XT/RG = 0). 0 = The Ring oscillator is selected as the system clock. (Default) 1 = The crystal oscillator is selected as the system clock.
RGMD	Ring Mode. Now system clock is Ring or crystal oscillator Generally RGMD is the invert of XT/RG.
RGSL	Ring Select Bit when Power-down Wake-up. 1 = When wake-up from Power-down mode in XTAL clock, the CPU uses Ring oscillator as system clock during 65,536 XTAL clock. (Default = 0)
BGS	Band-Gap Select. This bit enables/disables the band-gap reference during power-down mode. Disabling the band-gap reference provides significant power savings in power-down mode, but sacrifices the ability to perform power-fail reset while stopped. The state of this bit will be undefined on devices which do not incorporate a band-gap reference. 0 = The band-gap reference is disabled in power-down mode, but will function during normal operation. It will support the significant power savings in power-down mode. 1 = The band-gap reference will operate in Stop mode. (Default)

6.2.10.2 **OSCICN** (BEh) : Internal Ring Oscillator Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	RINGON	DIV1	DIV0
						R/W(1)	R/W(0)	R/W(0)

Symbol	Description			
-	Reserved.			
RINGON	Internal Ring Oscillator Enable. Note : Don't clear this bit when F_{OSC} is Ring oscillator (XT/RG = 0). But, although F_{OSC} is crystal oscillator, internal ring oscillator can be run (RINGON=1). 0 = Internal Ring Oscillator is killed. (Default) 1 = Internal Ring oscillator is running.			
DIV1 DIV0	Ring Oscillator Divider			
	Mode	DIV1	DIV0	Description (Ring Oscillator @ 5V)
	0	0	0	4 MHz / 1
	1	0	1	4 MHz / 2
	2	1	0	4 MHz / 4
	3	1	1	4 MHz / 8

6.2.10.3 **STATUS** (C5h) : Crystal Status Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	XTUP	-	-	-	-
				R(0)				

Symbol	Description
-	Reserved.
XTUP	Crystal Oscillator Warm-up Status. This bit indicates whether the CPU crystal oscillator has completed the 65,536 cycle warm-up, and represents the system clock of the crystal oscillator is stable(1) or not(0). This bit is cleared by H/W when Power-on Reset, during Power-down wake-up. This bit is set to 1 following a crystal stabilization by H/W.

6.2.10.3.1 **PMR** (C4h) : Power Management Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	XTOFF	-	-	-
					R/W(0)			

Symbol	Description
XTOFF	Internal Amplifier Disable for External Crystal Oscillator. This bit controls the enable/disable the internal amplifier for external crystal oscillator. 0 = Internal amplifier is enabled. (Default) 1 = Internal amplifier is disabled. Set the XTOFF bit to "1" when user wants to use P1[0] and P1[2] as general I/O pins. Don't set XTOFF bit to "1" when XT/RG bit (EXIF.3) is "1". (XT/RG = "1" : The external clock is selected as system clock. Refer to EXIF register.)

6.2.10.4 PCON (87h) : Power Control Register

Bit No.	7	6	5	4	3	2	1	0
	SMOD1	-	-	POF	GF1	GF0	PD	IDL
	R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
-	Reserved
PD	Power-down (Stop) Mode Enable. Setting this bi causes the MCU (MiDAS1.1 family) to go into the power-down mode. In this mode, all the clocks are stopped and program execution is frozen. 0 (Default), 1 = Power-down mode enable.
IDL	Idle Mode Enable. Setting this bit causes the MiDAS1.1 family to go into the IDLE mode. In this mode the clocks to the CPU are stopped, so program execution is frozen. But the clock to the peripherals and interrupt blocks is not stopped, and these blocks continue operating. 0 (Default), 1 = Idle mode enable.

6.2.11 Power Management

The MiDAS1.1 family has two power saving features (Power-down mode and IDLE mode) that help the user to modify the power consumption of the device.

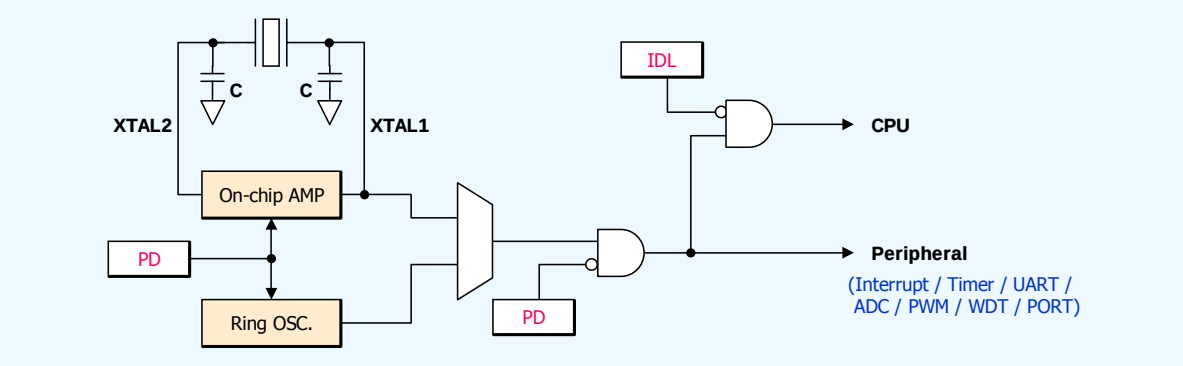


Figure 6-24 Power Management Circuit

6.2.11.1 IDLE Mode

The user can put the device into idle mode by writing 1 to the bit PCON.0. The instruction that sets the idle bit is the last instruction that will be executed before the device goes into IDLE mode, the clock to the CPU is halted, but not to the interrupt and peripherals such as Timer, Watchdog timer and serial port blocks. This forces the CPU state to be frozen; the Program counter, the Stat Pointer, the Program Status Word, the Accumulator and the other registers hold their contents. The ALE and /PSEN pins are held high during the IDLE state. The port pins hold the logical states they had at the time IDLE was activated. The IDLE mode can be terminated in two ways. Since the interrupt controller is still active, the activation of any enabled interrupt can wake up the processor. This will automatically clear the IDL bit, terminate the IDLE mode, and the Interrupt Service Routine (ISR) will be executed. After the ISR, execution of the program will continue from the instruction which put the device into IDLE mode.

The IDLE mode can also be exited by activating the reset. The device can be put into reset either by applying a high on the external RESETB pin, a Power-On-Rest condition or a Watchdog timer reset. The external reset pin has to be held thigh for at least six machine cycles i.e. 24 clock periods to be recognized as a valid reset. At the reset condition the program counter is reset to 0000h and all the SFRs are set to the reset condition. Since the clock is already running there is no delay and execution starts immediately. In the IDLE mode, the Watchdog timer continues to run, and if enabled, a time-out will cause a watchdog timer interrupt which will wake up the device. The software must reset the Watchdog timer in order to preempt the reset which will occur after 512 clock periods of the time-out. When the MiDAS1.1 family is exiting from an IDLE mode with a reset, the instruction will start from address 0000h. So there is no danger of unexpected writings.

6.2.11.2 Power-down Mode

The device can be put into Power-down mode by writing 1 to bit PCON.1. The instruction that does this will be the last instruction to be executed before the device goes into Power-down mode. In the Power-down mode, all the clocks are stopped and the device comes to a halt. All activity is completely stopped and the power consumption is reduced to the lowest possible value. In this state, ALE and /PSEN pins are pulled low. The port pins output the values held by their respective SFRs.

The MiDAS1.1 family will exit the Power-down mode with a WDT interrupt, RESETB, or by the two external interrupt pin, INT0B, and INT1B, enabled as level detect (0 or 1). An external RESETB can be used to exit the Power-down state. The high on RESETB pin terminates the Power-down mode, and restarts the clock. The program execution will restart from 0000H. In the Power-down mode, the clock is stopped, so the Watchdog timer cannot be used to provide the reset to exit Power-down mode.

The Power-down mode has two stop mode by WDT ON/OFF. And the stop mode is that both CPU and peripherals are not running.

In the stop mode 1(WDT is OFF), the MiDAS1.1 family can be woken from the Power-down mode by forcing an external interrupt pin activated, provided the corresponding interrupt is enabled, while the global enable (EA) bit is set and the external input has been set to a level detect mode. If these conditions are met, then the low level on the external pin restarts the oscillator. User should hold external interrupt pin to “0” for at least 65,536clocks. Because of crystal stabilization time, 65,536 clocks must be needed to wake-up from Power-down mode. After 65,536 clocks, system clock will be running normally and interrupt logic will catch the external interrupt. Then device executes the interrupt service routine for the corresponding external interrupt. After the interrupt service routine is completed, the program execution returns to the instruction after the one which put the device into Power-down mode and continues from there.

In stop mode 2 (WDT is ON), The MiDAS1.1 family can be woken from the Power-down mode by wake-up from WDT interrupt or RESETB.

Table 6-14 Status of External Pins during IDLE and Power-down Modes

Mode	Program Memory	ALE	PSENB	Port 0	Port 1	Port 2	Port 3	Port 4
IDLE	Internal	H	H	Data	Data	Data	Data	Data
	External	H	H	Float	Data	Address	Data	Data
Power-down	Internal	L	L	Data	Data	Data	Data	Data
	External	L	L	Float	Data	Data	Data	Data

6.2.11.3 **PCON** (87h) : Power Control Register

Bit No.	7	6	5	4	3	2	1	0
	SMOD1	-	-	POF	GF1	GF0	PD	IDL
	R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
-	Reserved
PD	Power-down (Stop) Mode Enable. Setting this bi causes the MCU (MiDAS1.1 family) to go into the power-down mode. In this mode, all the clocks are stopped and program execution is frozen. 0 (Default), 1 = Power-down mode enable.
IDL	Idle Mode Enable. Setting this bit causes the MiDAS1.1 family to go into the IDLE mode. In this mode the clocks to the CPU are stopped, so program execution is frozen. But the clock to the peripherals and interrupt blocks is not stopped, and these blocks continue operating. 0 (Default), 1 = Idle mode enable.

7 Absolute Maximum Ratings

Table 7-1 Absolute Maximum Ratings

Items	Conditions	Ranges
Voltage on any pin relative to Ground		-0.5V to (V_{DD} + 0.5V)
Voltage in VDD relative to Ground		-0.5V to 6.5V
Output Voltage		-0.5V to (V_{DD} + 0.5V)
Output Current High	One I/O pin active	-25 mA
	All I/O pin active	-100 mA
Output Current Low	One I/O pin active	+30 mA
	All I/O pin active	+150 mA
Operating Temperature		-40°C to + 85°C
Storage Temperature		-65°C to + 150°C
Soldering Temperature		160°C for 10 seconds

8 DC Characteristics

8.1 General DC Characteristics

Table 8-1 General DC Characteristics

(T_A = -20°C ~ +85°C, V_{DD}=2.4V ~ 5.5V unless otherwise specified)

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input Low Voltage	V_{IL1}	P0, P2	V_{DD} = 2.4V ~ 5.5V	-0.5	-	$0.2V_{DD}$-0.1	V
	V_{IL2}	XTAL1, XTAL2, RESETB		-0.5	-	$0.3V_{DD}$	V
Input High Voltage	V_{IH1}	P0, P2	V_{DD} = 2.4V ~ 5.5V	$0.2V_{DD}$+1.0	-	V_{DD}+0.5	V
	V_{IH2}	XTAL1, XTAL2, RESETB		$0.7V_{DD}$	-	V_{DD}+0.5	V
Output Low Voltage	V_{OL1}	XTAL1, XTLA2, P0, P2	I_{OL}=20mA @V_{DD}=5V (I_{OL}=5mA @V_{DD}=2.6V)	-	-	$0.3V_{DD}$	V
	V_{OL2}	RESETB	I_{OL}=10mA @V_{DD}=5V (I_{OL}=2.5mA @V_{DD}=2.6V)	-	-	$0.3V_{DD}$	V
Output High Voltage	V_{OH}	XTAL, XTAL2, P0, P2	I_{OL}=-15mA @V_{DD}=5V (I_{OL}=2.5mA @V_{DD}=2.6V)	$0.7V_{DD}$	-	-	V
	V_{OH1}	P0, P2 (Pull-up R Only)	I_{OL}=-140uA @V_{DD}=5V (I_{OL}=-20uA @V_{DD}=2.6V)	$0.7V_{DD}$	-	-	V
	V_{OH2}	XTAL1, XTAL2 (Pull-up R Only)	I_{OL}=-10uA @V_{DD}=5V (I_{OL}=1.5mA @V_{DD}=2.6V)	$0.7V_{DD}$	-	-	V
Input Leakage Current	I_{IL}	All pins except XTAL1 and XTAL2	V_{IN} = V_{IH} or V_{IL}	-	-	±1.0	uA
Pin Capacitance	C_{IO}	All pins	V_{DD} = 5V	-	10	-	pF

