CORERIVER

MiDAS1.0 Family: Mask ROM/EPROM/ROMless 8-bit Turbo Microcontrollers

Preliminary

Rev. 1.6

August 2007

Copyright CORERIVER Semiconductor Co., Ltd. 2007

All Rights Reserved

Downloaded from Elcodis.com electronic components distributor

- *CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time.*
- *CORERIVER shall give customers at least a three advance notice of intended discontinuation of a product or a service through its homepage.*
- *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.*

Preliminary

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

Table of Contents

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

List of Figures

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

List of Tables

Downloaded from Elcodis.com electronic components distributor

1 Product Overview

CORERIVER's MiDAS1.0 family is a group of fast 80C52 compatible microcontrollers without wasted clock and memory cycles. Four clocks per one machine cycle are consumed in the redesigned processor core of the MiDAS 1.0 Family. As a result, all its 8052 instruction is executed about 3 times faster than that of traditional 80C52.

The MiDAS1.0 family offers three timer/counters, a serial port, maximum 36 I/O ports, four-channel 9-bit ADC (Analog to Digital Converter), two-channel 8-bit PWM (Pulse Width Modulator), a Watchdog timer, and LVD (Low Voltage Detector) as peripherals.

The MiDAS1.0 family provides power saving modes for the power-critical applications and also does noise tolerant scheme.

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

1.1 Ordering Information

1.1.1 MiDAS1.0 Family

Table 1-1 MiDAS1.0 Family – GC80C520A Series (ADC Application MCU)

Product	Mask ROM (byte)	EPROM (byte)	RAM (byte)	Voltage (V)	Frequency (MHz)	ADC (bit x ch)	PWM (bit x ch)	I / O (pin)	Package
GC87C520A0		8K	256	2.7~5.5	40	9x4	8x2	36	44-PLCC
								36	44-MQFP
								32	40-PDIP
								22	28-SPDIP
								22	28-SOIC
GC81C520A0	8K							36	44-PLCC
								36	44-MQFP
								32	40-PDIP
								22	28-SPDIP
								22	28-SOIC
GC80C320A0	ROMless							36	44-PLCC
								36	44-MQFP
								32	40-PDIP

Downloaded from Elcodis.com electronic components distributor

2 Features

- 8-bit Turbo 80C52 Architecture
- 4 cycles/1 machine cycle
- Pin/instruction level compatible with Intel 80C52
- 0/8Kbyte Mask ROM
- 0/8Kbyte Programmable ROM (PROM)
- 256byte RAM
- Supply voltage: 2.7V ~ 5.5V
- Operating Frequency
  - ✓ Max. 40MHz @ 4.5 ~ 5.5V
  - ✓ Max. 20MHz @ 2.7 ~ 3.3V
- Operating temperature: -20 °C ~ 85 °C
- Max. 36 Programmable I/O pins
  - ✓ Intel 80C52 compatible by default
  - ✓ Input/Output direction programmable
  - ✓ TTL input level and CMOS compatible logic levels
- EMI reduction mode: Inhibit ALE
- Low Voltage Detector (LVD)
- 27-bit Programmable Watchdog Timer (WDT)
- Three 16-bit Timer/Counters
- Full-Duplex Programmable UART
  - ✓ Automatic address recognition
- 2-channel 8-bit high speed Pulse Width Modulator (PWM)
- 4-channel 9-bit Analog to Digital Converter (ADC)
  - ✓ Max. 100K samples per second @ 8MHz
  - ✓ Programmable input clock frequency
- 13 interrupt sources including 6 external ones
  - ✓ Timer 0/1/2, UART, ADC, WDT, LVD and six external
  - ✓ Four-level interrupt priority and NMI (Non-Maskable Interrupt)
- Reset scheme
  - ✓ On-chip Power-On-Reset (POR)
  - ✓ External reset
  - ✓ Low Voltage Detector (LVD) reset
  - ✓ Watchdog timer reset
- Power consumption
  - ✓ Active current: Max 20mA @ 5V, 40MHz
  - ✓ Stop current: Max 1uA
- ESD protection up to 2,000V
- Latch-up protection up to ±200mA
- Package : 44-PLCC, 44-MQFP, 40-PDIP, 28-SPDIP, 28-SOIC

Downloaded from Elcodis.com electronic components distributor

3 Block Diagram

Figure 3-1 shows the block diagram of MiDAS1.0 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.

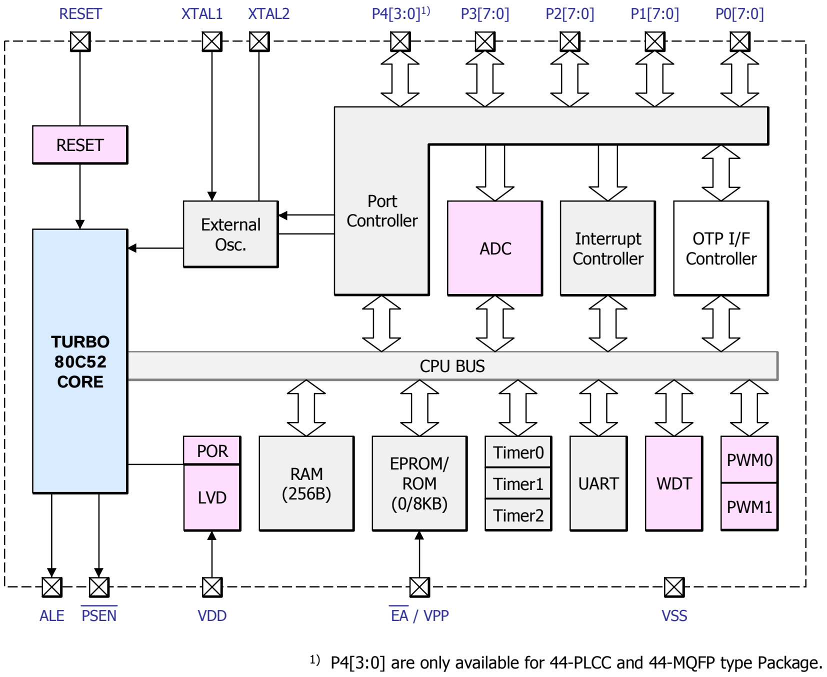
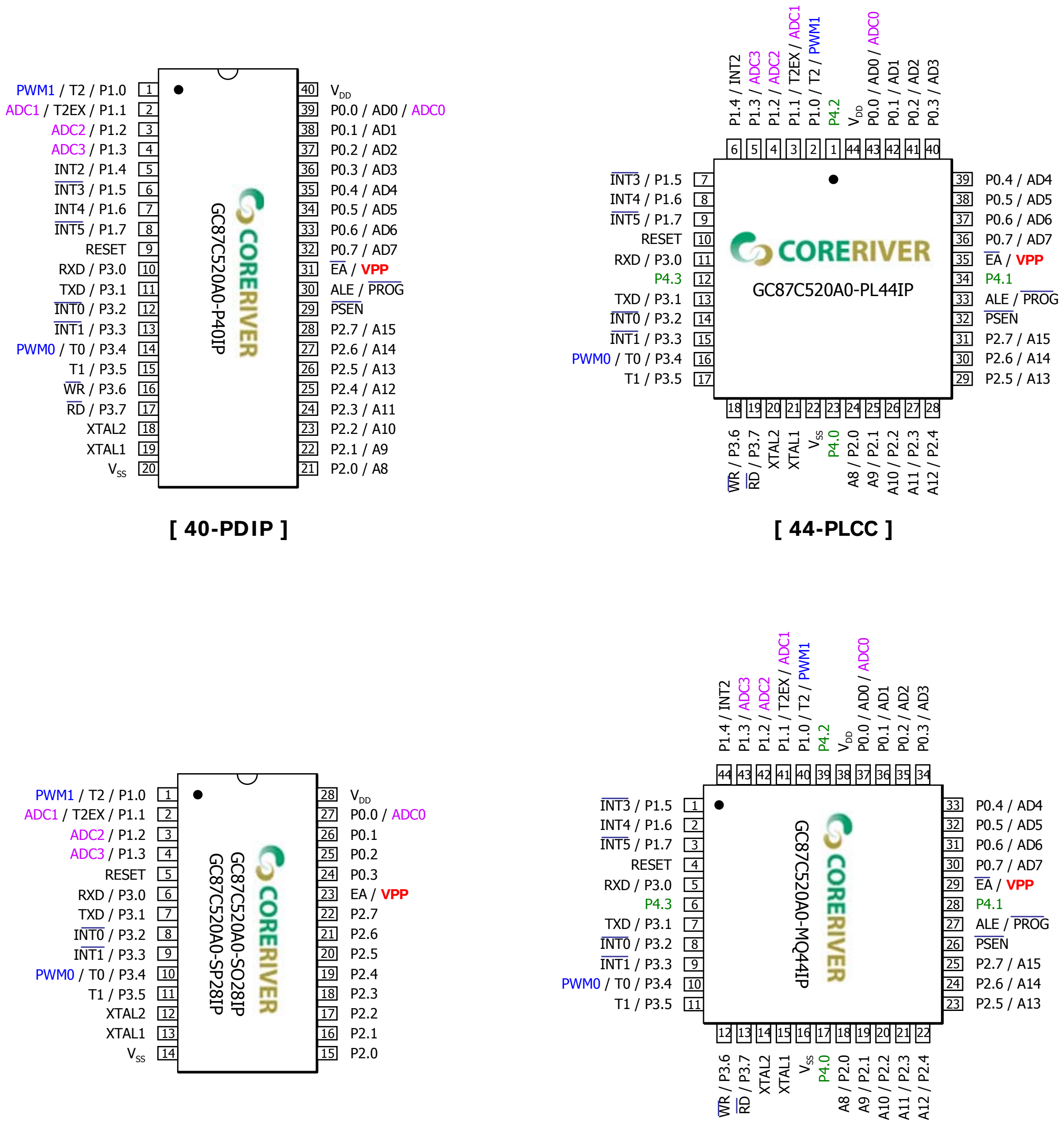


Figure 3-1 Block Diagram

Downloaded from Elcodis.com electronic components distributor

4 Pin Configurations

The MiDAS1.0 family supports various packages, e.g. 40-pin DIP, 44-pin PLCC, 44-pin MQFP, 28-SPDIP/SOIC. The pin configurations are shown in Figure 4-1.



[40-PDIP]

[44-PLCC]

[28-SPDIP/SOIC]

[44-MQFP]

Figure 4-1 Pin Configuration

Downloaded from Elcodis.com electronic components distributor

5 Pin Description

Table 5-1 Pin Description

Symbol	Type	Description	Alternate Pins
VDD	Input	Power supply	-
VSS	Input	Ground	-
RESET	Input	External reset	-
XTAL1	Input	Input to the inverting oscillator amplifier	-
XTAL2	Output	Output from the inverting oscillator amplifier	-
/EA	Input	External Access Enable (0 → fetched from external ROM, 1 → from internal ROM). This pin must not be floating.	VPP
ALE	In/Out	Address latch enable	/PROG
/PSEN	In/Out	Program strobe enable (External ROM output enable signal)	-
P0[7:0]	In/Out	Programmable I/O Port: Schmitt Trigger Input or Push-pull/Open-Drain Output. Pull-up Resistors can be switched on/off by Software.	P0.0/AD0/ADC0 P0.1~P0.7 AD1~7
P1[7:0]	In/Out	Programmable I/O Port: Schmitt Trigger Input or Push-pull Output. Pull-up Resistors can be switched on/off by Software.	P1.0/T2/PWM1 P1.1/T2EX/ADC1 P1.2~P1.3/ADC2~ADC3 P1.4~P1.7/INT2~INT5
P2[7:0]	In/Out	Programmable I/O Port: Schmitt Trigger Input or Push-pull Output. Pull-up Resistors can be switched on/off by Software	P2.0~P2.7/A8~A15
P3[7:0]	In/Out	Programmable I/O Port: Schmitt Trigger Input or Push-pull Output. Pull-up Resistors can be switched on/off by Software.	P3.0/RXD, P3.1/TXD P3.2~P3.3/INT0~INT1 P3.4/T0/PWM0 P3.5/T1, P3.6/WR P3.7 / RD
P4[3:0]	In/Out	Programmable I/O Port: Schmitt Trigger Input or Push-pull Output. Pull-up Resistors can be switched on/off by Software.	-