8.2 ADC Specifications

Table 8-2 ADC Specifications

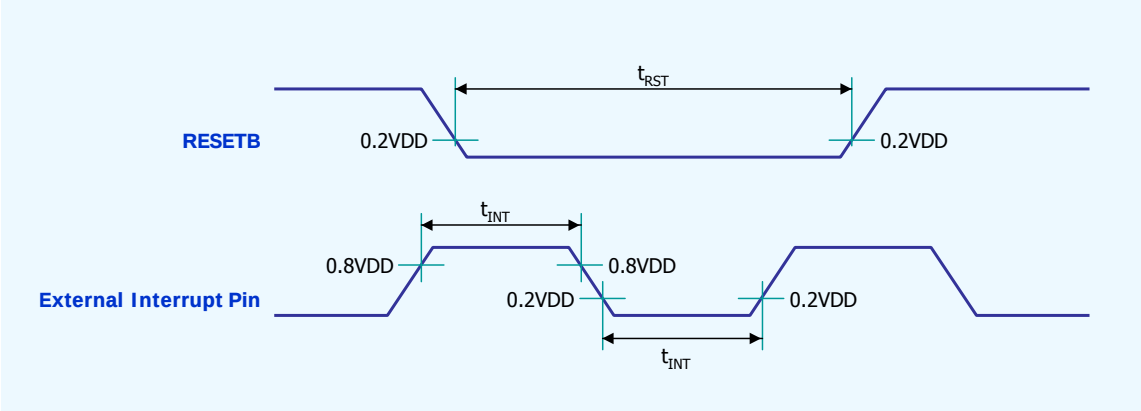
Parameter		Symbol	Conditions	Value			Unit
				Min.	Typ.	Max.	
Supply Voltage		V_{DDADC}	-	2.4	-	5.5	V
Input Voltage		V_{INADC}	-	V_{SS}	-	V_{DD}	V
Resolution		RES_{ADC}	-	-	10	-	Bit
Operating Frequency		F_{ADC}	V_{DD} = 4.5V ~ 5.5V V_{DD} = 2.4 ~ 3.3V	-	-	10 5	MHz
Conversion Time		t_{ADC}	-	-	96/F_{ADC}	-	sec
Overall Accuracy		OA_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz V_{DD} = 3V, F_{ADC} = 5MHz	-	±2.0	±4.0	LSB
Integral Nonlinearity		INL_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz V_{DD} = 3V, F_{ADC} = 5MHz	-	±2.0	±4.0	LSB
Differential Nonlinearity		DNL_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz V_{DD} = 3V, F_{ADC} = 5MHz	-	±0.5	±1.0	LSB
Zero Input Error		ZIE_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz V_{DD} = 3V, F_{ADC} = 5MHz	-	±2.0	±4.0	LSB
Full Scale Error		FSE_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz V_{DD} = 3V, F_{ADC} = 5MHz	-	±2.0	±4.0	LSB
Analog Input Capacitance		C_{INADC}	-	-	10	15	pF
ADC Current	Active	I_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz	-	1.0	2.0	mA
			V_{DD} = 3V, F_{ADC} = 5MHz	-	0.3	0.6	mA
	Power-down		V_{DD} = 5V	-	-	100	nA

9 AC Characteristics

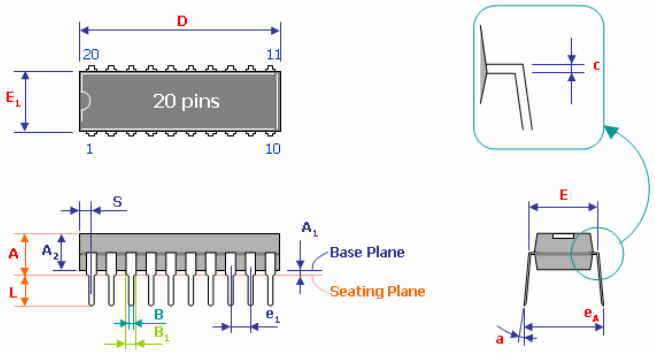
Table 9-1 AC Characteristics

(T_A = -20°C ~ +85°C unless otherwise specified)

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Operating Frequency	FOSC	XTAL1, XAL2	V_{DD} = 5V ±10%	1	-	20	MHz
			V_{DD} = 3V ±10%	1	-	10	
RESETB Input Width	t_{RST}	RESETB	V_{DD} = 5V ±10%	24	-	-	F_{OSC}
			V_{DD} = 3V ±10%	24	-	-	
External interrupt Input Width	t_{INT}	External Interrupt	V_{DD} = 5V ±10%	4	-	-	F_{OSC}
			V_{DD} = 3V ±10%	4	-	-	



10 Package Dimension

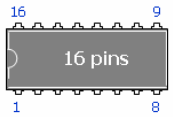


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	-	-	0.200	-	-	5.080
A_1	0.015	-	-	0.381	-	-
A_2	0.150	0.155	0.160	3.810	3.937	4.064
B	0.016	0.018	0.022	0.406	0.457	0.559
B_1	0.045	0.055	0.065	1.143	1.397	1.651
c	0.008	0.010	0.012	0.203	0.254	0.356
D	1.045	1.055	1.075	26.543	26.797	27.305
E	0.290	0.300	0.310	7.366	7.62	7.874
E_1	0.530	0.545	0.550	13.720	13.840	13.970
e_1	0.090	0.100	0.110	2.286	2.540	2.794
L	0.120	0.130	0.140	3.048	3.302	3.556
a	0°	-	15°	0°	-	15°
e_A	0.330	0.350	0.370	8.382	8.89	9.398
S	-	-	0.090	-	-	2.286

Notes:

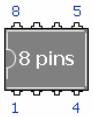
1. Dimension D Max. & S include mold flash or tie bar Burns.
2. Dimension E_1 dose not include interlead flash.
3. Dimension D & E_1 include mold mismatch and are determined at the mold parting line.
4. Dimension B_1 does not include dambar protrusion/intrusion.
5. General appearance spec. should be based on final visual inspection spec.

Figure 10-1 SPDIP 20-pin Package Dimension



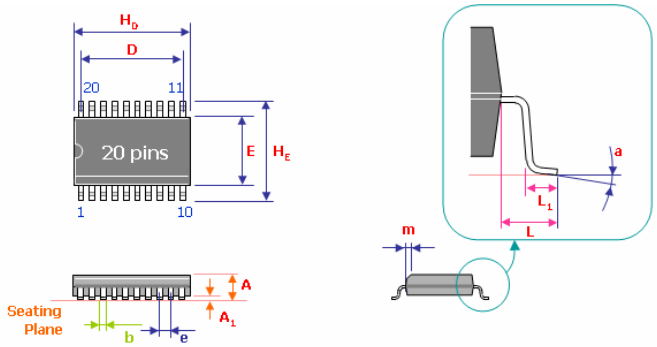
Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
D	0.845	0.855	0.875	21.463	21.717	22.225

Figure 10-2 SPDIP-16pin Package Dimension



Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
D	0.445	0.455	0.475	11.303	11.557	12.065

Figure 10-3 SPDIP-8pin Package Dimension

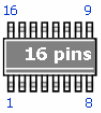


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.093	0.099	0.104	2.35	2.45	2.65
A_1	0.004	0.008	0.012	0.10	0.20	0.30
b	0.014	0.016	0.019	0.35	0.42	0.49
D	-	0.450	-	-	11.43	-
E	0.291	0.295	0.299	7.40	7.50	7.60
H_D	0.496	0.504	0.512	12.60	12.80	13.00
H_E	0.404	0.411	0.419	10.26	10.45	10.65
L	0.057	0.058	0.060	1.43	1.48	1.53
L_1	0.034	0.038	0.042	0.86	0.96	1.07
a	0°	-	8°	0°	-	8°
e	0.050 BSC			1.27 BSC		
m	0.020	0.025	0.030	0.50	0.62	0.75

Notes:

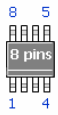
1. Dimension D Max. & S include mold flash or tie bar Burns.
2. Dimension E_1 dose not include interlead flash.
3. Dimension D & E_1 include mold mismatch and are determined at the mold parting line.
4. Dimension B_1 does not include dambar protrusion/intrusion.
5. General appearance spec. should be based on final visual inspection spec.

Figure 10-4 SOIC 20-pin Package Dimension



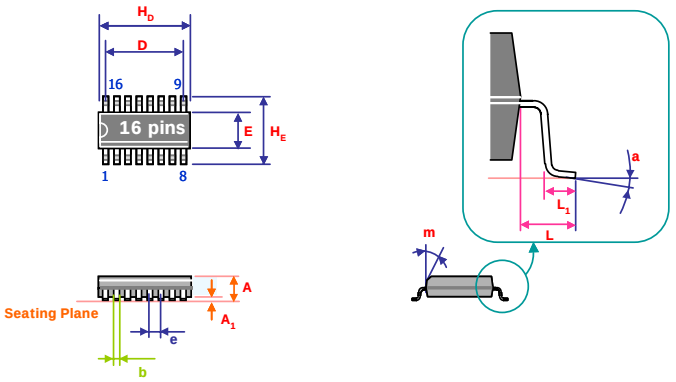
Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
D	-	.350	-	-	8.89	-
E	0.150	0.153	0.157	3.80	3.90	4.00
H_D	0.398	0.405	0.413	10.10	10.29	10.50
H_E	0.234	0.239	0.244	5.95	6.07	6.20
L	0.038	0.043	0.048	0.97	1.08	1.2
L_1	0.022	0.027	0.032	0.58	0.70	0.82
m	0.010	0.015	0.020	0.25	0.37	0.50

Figure 10-5 SOIC-16pin Package Dimension



Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
D	-	0.150	-	-	3.81	-
H_D	0.189	0.193	0.197	4.80	4.90	5.00

Figure 10-6 SOIC-8pin Package Dimension



[16-TSSOP]

Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.037	0.039	0.041	0.95	1.00	1.05
A_1	0.015	0.017	0.019	0.3865	0.4365	0.4865
b	0.008	0.009	0.009	0.20	0.22	0.24
D	0.176	0.179	0.182	4.47	4.55	4.63
E	0.171	0.173	0.175	4.35	4.4	4.45
H_D	0.200	0.202	0.204	5.077	5.127	5.177
H_E	0.248	0.252	0.248	6.30	6.40	6.30
L	0.033	0.037	0.041	0.85	0.95	1.05
L_1	0.020	0.024	0.028	0.50	0.60	0.70
a	1°	3°	5°	1°	3°	5°
e	0.026 BSC			0.65 BSC		
m	10°	12°	14°	10°	12°	14°

Notes:
1. Dimension D & E include mold mismatch and are determined at the mold parting line.
2. General appearance spec. should be based on final visual inspection spec.

Figure 10-7 TSSOP 16-pin Package Dimension

11 Product Numbering System

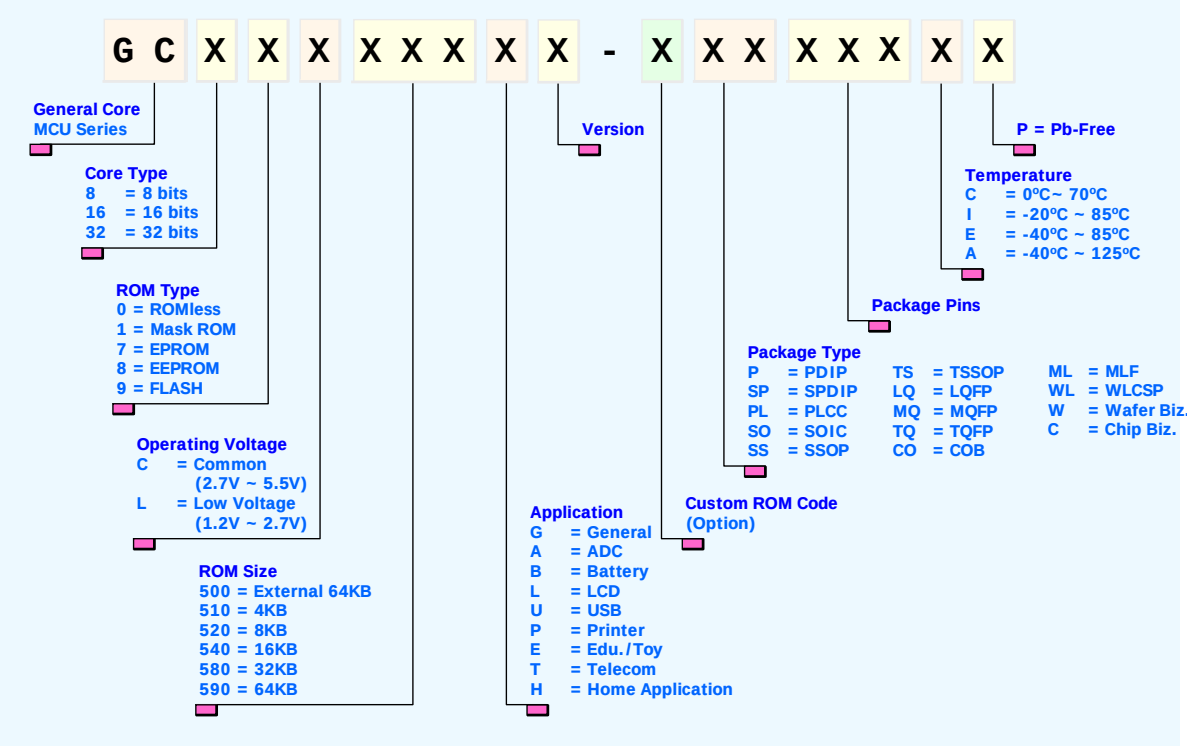


Figure 11-1 Product Numbering System

12 Appendix A: Instruction Set

Table 12-1 Note on Instruction Set and Addressing Modes

Notation	Description
Rn	Register R0 ~ R7 of the currently selected Register Bank (RB0 ~ RB3).
direct	The address of 8-bit internal data location. This could be an **IRAM** location (0x00 ~ 0x7F; 128 bytes) or a **SFR** (0x80 ~ 0xFF).
@Ri	8-bit IRAM location (0x00 ~ 0xFF; 256 bytes) addressed indirectly through register R0 or R1.
#data	8-bit constant included in instruction.
#data16	16-bit constant included in instruction.
addr16	16-bit destination address. Used by **LCALL** & **LJMP**. The branch can be anywhere within the 64 kbytes program memory address space.
addr11	11-bit destination address. Used by **ACALL** & **AJMP**. The branch will be within the same 2 kbytes page of program memory as the first byte of the following instruction.
rel	Signed (2's complement number) 8-bit offset byte. Used by **SJMP** and all conditional jumps. Range is -128 to +127 byte relative to first byte of the following instruction.
bit	Direct Addressed bit n IRAM of SFR.

ACALL addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The called subroutine must therefore start within the same 2K block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07h. The label “SUBRTN” is at program memory location 0345h. After executing the instruction,

ACALL SUBRTN

at location 0123h, SP will contain 09h, internal RAM locations 08h and 09h will contain 25h and 01h, respectively, and the PC will contain 0345h.

Bytes: 2

Cycles: 3

Encoding:

a10 a9 a8 1	0 0 0 1	a7 a6 a5 a4	a3 a2 a1 a0

Operation:
ACALL
(PC) ← (PC)+ 2
(SP) ← (SP) + 1
((SP)) ← (PC_{7-0})
(SP) ← (SP) + 1
((SP)) ← (PC_{15-8})
(PC_{10-0}) ← page address

ADD A, <src-byte>

Function: Add

Description: ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3h (11000011b) and register 0 holds 0AAh (10101010b). The instruction,

ADD A, R0

will leave 6Dh (01101101b) in the Accumulator with the AC flag cleared and both the carry flag and OV set to 1.

ADD A, Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	1	0	1	r	r	r

Operation:
ADD
(A) ← (A) + (Rn)

ADD A, direct

Bytes: 2

Cycles: 2

Encoding:

0	0	1	0	0	1	0	1	direct address

Operation:
ADD
(A) ← (A) + (direct)

ADD A, @Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	1	0	0	1	1	i

Operation:
ADD
(A) ← (A) + ((Ri))

ADD A, #data

Bytes: 2

Cycles: 2

Encoding:

0	0	1	0	0	1	0	0	immediate data

Operation:
ADD
(A) ← (A) + #data

ADDC A, <src-byte>

Function: Add with Carry

Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary–carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3h (11000011b) and register 0 holds 0AAh (10101010b) with the carry flag set. The instruction,

ADDC A, R0

will leave 6Eh (01101110b) in the Accumulator with AC cleared and both the carry flag and OV set to 1.

ADDC A, Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	1	1	1	r	r	r

Operation:
ADDC
(A) ← (A) + (C) + (Rn)

ADDC A, direct

Bytes: 2

Cycles: 2

Encoding:

0	0	1	1	0	1	0	1	direct address

Operation:
ADDC
(A) ← (A) + (C) + (Rn)

ADDC A, @Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	1	1	0	1	1	i

Operation: ADDC
(A) ← (A) + (C) + ((Ri))

ADDC A, #data

Bytes: 2

Cycles: 2

Encoding:

0	0	1	1	0	1	0	0	immediate data

Operation: ADDC
(A) ← (A) + (C) + #data

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (after incrementing the PC twice), opcode bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2K block of program memory as the first byte of the instruction following AJMP.

Example: The label “JMPADR” is at program memory location 0123h. The instruction,

AJMP JMPADR

is at location 0345h and will load the PC with 0123h.

Bytes: 2

Cycles: 3

Encoding:

a10 a9 a8 0	0 0 0 1	a7 a6 a5 a4	a3 a2 a1 a0

Operation:

AJMP
(PC) ← (PC) + 2
(PC_{10-0}) ← page address

ANL <dest-byte>, <src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.
The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register, indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.
Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch not the input pins.

Example: If the Accumulator holds 0C3h (11000011b) and register 0 holds 55h (0101010b) then the instruction,

ANL A, R0

will leave 41h (01000001b) in the Accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The instruction,

ANL P1, # 01110011b

will clear bits 7, 3, and 2 of output port 1.

ANL A, Rn

Bytes: 1

Cycles: 1

Encoding:

0	1	0	1	1	r	r	r

Operation: ANL
(A) ← (A) ∧ (Rn)

ANL A, direct

Bytes: 2

Cycles: 2

Encoding:

0	1	0	1	0	1	0	1	direct address

Operation: ANL
(A) ← (A) ∧ (direct)

ANL A, @Ri

Bytes: 1

Cycles: 1

Encoding:

0	1	0	1	0	1	1	i

Operation:
ANL
(A) ← (A) ∧ ((Ri))

ANL A, #data

Bytes: 2

Cycles: 2

Encoding:

0	1	0	1	0	1	0	0	immediate data

Operation:
ANL
(A) ← (A) ∧ #data

ANL direct, A

Bytes: 2

Cycles: 2

Encoding:

0	1	0	1	0	0	1	0	direct address

Operation:
ANL
(direct) ← (direct) ∧ (A)

ANL direct, #data

Bytes: 3

Cycles: 3

Encoding:

0	1	0	1	0	0	1	1	direct address	immediate data

Operation:
ANL
(direct) ← (direct) ∧ #data

ANL C, <src-bit>

Function: Logical-AND for bit variables

Description: If the boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC. 7 = 1, and OV = 0:

```
MOV     C, P1.0         ; LOAD CARRY WITH INPUT PIN STATE
ANL     C, ACC.7        ; AND CARRY WITH ACCUM. BIT 7
ANL     C, /OV          ; AND WITH INVERSE OF OVERFLOW FLAG
```

ANL C, bit

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	0 0 1 0	bit address

Operation:
ANL
(C) ← (C) ∧ (bit)

ANL C, /bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 1	0 0 0 0	bit address

Operation:
ANL
(C) ← (C) ∧ ~(bit)

CJNE <dest-byte>, <src-byte>, rel

Function: Compare and Jump if Not Equal.

Description: CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction.
The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example: The Accumulator contains 34h. Register 7 contains 56h. The first instruction in the sequence,

```
                CJNE    R7, #60h, NOT_EQ
;               •••. .  ••••              ; R7 = 60h.
NOT_EQ:         JC      REQ_LOW           ; IF R7 < 60h.
;               •••     ••••              ; R7 > 60h.
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60h.
If the data being presented to Port 1 is also 34h, then the instruction,

```
WAIT:           CJNE    A, P1, WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34h.)

CJNE A, direct, rel

Bytes: 3

Cycles: 4

Encoding:

1 0 1 1	0 1 0 1	direct address	relative

Operation:
(PC) ← (PC) + 3
IF (A) < > (direct)
THEN (PC) ← (PC) + relative offset
IF (A) < (direct)
THEN (C) ← 1
ELSE (C) ← 0

CJNE A, #data, rel

Bytes: 3

Cycles: 4

Encoding:

1 0 1 1	0 1 0 0	immediate data	relative

Operation:
(PC) ← (PC) + 3
IF (A) < > data
THEN (PC) ← (PC) + relative offset
IF (A) < data
THEN (C) ← 1
ELSE (C) ← 0

CJNE Rn, #data, rel

Bytes: 3

Cycles: 4

Encoding:

1 0 1 1	1 r r r	immediate data	relative

Operation:
(PC) ← (PC) + 3
IF (Rn) < > data
THEN (PC) ← (PC) + relative offset
IF (Rn) < data
THEN (C) ← 1
ELSE (C) ← 0

CJNE @Ri, #data, rel

Bytes: 3

Cycles: 4

Encoding:

1 0 1 1	0 1 1 i	immediate data	relative

Operation:
(PC) ← (PC) + 3
IF ((Ri)) < > data
THEN (PC) ← (PC) + relative offset
IF ((Ri)) < data
THEN (C) ← 1
ELSE (C) ← 0

CLR A

Function: Clear Accumulator

Description: The Accumulator is cleared (all bits set on zero). No flags are affected.

Example: The Accumulator contains 5Ch (01011100b). The instruction,

CLR A

will leave the Accumulator set to 00h (00000000b).

Bytes: 1

Cycles: 1

Encoding:

1	1	1	0	0	1	0	0

Operation:
CLR
A ← 0

CLR bit

Function: Clear bit

Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with 5Dh (01011101b). The instruction,

CLR P1.2

will leave the port set to 59h (01011001b).

CLR C

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 0 1 1

Operation:
CLR
(C) ← 0

CLR bit

Bytes: 2

Cycles: 2

Encoding:

1 1 0 0	0 0 1 0	bit address

Operation:
CLR
(bit) ← 0

CPL A

Function: Complement Accumulator

Description: Each bit of the Accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to a zero and vice-versa. No flags are affected.

Example: The Accumulator contains 5Ch (01011100b). The instruction,

CPL A

will leave the Accumulator set to 0A3h(10100011b).

Bytes: 1

Cycles: 1

Encoding:

1	1	1	1	0	1	0	0

Operation: CPL
(A) ← ~(A)

CPL bit

Function: Complement bit

Description: The bit variable specified is complemented. A bit which had been a one is changed to zero and vice-versa. No other flags are affected. CPL can operate on the carry or any directly addressable bit.

Note: When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, not the input pin.

Example: Port 1 has previously been written with 5Bh (01011101b). The instruction sequence,

CPL P1.1

CPL P1.2

will leave the port set to 5Bh (01011011b).

CPL C

Bytes: 1

Cycles: 1

Encoding:

1	0	1	1	0	0	1	1

Operation: CPL
(C) ← ~(C)

CPL bit

Bytes: 2

Cycles: 2

Encoding:

1	0	1	1	0	0	1	0	bit address

Operation: CPL
(bit) ← ~(bit)

DA A

Function: Decimal-adjust Accumulator for Addition

Description: DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3-0 are greater than nine (xxxx1010-xxxx1111), or if the AC flag is one, six is added to the Accumulator producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxxx-1111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carryout of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00h, 06h, 60h, or 66h to the Accumulator, depending on initial Accumulator and PSW conditions.

Note: DA A cannot simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The Accumulator holds the value 56h (01010110b) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67h (01100111b) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence.

```
ADDC    A, R3
DA      A
```

will first perform a standard two's-complement binary addition, resulting in the value 0BEh (10111110b) in the Accumulator. The carry and auxiliary carry flags will be cleared.

The Decimal Adjust instruction will then alter the Accumulator to the value 24h (00100100b), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag will be set by the Decimal Adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01h or 99h. If the Accumulator initially holds 30h (representing the digits of 30 decimal), then the instruction sequence,

```
ADD     A, #99h
DA      A
```

will leave the carry set and 29h in the Accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 – 1 = 29.