Downloaded from Elcodis.com electronic components distributor

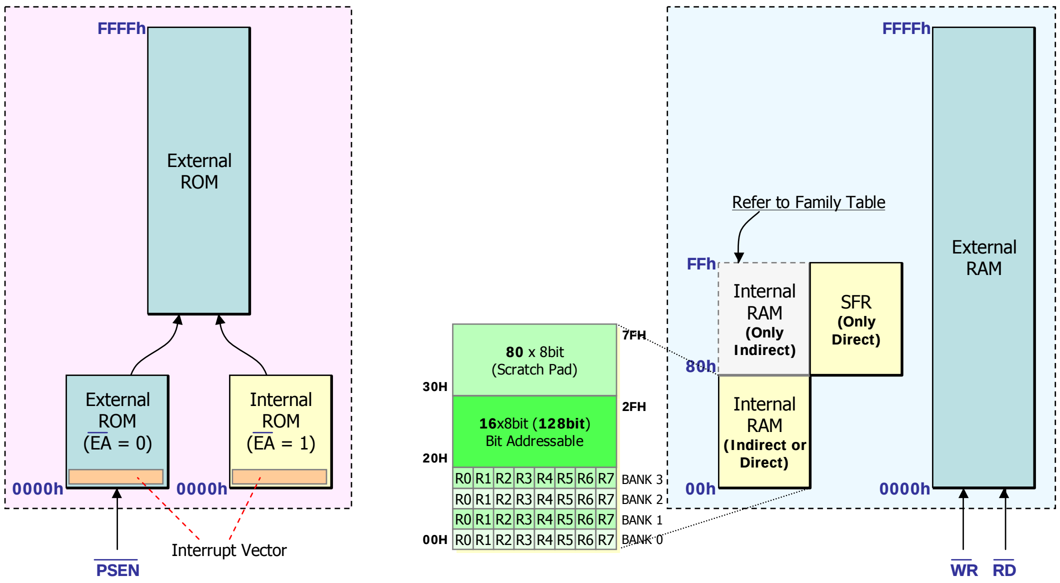
6 Functional Description

6.1 CPU Description

6.1.1 Memory Organization

MiDAS1.0 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 64K bytes of Program Memory.


FFFFh
External ROM
External ROM (EA = 0)
Internal ROM (EA = 1)
0000h
0000h
PSEN
Interrupt Vector
80 x 8bit (Scratch Pad)
16x8bit (128bit) Bit Addressable
7FH
30H
2FH
20H
00H
R0 R1 R2 R3 R4 R5 R6 R7 BANK 3
R0 R1 R2 R3 R4 R5 R6 R7 BANK 2
R0 R1 R2 R3 R4 R5 R6 R7 BANK 1
R0 R1 R2 R3 R4 R5 R6 R7 BANK 0
Refer to Family Table
FFh
Internal RAM (Only Indirect)
SFR (Only Direct)
80h
Internal RAM (Indirect or Direct)
00h
FFFFh
External RAM
0000h
WR
RD

[Program Memory]
(Read Only)

[Data Memory]
(Read and Write)

Figure 6-1 Memory Organization

6.1.1.1 Program Memory

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

Downloaded from Elcodis.com electronic components distributor

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, for example, is assigned to the 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 used Program Memory may be less than 64K bytes.

Program Memory can only be read, not written to. There can be up to 64K 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 diagram of Figure 6-1:

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

While the upper internal RAM area and the SFR area share the same address locations (80h – FFh), they are accessed through different addressing modes. The upper internal RAM can only be accessed through indirect addressing mode (indirect addressing with general purpose registers R0 or R1), and the special function registers (SFRs) are accessible only by direct addressing mode (address is part of the instruction).

6.1.1.2.1 Register Banks

The lowest 32 bytes of the internal data RAM are grouped into 4 banks of 8 general purpose registers (GPRs). Program instructions call out these registers as R0 through R7. RS1 and RS0 bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient

Downloaded from Elcodis.com electronic components distributor

use of code space, since register instructions are shorter than instructions that use direct addressing.

6.1.1.2.2 Bit-addressable Memory Space

The next 16 bytes, locations 20H through 2FH, above the register banks form a block of bit-addressable memory space. There are 128 bits in this area. The 128 bits can be directly addressed by single-bit instructions. The bit addresses in this area are 00H through 7FH.
Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 000B. The bit addresses in this area are 80H through FFH.

6.1.2 Special Function Registers (SFRs) Map and Description

The MiDAS1.0 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 MiDAS1.0 family contains all the SFRs present in the standard 80C52. However, some additional SFRs have been added. In some cases unused bits in the original 80C52 have been given new functions. The list of SFRs is as follows. Refer to Appendix B for more details about SFRs.

Downloaded from Elcodis.com electronic components distributor

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

🟦 : Newly added SFR at MiDAS1.0 Family

🟧 : Reserved for future use.

Bit addressable

F8h	EIP								FFh
F0h	B								F7h
E8h	EIE						ADCR	ADCON	EFh
E0h	ACC		ADCSEL	[1)]ALTSEL	P0SEL	P1SEL	P2SEL	P3SEL	E7h
D8h	WDCON				PWM0CON	PWM1CON	PWM0D	PWM1D	DFh
D0h	PSW								D7h
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2			CFh
C0h					PMR	STATUS			C7h
B8h	IP	SADEN							BFh
B0h	P3							IPH	B7h
A8h	IE	SADDR							AFh
A0h	P2					P4	P4SEL		A7h
98h	SCON	SBUF							9Fh
90h	P1	EXIF							97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON		8Fh
80h	P0	SP	DPL	DPH				PCON	87h

Note: Newly added SFRs at MiDAS1.0 family are written in bold style.

Note: Don't use ALTSEL[1)] in 40 PDIP/44 PLCC/44 MQFP/28PDIP/28SOP.

Note: SFR P4 is available only in 44 PLCC/44 MQFP package.

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

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	00*10000	No

Downloaded from Elcodis.com electronic components distributor

T/C Control	TCON	88h	00000000	Yes
T/C Mode Control	TMOD	89h	00000000	No
T/C 0 Low Byte	TL0	8Ah	00000000	No
T/C 1 Low Byte	TL1	8Bh	00000000	No
T/C 0 High Byte	TH0	8Ch	00000000	No
T/C 1 High Byte	TH1	8Dh	00000000	No
Watchdog Timer and 4-cycle switching Control	CKCON	8Eh	00000***	No
Port 1	P1	90h	11111111	Yes
Added external interrupt and LVD control	EXIF	91h	00001**1	No
Serial Control	SCON	98h	00000000	Yes
Serial Buffer	SBUF	99h	00000000	No
Port 2	P2	A0h	11111111	Yes
Port 4	P4	A5h	****1111	No
Port 4 pull-up on/off Control	P4SEL	A6h	****0000	No
Interrupt Enable control	IE	A8h	00000000	Yes
Slave Address	SADDR	A9h	00000000	No
Port 3	P3	B0h	11111111	Yes
Interrupt Priority High	IPH	B7h	10000000	No
Interrupt Priority Low	IP	B8h	10000000	Yes
Slave Address Mask Enable	SADEN	B9h	00000000	No
Power Management Control	PMR	C4h	*****0**	No
Crystal Status	STATUS	C5h	***0****	No
T/C2 Control	T2CON	C8h	00000000	Yes
T/C 2 Mode selection	T2MOD	C9h	******00	No
T/C 2 Capture Low Byte	RCAP2L	CAh	00000000	No
T/C 2 Capture High Byte	RCAP2H	CBh	00000000	No
T/C 2 Low Byte	TL2	CCh	00000000	No
T/C 2 High Byte	TH2	CDh	00000000	No
Program Status Word	PSW	D0h	00000000	Yes
Power flag and Watchdog Timer Control	WDCON	D8h	*1010000	Yes
PWM 0 Control	PWM0CON	DCh	00000000	No
PWM 1 Control	PWM1CON	DDh	00000000	No
PWM 0 Duty Data	PWM0D	DFh	00000000	No

Downloaded from Elcodis.com electronic components distributor

PWM 1 Duty Data	PWM1D	DFh	00000000	No
Accumulator	ACC	E0h	00000000	Yes
ADC channel selection and ADC input clock divide control	ADCSEL	E2h	000*0000	No
Alternate function Selection	ALTSEL	E3h	**000000	No
Port 0 Pull-up Control	P0SEL	E4h	11111111	No
Port 1 Pull-up Control	P1SEL	E5h	00000000	No
Port 2 Pull-up Control	P2SEL	E6h	00000000	No
Port 3 Pull-up Control	P3SEL	E7h	00000000	No
Extended Interrupt Enable	EIE	E8h	*0000000	Yes
ADC Result value including MSB bit	ADCR	EEh	00000000	No
ADC Control & LSB bit of ADC Result value	ADCON	EFh	001000*0	No
B Register	B	F0h	00000000	Yes
Extended Interrupt Priority	EIP	F8h	*0000000	Yes

CAUTION: Don’t modify * bits. Modifying these bits can cause the malfunctions. Especially the following bits of SFRs should not be updated.

- EXIF.3(XT) : This bit must always remain high which is the default value.
- PMR.3(*) : This bit should not be modified.

6.1.3 Instruction Set Summary

The MiDAS1.0 family executes all the instructions of the standard 80C52. The instructions of the MiDAS1.0 family produce the same result with those of the standard 80C52.

The MiDAS1.0 family has a special internal architecture and different timing to those of 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.

Downloaded from Elcodis.com electronic components distributor

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
	MOVX	Move data to external RAM
	PUSH	Push
	POP	Pop
	XCH	Exchange
	XCHD	Exchange low-digit
Boolean	CLR	Clear bit
	SETB	Set bit
	CPL	Complement bit
	ANL	AND bit
	ORL	OR bit
	MOV	Move bit

Downloaded from Elcodis.com electronic components distributor

	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 MiDAS1.0 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.

Downloaded from Elcodis.com electronic components distributor

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 lower 128 bytes of the internal RAM can be addressed in this mode. It is the only method of accessing the SFRs.

Examples:

MOV A, TCON	TCON is the 8-bit address of the SFR TCON.
MOV R4, variable	The 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 or 16-bit address that is stored in a register. For 8-bit addresses, the content of either R0 or R1 (in the selected register bank) is used. For 16-bit addresses, 16-bit wide data pointer (DPTR) can be used.

Note: The lower and upper part of the internal RAM can be addressed by using 8-bit addresses. The external data memory can be addressed by 16-bit addresses.

Examples:

MOV A, @R0	RAM is addressed by contents of R0 (8-bit address)
MOVX A, @DPTR	External data memory is addressed by contents of the selected data pointer DPTR (16-bit address)

Downloaded from Elcodis.com electronic components distributor

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 data. 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 data and the accumulator data.

Examples:

MOVC A, @A + DPTR	The address is the sum of DPTR and accumulator.
MOVC A, @A + PC	The 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 MiDAS1.0 family is an important aspect, especially for those who wish to use software instructions to generate timing delays. Also, it provides the user with an insight into the timing differences between the MiDAS1.0 family and the standard 80C52 as shown in Figure 6-2. In the MiDAS1.0 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

Downloaded from Elcodis.com electronic components distributor

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 MiDAS1.0 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.

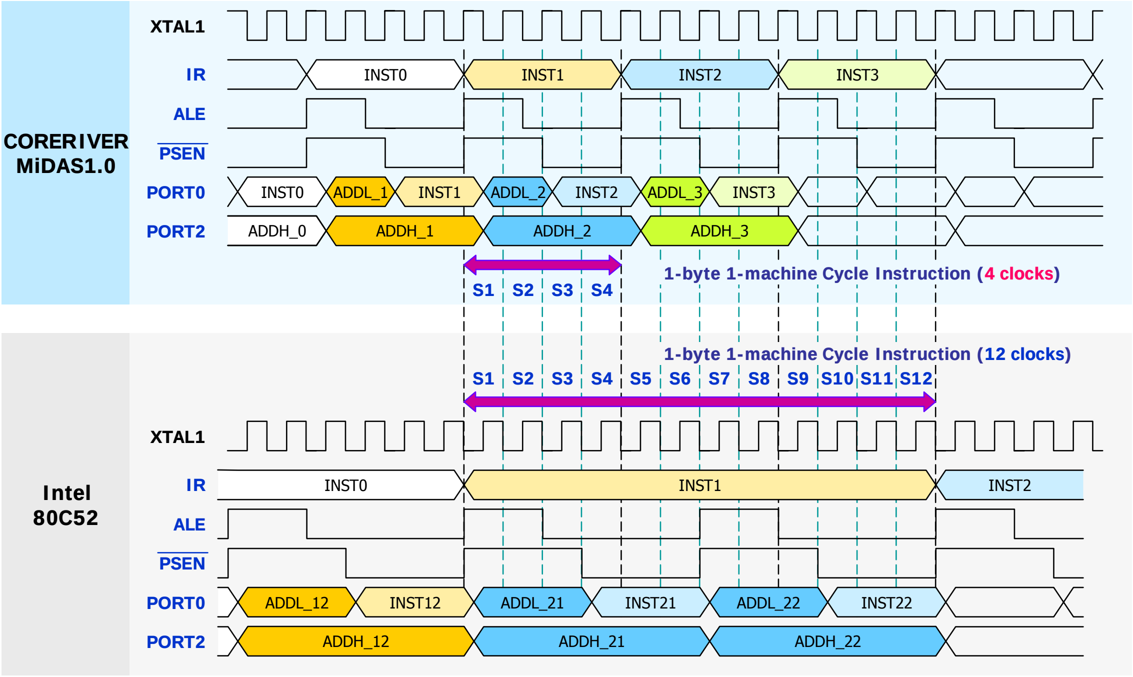


Figure 6-2 Comparative Timing of the MiDAS1.0 family and Intel 80C52

6.1.4.1 MOVX Instruction

The MiDAS1.0 family, like the standard 80C52, uses the MOVX instruction to access external Data Memory. This Data Memory includes both off-chip memory as well as memory-mapped peripherals. While the results of the MOVX instruction are the same as in the standard 80C52, the operation and the timing of the strobe signals have been modified in order to give the user much greater flexibility as shown in Figure 6-3 and Figure 6-4.