Bytes: 1

Cycles: 1

Encoding:

1	1	0	1	0	1	0	0

Operation:

DA
- contents of Accumulator are BCD

IF {[(A_{3-0}) > 9] ∨ [(AC) = 1]}
THEN $(A_{3-0}) \leftarrow (A_{3-0}) + 6$
AND
IF {[(A_{7-4}) > 9] ∨ [(C) = 1]}
THEN $(A_{7-4}) \leftarrow (A_{7-4}) + 6$

DEC byte

Function: Decrement

Description: The variable indicated is decremented by 1. An original value of 00h will underflow to 0FFh.
No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Register 0 contains 7Fh (01111111b). Internal RAM locations 7Eh and 7Fh contain 00h and 40h, respectively. The instruction sequence

DEC @R0

DEC R0

DEC @R0

will leave register 0 set to 7Eh and internal RAM locations7Eh and 7Fh set to 0FFh and 3Fh.

DEC A

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	0	1	0	0

Operation: DEC
(A) ← (A) – 1

DEC Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	1	r	r	r

Operation: DEC
(Rn) ← (Rn) – 1

DEC direct

Bytes: 2

Cycles: 2

Encoding:

0	0	0	1	0	1	0	1	direct address

Operation:
DEC
(direct) ← (direct) – 1

DEC @Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	0	1	1	i

Operation:
DEC
((Ri)) ← ((Ri)) – 1

DEC DPTR

Function: Decrement Data Pointer

Description: Decrement the 16-bit data pointer by 1. A 16-bit decrement (modulo 2^{16}) is performed; an underflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will decrement the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be decremented.

Example: Registers DPH and DPL contain 12h and 01h, respectively. The instruction sequence,

```
DEC   DPTR
DEC   DPTR
DEC   DPTR
```

will change DPH and DPL to 11h and 0FEh.

Bytes: 1

Cycles: 1

Encoding:

1	0	1	0	0	1	0	1

Operation: DEC
(DPTR) ← (DPTR) - 1

DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B. The Accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

Exception: if B had originally contained 00h, the values returned in the Accumulator and B register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example: The Accumulator contains 251 (0FBh or 11111011b) and B contains 18 (12h or 00010010b). The instruction,

DIV AB

will leave 13 in the Accumulator (0Dh or 00001101b) and the value 17 (11h or 00010001b) in B, since 251 = (13 X 18) + 17. Carry and OV will both be cleared.

Bytes: 1

Cycles: 3

Encoding:

1	0	0	0	0	1	0	0

Operation: DIV

$(A)_{15\text{-}8}$, $(B)_{7\text{-}0} \leftarrow (A) / (B)$

DJNZ <byte>, <rel-addr>

Function: Decrement and Jump if Not Zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00h will underflow to 0FFh. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Internal RAM locations 40h, 50h and 60h contain the values 01h, 70h, and 15h, respectively. The instruction sequence,

```
DJNZ   40H,LABEL_1
DJNZ   50H,LABEL_2
DJNZ   60H,LABEL_3
```

will cause a jump to the instruction at label LABEL_2 with the values 00h, 6Fh, and 15h in the three RAM locations. The first jump was not taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
            MOV    R2, #8
TOGGLE:     CPL    P1.7
            DJNZ   R2, TOGGLE
```

will toggle P1.7 eight times, causing four output pukes to appear at bit 7 of output Port 1. Each pulse will last three machine cycles; two for DJNZ and one to alter the pin.

DJNZ Rn, rel

Bytes: 2

Cycles: 3

Encoding:

1 1 0 1	1 r r r	relative

Operation:

DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) – 1
IF (Rn) > 0 or (Rn) < 0
THEN (PC) ← (PC) + rel

DJNZ direct, rel

Bytes: 3

Cycles: 4

Encoding:

0 1 0 1	0 0 1 1	direct address	relative

Operation:

DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) – 1
IF (direct) > 0 or (direct) < 0
THEN (PC) ← (PC) + rel

INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFh will overflow to 00h. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Register 0 contains 7Eh (01111110b). Internal RAM locations 7Eh and 7Fh contain 0FFh and 40H, respectively. The instruction sequence,

```
INC   @R0
INC   R0
INC   @R0
```

will leave register 0 set to 7Fh and internal RAM locations 7Eh and 7Fh holding 00h and 41h, respectively.

INC A

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	1	0	0

Operation: INC
(A) ← (A) + 1

INC Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	1	r	r	r

Operation: INC
(Rn) ← (Rn) + 1

INC direct

Bytes: 2

Cycles: 2

Encoding:

0	0	0	0	0	1	0	1	direct address

Operation: INC
(direct) ← (direct) + 1

INC @Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	1	1	i

Operation: INC
((Ri)) ← ((Ri)) + 1

INC DPTR

Function: Increment Data Pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo 2^{16}) is performed; an overflow of the low-order byte of the data pointer (DPL) from 0FFh to 00h will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12h and 0FEh, respectively. The instruction sequence,

```
INC   DPTR
INC   DPTR
INC   DPTR
```

will change DPH and DPL to 13h and 01h.

Bytes: 1

Cycles: 1

Encoding:

1	0	1	0	0	0	1	1

Operation: INC
(DPTR) ← (DPTR) + 1

JB bit, rel

Function: Jump if Bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

Example: The data present at input port 1 is 11001010b. The Accumulator holds 56h (01010110b). The instruction sequence,

```
JB   P1.2, LABEL1
JB   ACC.2, LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Bytes: 3

Cycles: 4

Encoding:

0 0 1 0	0 0 0 0	bit address	relative

Operation:

```
JB
(PC) ← (PC) + 3
IF   (bit) = 1
     THEN   (PC) ← (PC) + rel
```

JBC bit, rel

Function: Jump if Bit is set and Clear bit

Description: If the indicated bit is one, branch to the address indicated; otherwise proceed with the next instruction. The bit will not be cleared if it is already a zero. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note: When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, not the input pin.

Example: The Accumulator holds 56h (01010110b). The instruction sequence,

```
JBC   ACC.3, LABELI
JBC   ACC.2, LABEL2
```

will cause program execution to continue at the instruction identified by the label LABEL2, with the Accumulator modified to 52h (01010010b).

Bytes: 3

Cycles: 4

Encoding:

0 0 0 1	0 0 0 0	bit address	relative

Operation:

```
JBC
(PC) ← (PC) + 3
IF   (bit) = 1
     THEN   (bit) ← 0
            (PC) ← (PC) + rel
```

JC rel

Function: Jump if Carry is set

Description: If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The instruction sequence,

```
JC LABEL1
CPL C
JC LABEL2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 3

Encoding:

0 1 0 0	0 0 0 0	relative

Operation: JC

(PC) ← (PC) + 2

IF (C) = 1

THEN (PC) ← (PC) + rel

JMP @A + DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the Accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
                MOV     DPTR, #JMP_TBL
                JMP     @A+DPTR
JMP_TBL:        AJMP    LABEL0
                AJMP    LABEL1
                AJMP    LABEL2
                AJMP    LABEL3
```

If the Accumulator equals 04h when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Bytes: 1

Cycles: 2

Encoding:

0	1	1	1	0	0	1	1

Operation: JMP
(PC) ← (A) + (DPTR)

JNB bit, rel

Function: Jump if Bit Not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

Example: The data present at input port 1 is 11001010b. The Accumulator holds 56h (01010110b). The instruction sequence,

```
JNB     P1.3, LABEL1
JNB     ACC.3, LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Bytes: 3

Cycles: 4

Encoding:

0 0 1 1	0 0 0 0	bit address	relative

Operation:

```
JNB
(PC) ← (PC) + 3
IF   (bit) = 0
     THEN   (PC) ← (PC) + rel
```

JNC rel

Function: Jump if Carry not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The instruction sequence,

```
JNC   LABEL1
CPL   C
JNC   LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 3

Encoding:

0 1 0 1	0 0 0 0	relative

Operation:

```
JNC
(PC) ← (PC) + 2
IF   (C) = 0
     THEN   (PC) ← (PC) + rel
```

JNZ rel

Function: Jump if Accumulator Not Zero

Description: If any bit of the Accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 00h. The instruction sequence,

```
JNZ   LABEL1
INC   A
JNZ   LAEEL2
```

will set the Accumulator to 01h and continue at label LABEL2.

Bytes: 2

Cycles: 3

Encoding:

0 1 1 1	0 0 0 0	relative

Operation:

JNZ
(PC) ← (PC) + 2
IF (A) ≠ 0
THEN (PC) ← (PC) + rel

JZ rel

Function: Jump if Accumulator Zero

Description: If all bits of the Accumulator are zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally contains 01h. The instruction sequence,

```
JZ      LABEL1
DEC     A
JZ      LABEL2
```

will change the Accumulator to 00h and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 3

Encoding:

0 1 1 0	0 0 0 0	relative

Operation:

```
JNZ
(PC) ← (PC) + 2
IF   (A) = 0
     THEN   (PC) ← (PC) + rel
```

LCALL addr16

Function: Long call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64Kbyte program memory address space. No flags are affected.

Example: Initially the Stack Pointer equals 07h. The label “SUBRTN” is assigned to program memory location 1234h. After executing the instruction,

LCALL SUBRTN

at location 0123h, the Stack Pointer will contain 09h, internal RAM locations 08h and 09h will contain 26h and 01h, and the PC will contain 1234h.

Bytes: 3

Cycles: 4

Encoding:

0	0	0	1	0	0	1	0	addr15 ~ addr8							
addr7 ~ addr0															

Operation: LCALL
(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC_{7-0})
(SP) ← (SP) + 1
((SP)) ← (PC_{15-8})
(PC) ← (PC) + 3
(PC) ← $addr_{15-0}$

LJMP addr16

Function: Long Jump

Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64Kbyte program memory address space. No flags are affected.

Example: The label “JMPADR” is assigned to the instruction at program memory location 1234h. The instruction,

LJMP JMPADR

at location 0123h will load the program counter with 1234h.

Bytes: 3

Cycles: 4

Encoding:

0 0 0 0	0 0 1 0	addr15 ~ addr8	addr7 ~ addr0

Operation: LJMP
(PC) ← $addr_{15-0}$

MOV <dest-byte>, <src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30h holds 40h. The value of RAM location 40h is 10h. The data present at input port 1 is 11001010b (0CAh).

```
MOV   R0, #30h  ;R0 < = 30h
MOV   A, @R0    ;A < = 40h
MOV   R1, A     ;R1 < = 40h
MOV   B, @R1    ;B < = 10h
MOV   @R1, P1   ;RAM (40h)< = 0CAh
MOV   P2, P1    ;P2   #0CAh
```

leaves the value 30h in register 0, 40h in both the Accumulator and register 1, 10h in register B, and 0Cah (11001010b) both in RAM location 40h and output on port 2.

MOV A, Rn

Bytes: 1

Cycles: 1

Encoding:

1	1	1	0	1	r	r	r

Operation:
MOV
(A) ← (Rn)

MOV A, direct

Bytes: 2

Cycles: 2

Encoding:

1	1	1	0	0	1	0	1	direct address

Operation:
MOV
(A) ← (direct)

MOV A, ACC is not a valid instruction

MOV A, @Ri

Bytes: 1

Cycles: 1

Encoding:

1	1	1	0	0	1	1	i

Operation: MOV
(A) ← ((Ri))

MOV A, #data

Bytes: 2

Cycles: 2

Encoding:

0	1	1	1	0	1	0	0	immediate data

Operation: MOV
(B) ← #data

MOV Rn, A

Bytes: 1

Cycles: 1

Encoding:

1	1	1	1	1	r	r	r

Operation: MOV
(Rn) ← (A)

MOV Rn, direct

Bytes: 2

Cycles: 2

Encoding:

1	0	1	0	1	r	r	r	direct address

Operation: MOV
(Rn) ← (direct)

MOV Rn, #data

Bytes: 2

Cycles: 2

Encoding:

0 1 1 1	1 r r r	immediate data

Operation: MOV
(Rn) ← #data

MOV direct, A

Bytes: 2

Cycles: 2

Encoding:

1 1 1 1	0 1 0 1	direct address

Operation: MOV
(direct) ← (A)

MOV direct, Rn

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	1 r r r	direct address

Operation: MOV
(direct) ← (Rn)

MOV direct, direct

Bytes: 3

Cycles: 3

Encoding:

1 0 0 0	0 1 0 1	dir. addr. (src)	dir. addr. (dest)

Operation: MOV
(direct) ← (direct)

MOV direct, @Ri

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	0 1 1 i	direct address

Operation: MOV
(direct) ← ((Ri))

MOV direct, #data

Bytes: 3

Cycles: 3

Encoding:

0	1	1	1	0	1	0	1	direct address	immediate data

Operation:
MOV
(direct) ← #data

MOV @Ri, A

Bytes: 1

Cycles: 1

Encoding:

1	1	1	1	0	1	1	i

Operation:
MOV
((Ri)) ← (A)

MOV @Ri, direct

Bytes: 2

Cycles: 2

Encoding:

1	0	1	0	0	1	1	i	direct address

Operation:
MOV
((Ri)) ← (direct)

MOV @Ri, #data

Bytes: 2

Cycles: 2

Encoding:

0	1	1	1	0	1	1	i	immediate data

Operation:
MOV
((Ri)) ← #data

MOV <dest-bit>, <src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input Port 3 is 11000101b. The data previously written to output Port 1 is 35h (00110101b).

```
MOV   P1.3, C
MOV   C, P3.3
MOV   P1.2, C
```

will leave the carry cleared and change Port 1 to 39h (00111001b).