The MOVX instruction is of two types: the MOVX @Ri and MOVX @DPTR. In the MOVX @Ri, the address of the external data comes from two sources. The lower 8-bits of the address are stored in the Ri register of the selected working register bank. The upper 8-bits of the address come from the port 2 SFR.

Downloaded from Elcodis.com electronic components distributor

In the MOVX @DPTR type, the full 16-bit address is supplied by the Data Pointers.

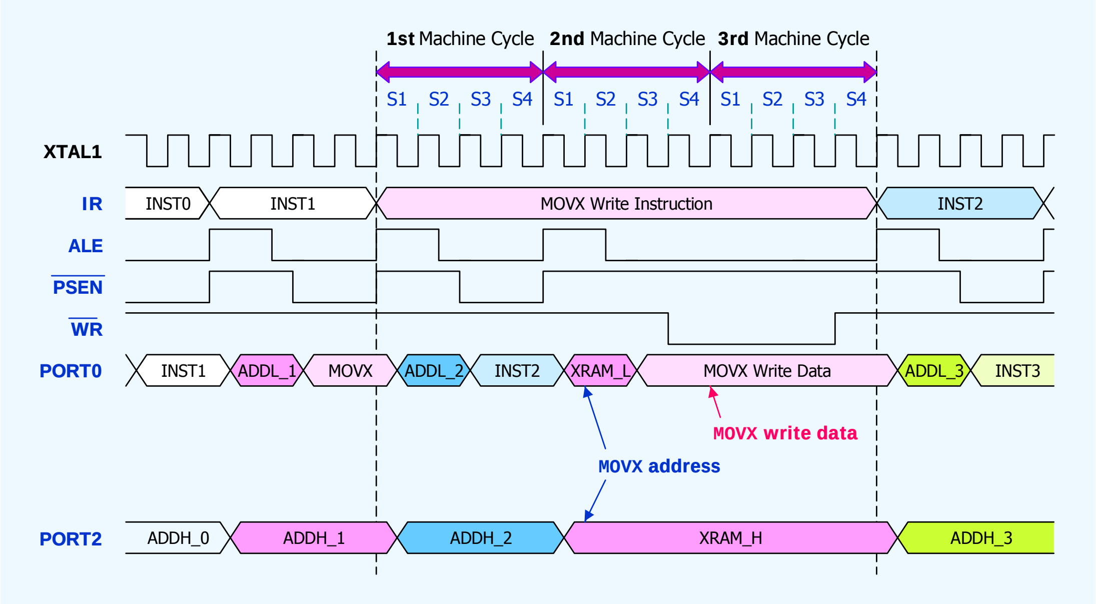


Figure 6-3 Write Timing of MOVX instruction

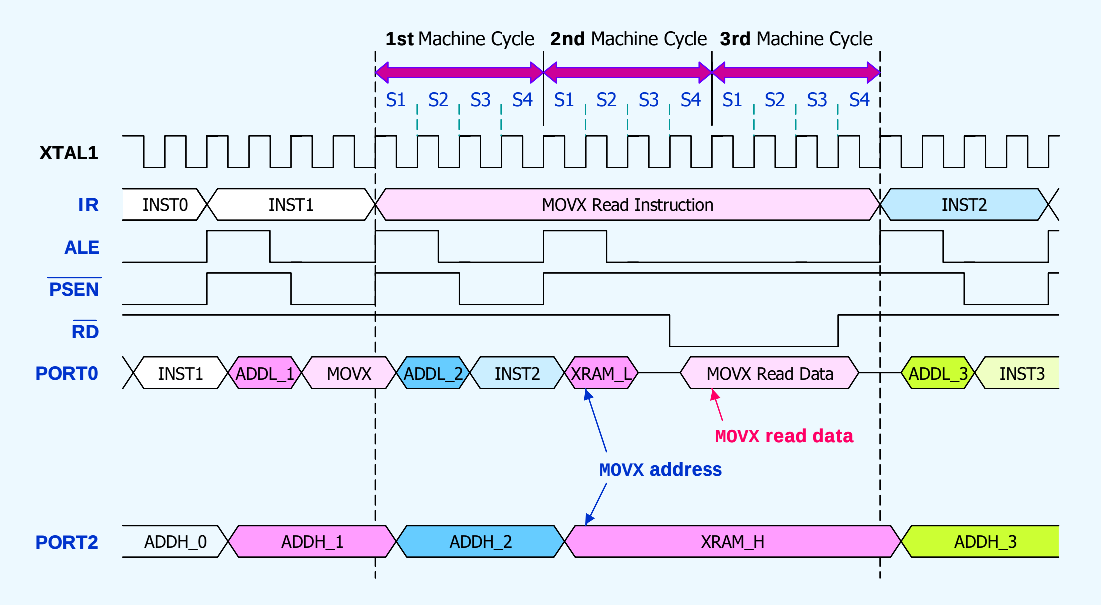


Figure 6-4 Read Timing of MOVX instruction

Downloaded from Elcodis.com electronic components distributor

6.2 Peripheral Description

6.2.1 I/O Ports

The I/O ports of MiDAS1.0 family have been developed so as to maintain the compatibility with those of the standard 80C52. So it is not necessary to modify the design of existing applications in order to replace a standard 80C52 product by the MiDAS1.0 family. All output ports of MiDAS1.0 have the compatible logic level and at least equal drive capability compared to those of the standard 80C52. As a result, their logic status transitions are similar but a little different. Due to the difference, the MiDAS1.0 family can improve speed and slew rates.

The MiDAS1.0 family provides bi-directional I/O ports. Each I/O port consists of a latch (Special Function Register P0 through P4), an output driver, and an input buffer. Except P4, all the latches are 8-bits, byte-addressable, and bit-addressable. The P4 latch is 4-bits and byte-addressable. A bit instruction has a different opcode from a byte instruction.

The output drivers of Ports 0 and 2 and the input buffers of Port 0 are used for accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

The ROMless devices of the MiDAS1.0 family use Ports 0 and 2 only for accesses to external memory. So their Port 0 does not include a latch.

6.2.1.1 PORT 0

Port 0 is an 8-bit, open-drain, bi-directional I/O port with internal pull-up resistors. The pull-up devices are switched on/off by the value of the SFR P0SEL. The Port 0 of the devices including internal program memory is available as general purpose I/O. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. If the internal pull-up resistors are switched on, Port 0 pins are pulled high and can be used as inputs.

It may not be recommended that even a device including program memory should use its Port 0 as a general purpose I/O port if it also accesses external memory. It is because the CPU will write FFH to the

Downloaded from Elcodis.com electronic components distributor

port 0 latch during any access to external memory. The internal pull-up resistors are switched on/off by the value of P0SEL. Only P0.0 can be used as an analog input of an A/D converter. It is assigned as analog inputs via the bit 0 of the register ADCSEL.

When used for accessing external memory, Port 0 provides the data byte time-multiplexed with the low byte of the address. In this state, Port 0 is disconnected from its own port latch, and the address/data signal drives push-pull FETs in its output buffer. In this application, Port 0 uses strong internal pull-ups when emitting 1s. During any access to external memory, the CPU writes FFH to the port 0 latch, thus obliterating whatever information the port SFR may have been holding. When maximizing the slew rate, special protection circuits prevent ringing, excessive noise, and interface problems. No external pull-up resistor is needed since it degrades the memory access timing.

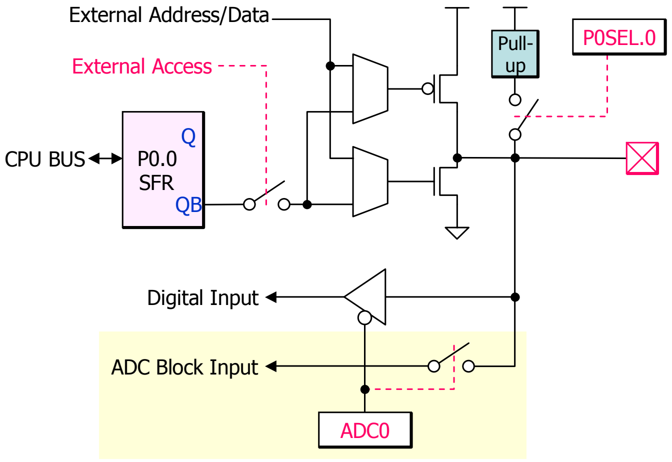


Figure 6-5 Configuration of PORT 0

Note: During external access, P0 SFR will automatically set to “FFh”.

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

6.2.1.2 PORT 1

Port 1 is an 8-bit bi-directional I/O port with internal pull-up resistors. The pull-up resistors are switched on/off by the value of the SFR P1SEL. Port 1 pins that have 1s written to them are pulled high weakly by the internal pull-up resistors and can be used as inputs. Port 1 pins can be also used alternate functions. To use the alternate functions, the corresponding bit of the Port 1 pin should be set to high. The alternate

Downloaded from Elcodis.com electronic components distributor

functions will be selected according to the combination of the SFR values.

The drive characteristics of an alternate function are the same with those of general I/O. The logic 0 output pulls a pin low strongly. The logic 1 output changes a pin status with a strong pull-up and maintains the logic 1 status with a weak pull-up. More than one port pins are assigned to alternate functions without changing the functions of the other pins.

Note: P1.1/P1.2/P1.3 are assigned to be used as analog input pins (ADC input channel 1/2/3) by the bits ADC1/ADC2/ADC3 (ADCSEL.1/ADCSEL.2/ADCSEL.3).

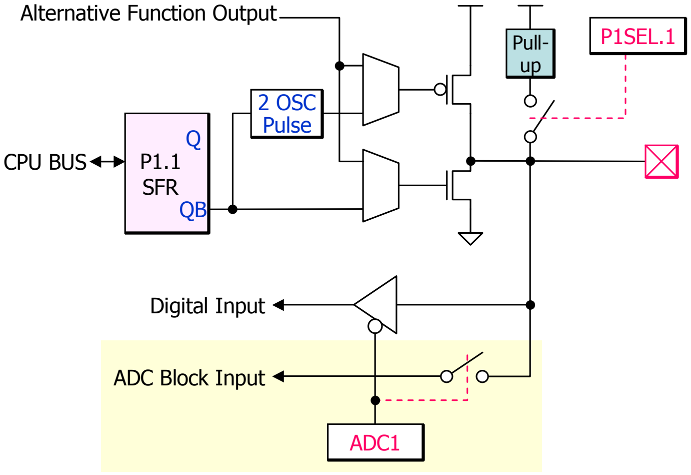


Figure 6-6 Configuration of PORT1

6.2.1.3 PORT 2

Port 2 is an 8-bit, bi-directional I/O port with internal pull-up resistors. The pull-up resistors are switched on/off by the value of the SFR P2SEL. The Port 2 of the devices including internal program memory is available as a general purpose I/O. Port 2 pins that have 1s written to them float and can be used as high-impedance inputs. If the internal pull-up resistors are switched on, Port 2 pins are pulled high and can be used as inputs.

Port 2 emits the high address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). It may not be recommended that even a device including program memory should use its Port 2 as a general purpose I/O port if the device also accesses external memory. It is because the state of the Port 2 latch will change during any access to

Downloaded from Elcodis.com electronic components distributor

external memory. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 special function register as the high address byte.

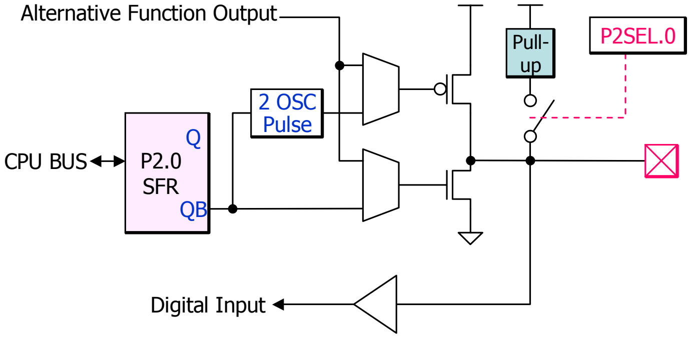


Figure 6-7 Configuration of PORT2

The ROMless devices of the MiDAS1.0 family use Ports 2 only for accesses to external memory. In this application, it uses strong driver all the time immediately after the rising edge of /PSEN.

6.2.1.4 PORT 3

Port 3 is an 8-bit bi-directional I/O port with internal pull-up resistors. The pull-up resistors are switched on/off by the value of the SFR P3SEL. Port 3 pins that have 1s written to them are pulled high weakly by the internal pull-up resistors and can be used as inputs. Port 3 pins can be also used alternate functions. To use the alternate functions, the corresponding bit of the Port 3 pin should be set to high. The alternate functions will be selected according to the combination of the SFR values.