MOV C, bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	0 0 1 0	bit address

Operation:
MOV
(C) ← (bit)

MOV bit, C

Bytes: 2

Cycles: 2

Encoding:

1 0 0 1	0 0 1 0	bit address

Operation:
MOV
(bit) ← (C)

MOV DPTR, #data16

Function: Load Data Pointer with a 16-bit constant

Description: The Data Pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction, which moves 16 bits of data at once.

Example: The instruction,

MOV DPTR, # 1234h

will load the value 1234h into the Data Pointer: DPH will hold 12h and DPL will hold 34h.

Bytes: 3

Cycles: 3

Encoding:

1	0	0	1	0	0	0	0	imm. data 15~8	imm. data 7~0

Operation:

MOV

(DPTR) ← $\#data_{15\text{-}0}$

{DPH, DPL} ← {$\#data_{15\text{-}8}$, $\#data_{7\text{-}0}$}

MOVC A, @A + <base-reg>

Function: Move Code byte

Description: The MOVC instructions load the Accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit Accumulator contents and the contents of a sixteen-bit base register, which may be either the Data Pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added with the Accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive.

```
REL-PC:   INC    A
          MOVC   A, @A+PC
          RET
          DB     66h
          DB     77h
          DB     88h
          DB     99h
```

If the subroutine is called with the Accumulator equal to 01h, it will return with 77h in the Accumulator. The INC A before the MOVC instruction is needed to “get around” the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the Accumulator instead.

MOVC A, @A + DPTR

Bytes: 1

Cycles: 2

Encoding:

1	0	0	1	0	0	1	1

Operation: MOV
(A) ← ((A) + (DPTR))

MOVC A, @A + PC

Bytes: 1

Cycles: 2

Encoding:

1	0	0	1	0	0	1	1

Operation: MOVC
PC ← PC + 1
(A) ← ((A) + (PC))

Note: It is recommended that all interrupts are disabled before using MOVC instruction.

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The Iow-order byte of the sixteen-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (0FFh) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Example: Originally the Accumulator holds the value 80 (50h). Register B holds the value 160 (0A0h). The instruction,

MUL AB

will give the product 12,800 (3200h), so B is changed to 32h (00110010b) and the Accumulator is cleared. The overflow flag is set, carry is cleared.

Bytes: 1

Cycles: 3

Encoding:

1	0	1	0	0	1	0	0

Operation: MUL

$(A)_{7\text{-}0}$, $(B)_{15\text{-}8} \leftarrow (A) \text{ X } (B)$

NOP

Function: No Operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of Port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enabled) with the instruction sequence,

```
CLR     P2.7
NOP
NOP
NOP
NOP
SETB    P2.7
```

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	0	0	0

Operation:
NOP
(PC) ← (PC) + 1

ORL <dest-byte> <src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: If the Accumulator holds 0C3h (11000011b) and R0 holds 55h (01010101b) then the instruction,

ORL A, R0

will leave the Accumulator holding the value 0D7h (11010111b).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The instruction,

ORL P1, # 00110010b

will set bits 5, 4, and 1 of output Port 1.

ORL A, Rn

Bytes: 1

Cycles: 1

Encoding:

0	1	0	0	1	r	r	r

Operation: ORL
(A) ← (A) ∨ (Rn)

ORL A, direct

Bytes: 2

Cycles: 2

Encoding:

0	1	0	0	0	1	1	i	direct address

Operation: ORL
(A) ← (A) ∨ (direct)

ORL A, @Ri

Bytes: 1

Cycles: 1

Encoding:

0	1	0	0	0	1	1	i

Operation: ORL
(A) ← (A) ∨ ((Ri))

ORL A, #data

Bytes: 2

Cycles: 2

Encoding:

0	1	0	0	0	1	0	0	immediate data

Operation: ORL
(A) ← (A) ∨ #data

ORL direct, A

Bytes: 2

Cycles: 2

Encoding:

0	1	0	0	0	0	1	0	direct address

Operation: ORL
(direct) ← (direct) ∨ (A)

ORL direct, #data

Bytes: 3

Cycles: 3

Encoding:

0	1	0	0	0	0	1	1	direct address	immediate data

Operation: ORL
(direct) ← (direct) ∨ #data

ORL C, <src-bit>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if and only if P1.0 = 1, ACC. 7 = 1, or OV = 0:

```
MOV     C, P1.0         ;LOAD CARRY WITH INPUT PIN P1.0
ORL     C, ACC.7        ;OR CARRY WITH THE ACC.BIT 7
ORL     C, /OV          ;OR CARRY WITH THE INVERSE OF OV.
```

ORL C, bit

Bytes: 2

Cycles: 2

Encoding:

0 1 1 1	0 0 1 0	bit address

Operation:
ORL
(C) ← (C) ∨ (bit)

ORL C, /bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	0 0 0 0	bit address

Operation:
ORL
(C) ← (C) ∨ ~(bit)

POP direct

Function: Pop from stack.

Description: The contents of the internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by 1. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example: The Stack Pointer originally contains the value 32h, and internal RAM locations 30h through 32h contain the values 20h, 23h, and 01h, respectively. The instruction sequence,

```
POP  DPH
POP  DPL
```

will leave the Stack Pointer equal to the value 30H and the Data Pointer set to 0123h. At this point the instruction,

```
POP  SP
```

will leave the Stack Pointer set to 20h. Note that in this special case the Stack Pointer was decremented to 2Fh before being loaded with the value popped (20h).

Bytes: 2

Cycles: 2

Encoding:

1 1 0 1	0 0 0 0	direct address

Operation:

POP
(direct) ← ((SP))
(SP) ← (SP) – 1

PUSH direct

Function: Push onto stack

Description: The Stack Pointer is incremented by 1. The contents of the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.

Example: On entering an interrupt routine the Stack Pointer contains 09h. The Data Pointer holds the value 0123h.The instruction sequence,

```
PUSH DPL
PUSH DPH
```

will leave the Stack Pointer set to 0Bh and store 23h and 01h in internal RAM locations 0Ah and 0Bh, respectively.

Bytes: 2

Cycles: 2

Encoding:

1 1 0 1	0 0 0 0	direct address

Operation:

PUSH
(SP) ← (SP) + 1
((SP)) ← (direct)

RET

Function: Return from subroutine

Description: RET pops the high-and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by 2. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.

Example: The Stack Pointer originally contains the value 0Bh. Internal RAM locations 0Ah and OBh contain the values 23h and 01h, respectively. The instruction,

RET

will leave the Stack Pointer equal to the value 09h. Program execution will continue at Iocation 0123h.

Bytes: 1

Cycles: 2

Encoding:

0	0	1	0	0	0	1	0

Operation:

RET
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RETI

Function: Return from interrupt

Description: RETI pops the high-and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected; the PSW is not automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower-or same-level interrupt had been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.

Example: The Stack Pointer originally contains the value 0Bh. An interrupt was detected during the instruction ending at location 0122h. Internal RAM locations 0Ah and 0Bh contain the values 23h and 01h, respectively. The instruction,

RETI

will leave the Stack Pointer equal to 09h and return program execution to location 0123h.

Bytes: 1

Cycles: 2

Encoding:

0	0	1	1	0	0	1	0

Operation:

RETI
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RL A

Function: Rotate Accumulator Left

Description: The eight bits in the Accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.

Example: The Accumulator holds the value 0C5h (11000101b). The instruction,

RL A

leaves the Accumulator holding the value 8Bh (10001011b) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0	0	1	0	0	0	1	1

Operation:

RL
$(A_n + 1) \leftarrow (A_n)$ n = 0 ~ 6
(A0) ← (A7)

RLC A

Function: Rotate Accumulator Left through the Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example: The Accumulator holds the value 0C5h (11000101b), and the carry is zero. The instruction,

RLC A

leaves the Accumulator holding the value 8Bh (10001010b) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0	0	1	1	0	0	1	1

Operation:

RLC
$(A_n + 1) \leftarrow (A_n)$ n = 0 ~ 6
(A0) ← (C)
(C) ← (A7)

RR A

Function: Rotate Accumulator Right

Description: The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example: The Accumulator holds the value 0C5h (11000101b). The instruction,

RR A

leaves the Accumulator holding the value 0E2h (11100010b) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	0	1	1

Operation:

RR
$(A_n) \leftarrow (A_n + 1)$ n = 0 ~ 6
(A7) ← (A0)

RRC A

Function: Rotate Accumulator Right through Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original value of the carry flag moves into the bit 7 position. No other flags are affected.

Example: The Accumulator holds the value 0C5h (11000101b), the carry is zero. The instruction,

RRC A

leaves the Accumulator holding the value 62h (01100010b) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	0	0	1	1

Operation:
RRC
$(A_n) \leftarrow (A_n + 1)$ n = 0 ~ 6
(A7) ← (C)
(C) ← (A0)

SETB <bit>

Function: Set Bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output Port 1 has been written with the value 34h (00110100b). The instructions,

```
SETB  C
SETB  PI.0
```

will leave the carry flag set to 1 and change the data output on Port 1 to 35h (00110101b).

SETB C

Bytes: 1

Cycles: 1

Encoding:

1	1	0	1	0	0	1	1

Operation:
SETB
(C) ← 1

SETB bit

Bytes: 2

Cycles: 2

Encoding:

1	1	0	1	0	0	1	0	bit address

Operation:
SETB
(bit) ← 1

SJMP rel

Function: Short Jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example: The label “RELADR” is assigned to an instruction at program memory location 0123h. The instruction,

SJMP RELADR

will assemble into location 0100h. After the instruction is executed, the PC will contain the value 0123h.

(Note: Under the above conditions the instruction following SJMP will be at 102h. Therefore, the displacement byte of the instruction will be the relative offset (0123h – 0102h) = 21h. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.)

Bytes: 2

Cycles: 3

Encoding:

1 0 0 0	0 0 0 0	relative

Operation: SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel

SUBB A, <src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set before executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the Accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

Example: When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

The Accumulator holds 0C9h (11001001b), register 2 holds 54h (01010100b), and the carry flag is set. The instruction,

SUBB A, R2

will leave the value 74h (01110100b) in the accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9h minus 54h is 75h. The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction.

SUBB A, Rn

Bytes: 1

Cycles: 1

Encoding:

1	0	0	1	1	r	r	r

Operation: SUBB
(A) ← (A) – (C) – (Rn)

SUBB A, direct

Bytes: 2

Cycles: 2

Encoding:

1	0	0	1	0	1	0	1	direct address

Operation: SUBB
(A) ← (A) – (C) – (direct)

SUBB A, @Ri

Bytes: 1

Cycles: 1

Encoding:

1	0	0	1	0	1	1	i

Operation: SUBB
(A) ← (A) – (C) – ((Ri))

SUBB A, #data

Bytes: 2

Cycles: 2

Encoding:

1	0	0	1	0	1	0	0	immediate data

Operation: SUBB
(A) ← (A) – (C) – #data

SWAP A

Function: Swap nibbles within the Accumulator

Description: SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the Accumulator (bits 3-0 and bits 7-4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.

Example: The Accumulator holds the value 0C5h (11000101b). The instruction,

SWAP A

leaves the Accumulator holding the value 5Ch (01011100b).

Bytes:

Cycles: 1

Encoding: 1

Operation:

1	1	0	0	0	1	0	0

XCH
$(A_{3-0}) \leftrightarrow (A_{7-4})$

XCH A, <byte>

Function: Exchange Accumulator with byte variable

Description: XCH leads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example: R0 contains the address 20h. The Accumulator holds the value 3Fh (00111111b). Internal RAM location 20h holds the value 75h (01110101b). The instruction,

XCH A, @R0

will leave RAM location 20h holding the values 3Fh (00111111b) and 75h (01110101b) in the accumulator.

XCH A, Rn

Bytes: 1

Cycles: 1

Encoding:

1	1	0	0	1	r	r	r

Operation:
XCH
(A) ↔ (Rn)

XCH A, direct

Bytes: 2

Cycles: 2

Encoding:

1	1	0	0	0	1	0	1	direct address

Operation:
XCH
(A) ↔ (direct)

XCH A, @Ri

Bytes: 1

Cycles: 1

Encoding:

1	1	0	0	0	1	1	i

Operation:
XCH
(A) ↔ ((Ri))

XCHD A, @Ri

Function: Exchange Digit

Description: XCHD exchanges the low-order nibble of the Accumulator (bits 3-0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected

Example: R0 contains the address 20h. The Accumulator holds the value 36h (00110110b). Internal RAM location 20h holds the value 75h (01110101b). The instruction,

XCHD A, @R0

will leave RAM location 20h holding the value 76h (01110110b) and 35h (00110101b) in the Accumulator.

Bytes: 1

Cycles: 1

Encoding:

1	1	0	1	0	1	1	i

Operation:
XCH
$(A_{3\text{-}0}) \leftrightarrow ((Rn_{3\text{-}0}))$

XRL <dest-byte>, <src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

(Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.)

Example: If the Accumulator holds 0C3h (11000011b) and register 0 holds 0Aah (10101010b) then the instruction,

XRL A, R0

will leave the Accumulator holding the value 69h (01101001b).

When the destination is a directly addressed byte this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The instruction,

XRL PI, #00110001b

will complement bits 5, 4, and 0 of output Port 1.

XRL A, Rn

Bytes: 1

Cycles: 1

Encoding:

0	1	1	0	1	r	r	r

Operation:
XRL
(A) ← (A) ⊗ (Rn)

XRL A, direct

Bytes: 2

Cycles: 2

Encoding:

0	1	1	0	0	1	0	1	direct address

Operation: XRL
(A) ← (A) ⊗ (direct)

XRL A, @Ri

Bytes: 1

Cycles: 1

Encoding:

0	1	1	0	0	1	1	i

Operation: XRL
(A) ← (A) ⊗ ((Ri))

XRL A, #data

Bytes: 2

Cycles: 2

Encoding:

0	1	1	0	0	1	0	0	immediate data

Operation: XRL
(A) ← (A) ⊗ #data

XRL direct, A

Bytes: 2

Cycles: 2

Encoding:

0	1	1	0	0	0	1	0	direct address

Operation: XRL
(direct) ← (direct) ⊗ (A)

XRL direct, #data

Bytes: 3

Cycles: 3

Encoding:

0	1	1	0	0	0	1	1	direct address	immediate data

Operation: XRL
(direct) ← (direct) ⊗ #data

13 Appendix B: SFR description

13.1 P0 (80h) : Port 0 Register

Bit No.	7	6	5	4	3	2	1	0
	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register functions as a general purpose I/O port. The initial value of P0 is FFh.

[About the alternative pins and description]

In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs.

Port 0	P0[7]	P0[6]	P0[6]	P0[4]	P0[3]	P0[2]	P0[1]	P0[0]
	ADC6	ADC5 PWM	ADC4	ADC3 AV_{REF}	ADC2 INT3B	ADC1 INT2 TXD	ADC0 INT1B RXD	INT0B TVO PWM

Alternative Pin	Description
ADC6 ~ ADC0	ADC Analog Input 6 ~ 0
AV_{REF}	ADC Voltage Reference Input.
PWM	PWM Waveform Output. (Refer to ALTSEL register)
INT3B	External Interrupt 3. A falling edge on this pin will cause an external interrupt 3 if enabled.
INT2	External Interrupt 2. A rising edge on this pin will cause an external interrupt 2 if enabled.
INT1B	External Interrupt 1. A falling edge or low level on this pin will cause an external interrupt 1 if enabled.
INT0B	External Interrupt 0. A falling edge or low level on this pin will cause an external interrupt 0 if enabled.
TXD	Serial Port Transmit. This pin transmits the serial port data in serial port modes 1 and emits the synchronizing clock in serial port mode 0. (Refer to ALTSEL register)
RXD	Serial Port Receive. This pin receives the serial port data in serial port modes 1 and is a bi-directional data transfer pin in serial port mode 0.

TVO	Timer 0 Overflow Output. (Refer to ALTSEL register)

13.2 SP (81h) : Stack Pointer Register

Bit No.	7	6	5	4	3	2	1	0
	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

The stack pointer identifies the location where the stack will begin. The stack pointer is incremented before every PUSH operation. This register defaults to 07H after reset.

13.3 DPL (82h) : Data Pointer Low Register

Bit No.	7	6	5	4	3	2	1	0
	DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register is the low byte of the 16-bit data pointer. DPL and DPH are used to point to non-scratchpad data RAM.

13.4 DPH (83h) : Data Pointer High Register

Bit No.	7	6	5	4	3	2	1	0
	DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register is the high byte of the 16-bit data pointer. DPL and DPH are used to point to non-scratchpad data RAM.

13.5 PCON (87h) : Power Control Register

Bit No.	7	6	5	4	3	2	1	0
	SMOD1	-	-	POF	GF1	GF0	PD	IDL
	R/W(0)			R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
SMOD1	Timer1 Buadrate Double in UART mode. 0 = Double buadrate disable (Default) 1 = Double baudrate enable
-	Reserved
POF	Power Off Flag. When POR (Power-On Reset), this bit will be set to "1" by H/W. For all resets except of POR, the value of POF is not changed if user sets it bit to "0".
GF1	General Purpose Flag Bit.
GF0	General Purpose Flag Bit.
PD	Power-down (Stop) Mode Enable. Setting this bi causes the MCU (MiDAS1.1 family) to go into the power-down mode. In this mode, all the clocks are stopped and program execution is frozen. 0 (Default), 1 = Power-down mode enable.
IDL	Idle Mode Enable. Setting this bit causes the MiDAS1.1 family to go into the IDLE mode. In this mode the clocks to the CPU are stopped, so program execution is frozen. But the clock to the peripherals and interrupt blocks is not stopped, and these blocks continue operating. 0 (Default), 1 = Idle mode enable.

13.6 TCON (88h) : Timer/Counter 0/1 Control Register

Bit No.	7	6	5	4	3	2	1	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
TF1	Timer 1 Overflow Flag. This bit indicates when Timer 1 overflows its maximum count as defined by the current mode. This bit can be cleared by software. 0 = No Timer/Counter 1 overflow has been detected. 1 = Timer/Counter 1 has overflowed with its maximum count
TR1	Timer 1 Run Enable.
TF0	Timer 0 Overflow Flag. This bit indicates when Timer 0 overflows its maximum count as defined by the current mode. This bit can be cleared by software. 0 = No Timer/Counter 0 overflow has been detected. 1 = Timer/Counter 0 has overflowed with its maximum count
TR0	Timer 0 Run Enable.
IE1	External Interrupt 1 Flag. This bit is set when an edge/level type defined by IT1 flag is detected. If IT1=1, this bit will remain set until cleared in software or the start of the External Interrupt 1 service routine. If IT1=0, this bit will inversely reflect the state of the INT1B pin.
IT1	External Interrupt 1 Type Select Flag. This bit selects whether the INT1B pin will detect the edge/level triggered interrupts. 0 = INT1B is level triggered (Level detect type). (Default) 1 = INT1B is edge triggered (Edge detect type).
IE0	External Interrupt 0 Flag. This bit is set when an edge/level type defined by IT0 flag is detected. If IT0=1, this bit will remain set until cleared in software or the start of the External Interrupt 0 service routine. If IT0=0, this bit will inversely reflect the state of the INT0B pin.
IT0	External Interrupt 0 Type Select Flag. This bit selects whether the INT0B pin will detect the edge/level triggered interrupts. 0 = INT0B is level triggered (Level detect type). (Default) 1 = INT0B is edge triggered (Edge detect type).

13.7 **TMOD** (89h) : Timer/Counter 0 Mode Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	GATE	C/T	M1	M0
					R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
-	Reserved
GATE	Timer 0 Gate Control. This bit enables/disables the ability of Timer 0 to increment. 0 = Timer 0 will clock when TR0=1, regardless of the states of INT0B. (Default) 1 = Timer 0 will clock only when TR0=1 and INT0B=1.
C/T	Timer 0 Counter/Timer Select. 0 = Timer. Timer 0 is incremented by internal clocks (system clock). (Default) 1 = Counter. Timer 0 is incremented by pulsed on T0 pin when TR0=1.
M1, M0	

Mode	M1	M0	Description
0	0	0	8 bits with 5-bit prescale
1	0	1	16 bits
2	1	0	8 bits with auto-reload
3	1	1	Two 8-bit timer/counter

13.8 TL0 (8Ah) : Timer/Counter 0 Low Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the least significant byte of Timer 0.

13.9 TL1 (8Bh) : Timer/Counter 1 Low Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the least significant byte of Timer 1.

13.10 TH0 (8Ch) : Timer/Counter 0 High Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the most significant byte of Timer 0.

13.11 TH1 (8Dh) : Timer/Counter 1 High Byte Register

Bit No.	7	6	5	4	3	2	1	0
	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register contains the most significant byte of Timer 1.

13.12 P1 (90h) : Port 1 Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	P0.2	P0.1	P0.0
						R/W(1)	R/W(1)	R/W(1)

This register functions as a general purpose I/O port. The initial value of P0 is FFh.

[About the alternative pins and description]

In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs

Port 1	-	-	-	-	-	P1[2]	P1[1]	P1[0]
						RESETB	XTAL2	XTAL1

Alternative Pin	Description
XTAL1	External Crystal Oscillator Input. (Refer to ALTSEL)
XTAL2	External Crystal Oscillator Output. (Refer to ALTSEL)
RESETB	Inverted Reset Input. Active Low. (Refer to ALTSEL)

13.13 **EXIF** (91h) : External Interrupt Flag Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	IE3	IE2	XT/RG	RGMD	RGSL	BGS
			R/W(0)	R/W(0)	R/W(1)	R(1)	R/W(0)	R/W(1)

Symbol	Description
-	Reserved.
IE3	External Interrupt 3 Flag. This bit will be set when a falling edge is detected on INT3B. This bit must be cleared manually by software. Setting this bit in software will cause an interrupt if enabled.
IE2	External Interrupt 2 Flag. This bit will be set when a rising edge is detected on INT2. This bit must be cleared manually by software. Setting this bit in software will cause an interrupt if enabled.
XT/RG	System Clock Selection. This bit is set to "0" after a power-on reset, and unchanged by all other resets. Note : Don't clear RINGON (OSCICN.2) flag when F_{OSC} is Ring oscillator (XT/RG = 0). 0 = The Ring oscillator is selected as the system clock. (Default) 1 = The crystal oscillator is selected as the system clock.
RGMD	Ring Mode. Now system clock is Ring or crystal oscillator Generally RGMD is the invert of XT/RG.
RGSL	Ring Select Bit when Power-down Wake-up. 1 = When wake-up from Power-down mode in XTAL clock, the CPU uses Ring oscillator as system clock during 65,536 XTAL clock. (Default = 0)
BGS	Band-Gap Select. This bit enables/disables the band-gap reference during power-down mode. Disabling the band-gap reference provides significant power savings in power-down mode, but sacrifices the ability to perform power-fail reset while stopped. The state of this bit will be undefined on devices which do not incorporate a band-gap reference. 0 = The band-gap reference is disabled in power-down mode, but will function during normal operation. It will support the significant power savings in power-down mode. 1 = The band-gap reference will operate in Stop mode. (Default)

13.14 SCON (98h) : Serial Port Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	REN	-	-	TI	RI
				R/W(0)			R/W(0)	R/W(0)

Symbol	Description
-	Reserved
REN	Serial Reception Enable. 0 = Serial reception disable. (Default) 1 = Serial reception enable.
TI	Transmission Interrupt Flag. This bit indicates that a byte of data in the serial port buffer (SBUF) has been completely shifted out. Must be cleared by software.
RI	Reception Interrupt Flag. This bit indicates that a byte of data has been received in the serial port buffer (SBUF). Must be cleared by software.

13.15 SBUF (99h) : Serial Data Buffer Register

Bit No.	7	6	5	4	3	2	1	0
	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Data for serial port is read from or written to this location. The buffers for serial transmission and serial reception are separate registers, but both are addressed at the same location.

13.16 P2 (A0h) : Port 2 Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register functions as a general purpose I/O port. The initial value of P2 is *1111111.

[About the alternative pins and description]

In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs.

Port 2	-	P2[6]	P2[5]	P2[4]	P2[3]	P2[2]	P2[1]	P2[0]
		ADC7 CLO	ADC8	ADC9	ADC10	ADC11		T0

Alternative Pin	Description
ADC11 ~ ADC7	ADC Analog Input 11 ~ 7
CLO	System Clock Output to P2[6]. (Refer to ALTSEL register)
T0	Timer0 Input or T0 Input. (Refer to TMOD register) A "1" to "0" transition on this T0 input pin will increment the timer0 counter.