The drive characteristics of an alternate function are the same with those of general I/O. The logic 0 output pulls a pin low strongly. The exceptions are pins P3.6 and P3.7, which employ the current-limited transition drivers described later when used as /RD and /WR signals. The logic 1 output changes a pin status with a strong pull-up and maintains the logic 1 status with a weak pull-up. More than one port pins are assigned to alternate functions without changing the functions of the other pins.

As an example, P3.6 is used as the alternate function of /WR and P3.7 as the alternate function of /RD. These strobes are used for accesses to external data memory. If only the /RD signal is used, P3.6 would still function as an I/O pin.

Downloaded from Elcodis.com electronic components distributor

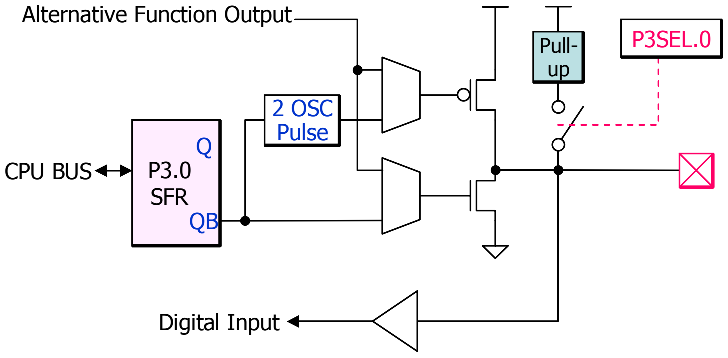


Figure 6-8 Configuration of PORT3

For another practical example, P3.2 is used as the /INT0 input pin. If external interrupt 0 is enabled, a logic 0 input to P3.2 will generate external interrupt 0. Then by disabling the external interrupt 0, P3.2 will return to a general purpose I/O pin. So the interrupt is used to "wake-up" the system from power saving modes.

6.2.1.5 PORT 4

Port 4 is an 4-bit bi-directional I/O port with internal pull-up resistors. The pull-up resistors are switched on/off by the value of the SFR P4SEL. Port 4 pins that have 1s written to them are pulled high weakly by the internal pull-up resistors and can be used as inputs. Its output pins are push-pull drived.

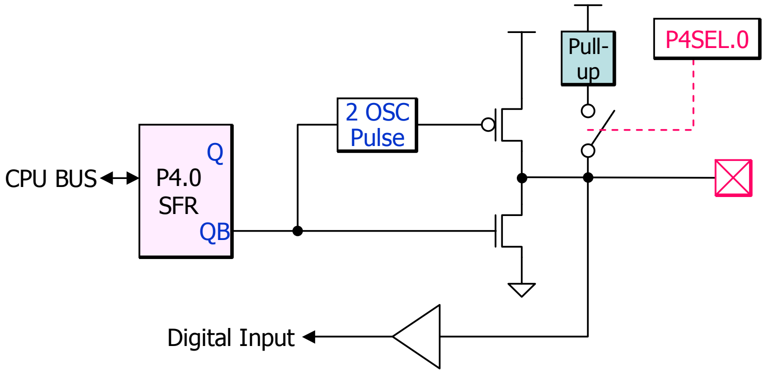


Figure 6-9 Configuration of PORT4

Downloaded from Elcodis.com electronic components distributor

6.2.1.6 ESD protection of I/O ports

As electrostatic discharge (ESD) problems become more common in electronic circuits, various devices have been used to protect circuits from ESD. All pins of MiDAS1.0 family excluding the /EA and VPP pins, use two diodes and one resistor for ESD protection. The ESD protection scheme is shown in Figure 6-10.

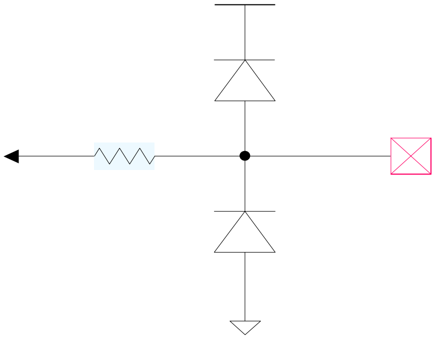


Figure 6-10 ESD protection scheme I

As voltage-clamping devices, the two diodes and a resistor limit the surge voltage to a safe level for the circuit being protected.

We protect the /EA and VPP pins from ESD with one diode and one resistor as shown in Figure 6-11

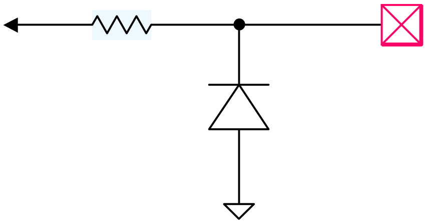


Figure 6-11 ESD protection scheme II

Similarly, the diode and the resistor limit the surge voltage to a safe level for the circuit being protected.

6.2.1.7 Output Functions

When software writes logic 0 to a port, strong pull-downs are used. However, in the case of logic 1, Port 1, 2, 3 or 4 use weak pull-up resistors and Port 0 will float (after the strong transition from 0 to 1). Thus as long as the ports are not pulled low by another output driver, the pin status will be high. The voltage of a

Downloaded from Elcodis.com electronic components distributor

port pin is determined by the DC current flowing out from it.

For status transition from logic 0 to logic 1, strong pull-ups are used. Since the pull-down drivers are strong, no driver for status transition from logic 1 to logic 0 is needed. The transition current is applied for changing from logic 0 to logic 1. This transition current is used for two clock cycles. The absolute current is approximately -650uA at 5V.

Now the operation an I/O port will be explained. A very strong pull-down FET is always activated if logic 0 is programmed to a port pin. A short circuit to Vcc must be avoided if the transistor is turned on, since the high current might destroy the FET. This also means that no logic 0 must be programmed into the latch of a pin that is used as input. A very strong pull-up FET is activated for two clock cycles if a 0-to-1 transition is programmed to a port pin, i.e. logic 1 is programmed to a port latch which contained logic 0. This provides a fast transition of the logic levels at the pin. A weak pull-up transistor is always activated when logic 1 is in the port latch, thus providing the logic high output level. This pull-up FET sources a much lower current than the pull-up FET for a 0-to-1 transition; therefore the pin may also be tied to ground, e.g. when used as input with logic low input level.

Substantial DC current can be supplied for both logic 1 and 0. However, it is not recommended to drive many heavy loads. It is because power dissipation may exceed the limitation.

6.2.1.8 Input Functions

To be used as an input, the port latch must contain logic 1, which turns off all the output drivers except the weak pull-up resistors. So the port pin is pulled high weakly by the internal pull-up resistors and can be pulled low by an external source. Thus when the port latch contains logic 1, the port pin will be configured as an input. When externally pulled high, it will maintain logic 1. When externally pulled low, it will source current and its logic state will drop to 0. So it can be used as an input. All instructions except the read-modify-write instructions read the port pin.