13.17 IE (A8h) : Interrupt Enable Register

Bit No.	7	6	5	4	3	2	1	0
	EA	EADC	-	ES	ET1	EX1	ET0	EX0
	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
EA	Global Interrupt Enable. This bit controls the global masking of all interrupts. 0 = Disable all individual interrupt masks. (Default) 1 = Enable all individual interrupt masks.
EADC	ADC Interrupt Enable. This bit controls the masking of the ADC interrupt. 0 = Disable the ADC interrupt. (Default) 1 = Enable the ADC interrupt.
-	Reserved.
ES	Serial Port Interrupt Enable. This bit controls the masking of the serial port interrupt. (RI + TI) 0 = Disable the serial port interrupt. (Default) 1 = Enable the serial port interrupt.
ET1	Timer1 Interrupt Enable. This bit controls the masking of the Timer1 interrupt. 0 = Disable the Timer1 interrupt. (Default) 1 = Enable the Timer1 interrupt.
EX1	External Interrupt 1 Enable. This bit controls the masking of the external interrupt 1. 0 = Disable the external interrupt 1. (Default) 1 = Enable the external interrupt 1 (INT1B).
ET0	Timer0 Interrupt Enable. 0 = Disable the Timer0 interrupt. (Default) 1 = Enable the Timer0 interrupt.
EX0	External Interrupt 0 Enable. 0 = Disable the external interrupt 0. (Default) 1 = Enable the external interrupt 0 (INT0B).

13.18 IP (B8h) : Interrupt Priority Register

Bit No.	7	6	5	4	3	2	1	0
	-	PADC	-	PS	PT1	PX1	PT0	PX0
	R(1)	R/W(0)		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
-	Reserved.
PADC	ADC Interrupt Priority. This bit controls the priority of the ADC interrupt up to two levels. 0 = Priority level 0 for ADC interrupt. (Default) 1 = Priority level 1 for ADC interrupt. Note : The priority level 0 means the low priority interrupt level. And, the priority level 1 means the high priority interrupt level.
PS	Serial Port (UART) Interrupt Priority. This bit controls the priority of the serial port interrupt up to two levels. 0 = Priority level 0 for serial port interrupt. (Default) 1 = Priority level 1 for serial port.
PT1	Timer1 Interrupt Priority. This bit controls the priority of the ADC interrupt up to two levels. 0 = Priority level 0 for timer1 interrupt. (Default) 1 = Priority level 1 for timer1 interrupt.
PX1	External Interrupt 1 Priority. This bit controls the priority of the external interrupt 1 up to two levels. 0 = Priority level 0 for external interrupt 1. (Default) 1 = Priority level 1 for external interrupt 1.
PT0	Timer0 Interrupt Priority. This bit controls the priority of the ADC interrupt up to two levels. 0 = Priority level 0 for timer0 interrupt. (Default) 1 = Priority level 1 for timer0 interrupt.
PX0	External Interrupt 1 Priority. This bit controls the priority of the external interrupt 0 up to two levels. 0 = Priority level 0 for external interrupt 0. (Default) 1 = Priority level 1 for external interrupt 0.

13.19 **OSCICN** (BEh) : Internal Ring Oscillator Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	RINGON	DIV1	DIV0
						R/W(1)	R/W(0)	R/W(0)

Symbol	Description			
-	Reserved.			
RINGON	Internal Ring Oscillator Enable. Note : Don't clear this bit when F_{OSC} is Ring oscillator (XT/RG = 0). But, although F_{OSC} is crystal oscillator, internal ring oscillator can be run (RINGON=1). 0 = Internal Ring Oscillator is killed. (Default) 1 = Internal Ring oscillator is running.			
DIV1 DIV0	Ring Oscillator Divider			
	Mode	DIV1	DIV0	Description (Ring Oscillator @ 5V)
	0	0	0	4 MHz / 1
	1	0	1	4 MHz / 2
	2	1	0	4 MHz / 4
	3	1	1	4 MHz / 8

13.20 **PMR** (C4h) : Power Management Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	XTOFF	-	-	-
					R/W(0)			

Symbol	Description
XTOFF	Internal Amplifier Disable for External Crystal Oscillator. This bit controls the enable/disable the internal amplifier for external crystal oscillator. 0 = Internal amplifier is enabled. (Default) 1 = Internal amplifier is disabled. Set the XTOFF bit to "1" when user wants to use P1[0] and P1[2] as general I/O pins. Don't set XTOFF bit to "1" when XT/RG bit (EXIF.3) is "1". (XT/RG = "1" : The external clock is selected as system clock. Refer to EXIF register.)

13.21 **STATUS** (C5h) : Crystal Status Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	XTUP	-	-	-	-
				R(0)				

Symbol	Description
-	Reserved.
XTUP	Crystal Oscillator Warm-up Status. This bit indicates whether the CPU crystal oscillator has completed the 65,536 cycle warm-up, and represents the system clock of the crystal oscillator is stable(1) or not(0). This bit is cleared by H/W when Power-on Reset, during Power-down wake-up. This bit is set to 1 following a crystal stabilization by H/W.

13.22 PSW (D0h) : Program Status Word Register

Bit No.	7	6	5	4	3	2	1	0
	CY	AC	F0	RS1	RS0	OV	F1	P
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(0)

<table>
<tr><th>Symbol</th><th>Description</th></tr>
<tr><td>CY</td><td>Carry Flag.
This bit is set when if the last arithmetic operation resulted in a carry (during addition) or borrow (during subtraction). Otherwise it is cleared to 0 by all arithmetic operations.</td></tr>
<tr><td>AC</td><td>Auxiliary Carry Flag.
This bit is set to 1 if the last arithmetic operation resulted in a carry into (during addition), or borrow (during subtraction) from the high order nibble. Otherwise it is cleared to 0 by all arithmetic operations</td></tr>
<tr><td>F0</td><td>User Flag 0.
This is a general purpose flag for software control.</td></tr>
<tr><td>RS1
RS0</td><td>Register Bank (RB) Select.
These bits select which register bank is addressed during register accesses.
<table>
<tr><th>RS1</th><th>RS0</th><th>Register Bank</th><th>Address</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>00h ~ 07h</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>08h ~ 0Fh</td></tr>
<tr><td>1</td><td>0</td><td>2</td><td>10h ~ 17h</td></tr>
<tr><td>1</td><td>1</td><td>3</td><td>18h ~ 1Fh</td></tr>
</table></td></tr>
<tr><td>OV</td><td>Overflow Flag.
This bit is set to 1 if the last arithmetic operation resulted in a carry (addition), borrow (subtraction), or overflow (multiply or divide). Otherwise it is cleared to 0 by all arithmetic operations.</td></tr>
<tr><td>F1</td><td>User Flag 1.
This is a general purpose flag for software control.</td></tr>
<tr><td>P</td><td>Parity Flag.
This bit is set to 1 if the modulo-2 sum of the eight bits of the accumulator is 1 (odd parity); and cleared to 0 on even parity.</td></tr>
</table>

13.23 P0TYPE (D4h) : Port 0 Type Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0TYPE.7	P0TYPE.6	P0TYPE.5	P0TYPE.4	P0TYPE.3	P0TYPE.2	P0TYPE.1	P0TYPE.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register configures the P0 output type to the push-pull or open-drain. The initial P0 output type is configured to push-pull output type.

0 = Push-pull output type. (Default)

1 = Open-drain output.

13.24 P1TYPE (D5h) : Port 1 Type Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	P1TYPE.1	P1TYPE.0
							R/W(0)	R/W(0)

This register configures the P0 output type to the push-pull or open-drain. The initial P1 output type is configured to push-pull output type. Note that the P1[2] pin is only the open-drain output type.

0 = Push-pull output type. (Default)

1 = Open-drain output.

13.25 P2TYPE (D6h) : Port 2 Type Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2TYPE.6	P2TYPE.5	P2TYPE.4	P2TYPE.3	P2TYPE.2	P2TYPE.1	P2TYPE.0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register configures the P2 output type to the push-pull or open-drain. The initial P2 output type is configured to push-pull output type.

0 = Push-pull output type. (Default)

1 = Open-drain output.

13.26 **WDCON** (D8h) : Watchdog Control Register

Bit No.	7	6	5	4	3	2	1	0
	WD1	WD0	-	-	WDIF	WTRF	EWT	RWT
	R/W(1)	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

<table>
<tr><th>Symbol</th><th colspan="5">Description</th></tr>
<tr><td rowspan="6">WD1
WD0</td><td colspan="5">Watchdog Timer Mode Select</td></tr>
<tr><td>Mode</td><td>WD1</td><td>WD0</td><td>Interrupt Time-Out</td><td>Reset Time-Out</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1 X 2^{16} clocks</td><td>1 X 2^{16} + 256 clocks</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>4 X 2^{16} clocks</td><td>4 X 2^{16} + 256 clocks</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>16 X 2^{16} clocks</td><td>16 X 2^{16} + 256 clocks</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>132X 2^{16} clocks</td><td>32 X 2^{16} + 256 clocks</td></tr>
<tr><td rowspan="7">WDIF</td><td colspan="5">Watchdog Interrupt Flag.
This bit, in conjunction with the Watchdog timer interrupt enable bit EWDT (EIE.4) and Enable Watchdog Timer Reset bit EWT (WDCON.1) indicates if a watchdog timer event has occurred and what action will be taken. This bit must be cleared in software before exiting the interrupt service routine, or another interrupt will be generated. Setting this bit in software will generate a watchdog interrupt if enabled.
(EWT → WDCON.1, EWDT → EIE.4, WDIF → WDCON.3)</td></tr>
<tr><td>EWT</td><td>EWDT</td><td>WDIF</td><td colspan="2">Description</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td colspan="2">No WDT (Watchdog timer) event has occurred.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="2">WDT time-out has expired. No interrupt has been generated.
WDT reset will occur after 512 clocks if RWT is not strobed.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="2">WDT interrupt has occurred.
WDT reset will occur after 512 clocks if RWT is not set.</td></tr>
<tr><td>WTRF</td><td colspan="5">Watchdog Timer Reset Flag.
When set, this bit indicates that a watchdog timer reset has occurred. It is typically interrogated to determine if a reset was caused by watchdog timer reset. It is cleared by a power-on-reset, but otherwise must be cleared by software before the next reset of any kind or software may erroneously determine that a watchdog timer reset has occurred. Setting this bit in software will not generate a watchdog timer reset. If the EWT bit is cleared, the watchdog timer will have no effect on this bit.</td></tr>
</table>

EWT	Enable Watchdog Timer Reset. This bit enables/disables the ability of the watchdog timer to reset the device. This bit has no effect on the ability of the watchdog timer to generate a watchdog interrupt. The time-out period of the watchdog timer is controlled by the Watchdog Timer Mode Select bits (CKCON.7-6). Clearing these bits will disable the ability of the watchdog timer to generate a reset, but have no affect on the timer itself, or its ability to generate a watchdog timer interrupt. 0 = A time-out of the watchdog timer will not cause the device to reset. 1 = A time-out of the watchdog timer will cause the device to reset.
RWT	Restart Watchdog Timer. This bit serves as the strobe for the Watchdog function. During the time-out period, software must set the RWT bit if the Watchdog is enabled. Failing to set the RWT will cause a reset when the time-out has elapsed. There is no need to set the RWT bit to a 0 because it is self-clearing. Note that RWT flag bit must be set to "1" at only WDT mode 0. After watchdog timer is restarted, WDT mode can be changed to mode 3 or any mode.

13.27 **PWMCON** (DCh) : PWM Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0SEL	PS2_P0	PS1_P0	PS0_P0	-	PWMF	CLR_P0	RUN_P0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	R/W(0)	R/W(0)

Symbol	Description			
P0SEL	PWM Waveform Output Enable to P0[6]. 0 = Disable the PWM waveform output to P[6]. (Default) 1 = Enable the PWM waveform output to P[6].			
PS2_P0 PS1_P0 PS0_P0	Prescaled Clock Selection. Note that PWM clock (F_{PWM}) to ADC should not be set to F_{OSC}/1.			
	PS2_P0	PS1_P0	PS0_P0	PWM Clock Rate (F_{PWM})
	0	0	0	F_{OSC} / 1
	0	0	1	F_{OSC} / 2
	0	1	0	F_{OSC} / 4
	0	1	1	F_{OSC} / 8

	1	0	0	F_{OSC} / 16
	1	0	1	F_{OSC} / 32
	1	1	0	F_{OSC} / 64
	1	1	1	F_{OSC} / 128
-	Reserved.			
PWMF	PWM Interrupt Flag. This flag bit is cleared by software.			
CLR_P0	Counter Reset Enable. This flag bit is cleared by hardware.			
RUN_P0	Counter Start Enable.			

13.28 **PWMD** (DEh) : PWM Duty Data Register

Bit No.	7	6	5	4	3	2	1	0
	PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

PWMD duty data register, located in DEh, determines the duty of the output value generated by each 8-bit PWM circuit. To program the required PWM output, you load the appropriate initialization values into the 8-bit data register (PWMD).

13.29 **ACC/A** (E0h) : Accumulator

Bit No.	7	6	5	4	3	2	1	0
	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register serves as the accumulator for arithmetic operations.

13.30 ADCSELH (E1h) : ADC Channel Selection High Register

Bit No.	7	6	5	4	3	2	1	0
	ADC11B	ADC10B	ADC9B	ADC8B	ADC7B	ADC6B	ADC5B	ADC4B
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

Symbol	Description	
ADC4B ADC5B ADC6B ADC7B ADC9B ADC10B ADC11B	ADC Channel Selection Low. These bits are configured to enable/disable the ADC channel inputs (ADC[11:4]). P0[5] ~ P0[7] : ADC4 ~ ADC6 input.	
	ADC Channel	Result
	ADC4B	1 = ADC4 input channel is disabled & digital input is enabled at P0[5]. (Default) 0 = ADC4 input channel is enabled & digital input is disabled.
	ADC5B	1 = ADC5 input channel is disabled & digital input is enabled at P0[6]. (Default) 0 = ADC5 input channel is enabled & digital input is disabled.
	ADC6B	1 = ADC6 input channel is disabled & digital input is enabled at P0[7]. (Default) 0 = ADC6 input channel is enabled & digital input is disabled.
	ADC7B	1 = ADC7 input channel is disabled & digital input is enabled at P2[6]. (Default) 0 = ADC7 input channel is enabled & digital input is disabled.
	ADC8B	1 = ADC8 input channel is disabled & digital input is enabled at P2[5]. (Default) 0 = ADC8 input channel is enabled & digital input is disabled.
	ADC9B	1 = ADC9 input channel is disabled & digital input is enabled at P2[4]. (Default) 0 = ADC7 input channel is enabled & digital input is disabled.
	ADC10B	1 = ADC10 input channel is disabled & digital input is enabled at P2[3]. (Default) 0 = ADC10 input channel is enabled & digital input is disabled.
	ADC11B	1 = ADC11 input channel is disabled & digital input is enabled at P2[2]. (Default) 0 = ADC11 input channel is enabled & digital input is disabled.

13.31 ADCSEL (E2h) : ADC Channel Selection Low & MUX Selection Register

Bit No.	7	6	5	4	3	2	1	0
	ADC3B	ADC2B	ADC1B	ADC0B	CH3	CH2	CH1	CH0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

<table>
<tr><th>Symbol</th><th colspan="6">Description</th></tr>
<tr><td rowspan="5">ADC0B
ADC1B
ADC2B
ADC3B</td><td colspan="6">ADC Channel Selection Low.
These bits are configured to enable/disable the ADC channel inputs (ADC[3:0]). P0[1] ~ P0[4] : ADC0 ~ ADC3 input. (Default = 1)</td></tr>
<tr><td>ADC Channel</td><td colspan="5">Result</td></tr>
<tr><td>ADC0B</td><td colspan="5">1 = ADC0 input channel is disabled & digital input is enabled at P0[1].
0 = ADC0 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC1B</td><td colspan="5">1 = ADC1 input channel is disabled & digital input is enabled at P0[2].
0 = ADC1 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC2B</td><td colspan="5">1 = ADC2 input channel is disabled & digital input is enabled at P0[3].
0 = ADC2 input channel is enabled & digital input is disabled.</td></tr>
<tr><td>ADC3B</td><td colspan="5">1 = ADC3 input channel is disabled & digital input is enabled at P0[4].
0 = ADC3 input channel is enabled & digital input is disabled.</td></tr>
<tr><td rowspan="12">CH0
CH1
CH2
CH3</td><td colspan="6">ADC MUX Selection.
Theses bits is configured to select the ADC input. The initial value of CH[3:0] is 0xFh (4’b1111). Ch, Dh, Eh and Fh means that all ADC inputs are disabled.</td></tr>
<tr><td>CH3</td><td>CH2</td><td>CH1</td><td>CH0</td><td colspan="2">Result</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">0x0h : ADC0 is selected</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td colspan="2">0x1h : ADC1 is selected</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="2">0x2h : ADC2 is selected</td></tr>
<tr><td colspan="4">…</td><td colspan="2"></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td colspan="2">0xAh : ADC10 is selected</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td colspan="2">0xBh : ADC11 is selected</td></tr>
<tr><td colspan="4">1100 (0xCh) ~ 1111 (0xFh)</td><td colspan="2">No ADC channel is not selected (Default Value = 0xFh)</td></tr>
</table>

13.32 ALTSEL (E3h) : Alternative Function Selection Register

Bit No.	7	6	5	4	3	2	1	0
	IOXEN	IORSTEN	CLO	PWM00	TVO	TX	-	-
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)		

Symbol	Description
IOXEN	I/O bits from XTAL1/XTAL2 Enable. (XTAL1 : Input, XTAL2 : Output) 0 = The P1[1] and P1[0] bits are configured as XTAL Input/Output (Default) 1 = The P1[1] and P1[0] bits are configured as general I/O pins. Note that you must set XTOFF bit (PMR.3) to "1" to turn off the internal amplifier.
IORSTEN	I/O bits from RESETB Enable. 0 = The P1[2] bit is configured as RESETB input. (Default) 1 = The P1[2] bit is configured as general I/O pin.
CLO	System Clock Output Enable to P2[6]. 0 = Disable the System Clock Output to P2[6]. (Default) 1 = Enable the System Clock Output to P2[6].
PWM00	PWM Waveform Output Enable to P0[0]. 0 = Disable the PWM waveform output to P0[0]. (Default) 1 = Enable the PWM waveform output to P0[0].
TVO	Timer0 Overflow Output Enable to P0[0]. 0 = Disable the Timer0 overflow output to P0[0]. (Default) 1 = Enable the Timer0 overflow output to P0[0].
TX	UART TX Data Output Enable to P0[2]. (TXD : TX Data) 0 = Disable the TX data output to P0[2]. (Default) 1 = Enable the TX data output to P0[2]. Note that the port directional mode (input/output) of P0[2] for TXD is don't care. But the directional mode (input/output) of P0[1] for RXD must set up to input mode, When user is setting to UART communication.
-	Reserved

13.33 P0SEL (E4h) : Port 0 Pull-up Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0SEL.7	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register is enables/disabled the internal pull-up in P0.

0 = Pull-up ON. The bit is configured to pull-up ON. (Default)

1 = Pull-up OFF. The bit is configured to pull-up OFF.

13.34 P1SEL (E5h) : Port 1 Pull-up Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	P1SEL.1	P1SEL.0
							R/W(1)	R/W(1)

This register is enables/disabled the internal pull-up in P1. Note that the P1[2] has no pull-up.

0 = Pull-up ON. The bit is configured to pull-up ON.

1 = Pull-up OFF. The bit is configured to pull-up OFF. (Default)

13.35 P2SEL (E6h) : Port 2 Pull-up Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0
		R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register is enables/disabled the internal pull-up in P2.

0 = Pull-up ON. The bit is configured to pull-up ON. (Default)

1 = Pull-up OFF. The bit is configured to pull-up OFF.

13.36 **EIE** (E8h) : Extended Interrupt Enable Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	EPWM	EWDT	-	-	EX3	EX2
			R/W(0)	R/W(0)			R/W(0)	R/W(0)

Symbol	Description
-	Reserved.
EPWM	PWM Interrupt Enable. This bit controls the masking of the PWM interrupt. 0 = Disable the PWM interrupt. (Default) 1 = Enable the PWM interrupt.
EWDT	WDT (Watchdog Timer) Enable. This bit controls the masking of the WDT interrupt. 0 = Disable the WDT interrupt. (Default) 1 = Enable the WDT interrupt.
EX3	External Interrupt 3 Enable. This bit controls the masking of the external interrupt 3. 0 = Disable the external interrupt 3. (Default) 1 = Enable the external interrupt 3 (INT3B).
EX2	External Interrupt 2 Enable. This bit controls the masking of the external interrupt 2. 0 = Disable the external interrupt 2. (Default) 1 = Enable the external interrupt 2 (INT2 pin).

13.37 **ADCR** (EEh) : ADC Result High Register

Bit No.	7	6	5	4	3	2	1	0
	SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

The MSB (ADC result value [9:2]) of ADC conversion result (total 10 bits) is stored into ADCR. LSB of conversion result will be stored into ADCON[1:0].

13.38 **ADCON** (EFh) : ADC Control & ADC Result Low Register

Bit No.	7	6	5	4	3	2	1	0
	AD_EN	AD_REQ	AD_END	ADCF	AVREF	ADIV	SAR1	SAR0
	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

Symbol	Description
AD_EN	ADC Ready Enable.
AD_REQ	ADC Start Enable. Cleared by H/W when AD_END goes to "1" from "0".
AD_END	Current ADC Status. Set by hardware when ADC conversion has been finished. Note that user should not use the AD_END but ADCF for finish of ADC conversion. 0 = ADC is running now. 1 = ADC is idle state. ADC is stopped. (Default)
ADCF	ADC Interrupt Flag. This bit must be cleared by software.
AVREF	ADC Reference Voltage Input Enable from P0[4]. 0 = ADC reference voltage from V_{DD} (System Power). (Default) 1 = ADC reference voltage from P0[4].
ADIV	ADC Input Clock Select. 0 = System Clock (F_{OSC}) / 2. (Default) 1 = PWM Input Clock (F_{PWM}).
SAR1 SAR0	ADC Result Value [1:0]. (Low bits of ADC result value)

13.39 **B** (F0h) : B Register

Bit No.	7	6	5	4	3	2	1	0
	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)

This register serves as a second accumulator for certain arithmetic operations.

13.40 P0DIR (F4h) : Port 0 Input/Output Control Register

Bit No.	7	6	5	4	3	2	1	0
	P0DIR.7	P0DIR.6	P0DIR.5	P0DIR.4	P0DIR.3	P0DIR.2	P0DIR.1	P0DIR.0
	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register is configured the P0 to input mode or output mode. The initial P0 mode is configured to input mode.

0 = Output mode. The bit is configured to output.

1 = Input mode. The bit is configured to input. (Default)

13.41 P1DIR (F5h) : Port 1 Input/Output Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	-	-	-	P1DIR.2	P1DIR.1	P1DIR.0
						R/W(1)	R/W(1)	R/W(1)

This register is configured the P1 to input mode or output mode. The initial P1 mode is configured to input mode.

0 = Output mode. The bit is configured to output.

1 = Input mode. The bit is configured to input. (Default)

13.42 P2DIR (F6h) : Port 2 Input/Output Control Register

Bit No.	7	6	5	4	3	2	1	0
	-	P2DIR.6	P2DIR.5	P2DIR.4	P2DIR.3	P2DIR.2	P2DIR.1	P2DIR.0
		R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)

This register is configured the P2 to input mode or output mode. The initial P2 mode is configured to input mode.

0 = Output mode. The bit is configured to output.

1 = Input mode. The bit is configured to input. (Default)

13.43 EIP (F8h) : Extended Interrupt Priority Register

Bit No.	7	6	5	4	3	2	1	0
	-	-	PPWM	PWDT	-	-	PX3	PX2
			R/W(0)	R/W(0)			R/W(0)	R/W(0)

Symbol	Description
-	Reserved.
PPWM	PWM Interrupt Priority. This bit controls the priority of the PWM interrupt up to two levels. 0 = Priority level 0 for PWM interrupt. (Default) 1 = Priority level 1 for PWM interrupt. Note : The priority level 0 means the low priority interrupt level. And, the priority level 1 means the high priority interrupt level.
PWDT	WDT (Watchdog Timer) Priority. This bit controls the priority of the WDT up to two levels. 0 = Priority level 0 for WDT interrupt. (Default) 1 = Priority level 1 for WDT interrupt.
PX3	External Interrupt 3 Priority. This bit controls the priority of the external interrupt 3 up to two levels. 0 = Priority level 0 for external interrupt 3. (Default) 1 = Priority level 1 for external interrupt 3.
PX2	External Interrupt 2 Priority. This bit controls the priority of the external interrupt 2 up to two levels. 0 = Priority level 0 for external interrupt 2. (Default) 1 = Priority level 1 for external interrupt 2.