6.2.1.9 Read-Modify-Write Instructions

Some instructions that read a port read the latch and others read the pin. The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called “read-modify-write” instructions. The instructions listed below are read-modify-write instructions.

Downloaded from Elcodis.com electronic components distributor

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

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of transistor. When logic 1 is written to the bit, the transistor is turned on. If the CPU the reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as logic 0. Reading the latch rather than the pin will return the correct value of logic 1.

6.2.1.10 Alternate Functions

All the port pins are multifunctional. They also serve the alternate functions specified below. The alternate functions can only be activated if the corresponding bit latch in the port SFR contains logic 1. Otherwise the port pin is stuck at 0. The alternate functions are activated respectively. They will be activated or inactivated according to the application. More information about each alternate function will be provided later.

Downloaded from Elcodis.com electronic components distributor

Table 6-5 Alternate Functions

Port	Alternate Functions
P0.0	(AD0.0) Multiplexed Address/Data bus or (ADC0) ADC input channel 0
P0.1	(AD0.1) Multiplexed Address/Data bus
P0.2	(AD0.2) Multiplexed Address/Data bus
P0.3	(AD0.3) Multiplexed Address/Data bus
P0.4	(AD0.4) Multiplexed Address/Data bus
P0.5	(AD0.5) Multiplexed Address/Data bus
P0.6	(AD0.6) Multiplexed Address/Data bus
P0.7	(AD0.7) Multiplexed Address/Data bus
P1.0	(T2) Timer2 output pulse or (PWM1) PWM1 output
P1.1	(T2EX) Timer 2 capture/reload input or (ADC1) ADC input channel 1
P1.2	(ADC2) ADC input channel 2
P1.3	(ADC3) ADC input channel 3
P1.4	(INT2) External Interrupt 2 rising edge active
P1.5	(/INT3) External Interrupt 3 falling edge active
P1.6	(INT4) External Interrupt 4 rising edge active
P1.7	(/INT5) External Interrupt 5 falling edge active
P2.0	(A8) Address high bus
P2.1	(A9) Address high bus
P2.2	(A10) Address high bus
P2.3	(A11) Address high bus
P2.4	(A12) Address high bus
P2.5	(A13) Address high bus
P2.6	(A14) Address high bus
P2.7	(A15) Address high bus
P3.0	(RXD) Serial Receive UART
P3.1	(TXD) Serial Transmit UART
P3.2	(/INT0) External Interrupt 0 active low
P3.3	(/INT1) External Interrupt 1 active low
P3.4	(T0) Timer 0 input or (PWM0) PWM0 output
P3.5	(T1) Timer 1 input
P3.6	(/WR) External RAM Write strobe
P3.7	(/RD) External RAM Read strobe

Downloaded from Elcodis.com electronic components distributor

6.2.2 LVD (Low Voltage Detector)

6.2.2.1 Power–On Reset

When power is turned on, MiDAS1.0 Family starts an automatic reset by generating a power-on reset pulse internally. Once the supply voltage, V_{CC}, rises above the threshold (V_{RST} = 2.5V), the processor will restart clock generation with the external crystal. After 65,536 clock cycles, the program execution will start at location 0000h. As a result, the device starts reliable operation in the defined state established by the reset.

A software can find out using the Power–On Reset flag (POR in WDCON[6] or PCON[4]) that a Power–On Reset has occurred. Since the program execution always starts at location 0000h after all kinds of resets, software should clear the POR bit after reading it.

6.2.2.2 Power-Fail Reset

When power falls below V_{RST} (Power-Fail), the device will once again go into reset state. This reset condition will be maintained until power returns to the proper operating levels. When power rises above the reset threshold (V_{RST}), the device will perform the power–on reset and set the POR flag. Thus the drop of Vcc below V_{RST} produces the same result with power-up.

6.2.2.3 Power–Fail Interrupt

All MiDAS1.0 devices detect the fall of V_{CC} below V_{PFI}(4.0V +/- 10%) and set up the power-fail interrupt flag. As a result, the power-fail interrupt occurs if enabled To do so, they compare V_{CC} with their internal reference voltage, V_{PFI}. Among all interrupts, the Power–fail Interrupt has the highest priority. Nobody can change the priority level. But it can be disabled if not needed. The level of V_{PFI} is provided in the data sheet specifications associated with each device. The EPFI bit (WDCON[5]) enables the Power–fail Interrupt. This interrupt is independent of the global interrupt enable (EA). The Power–fail interrupt is a level–sensitive interrupt and its flag will be maintained 'high' as long as V_{CC} remains below V_{PFI}.

The related SFRs are :

- EXIF[0] (BGS) : If BGS = 0, LVD block stops when a device operates in power-down mode. Default value is 1.

Downloaded from Elcodis.com electronic components distributor

- WDCON[6] (POR) : After Power-on reset, POR will be set.
- WDCON[5] (EPFI) : Power-fail interrupt enable.
- WDCON[4] (PFI) : Power-fail interrupt flag. When VCC drops below V_{PFI} or goes to above V_{PFI} PFI will be set. If Vdd < VPFI, PFI is always set to "1".
- PCON[4] (POF): After Power-on reset, POF will be set.

The power-On/Fail reset and interrupt signals are generated when V_{CC} rises above or falls below the reference voltages as shown in Figure 6-12.

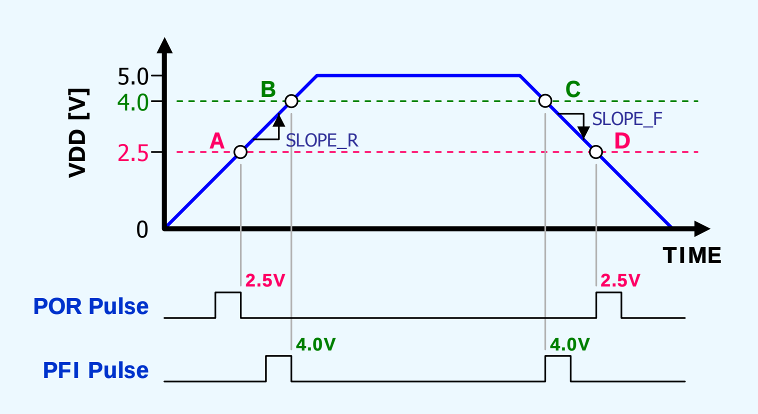


Figure 6-12 Power-On Reset/Power-fail Reset and Power-fail interrupt (V_{PFI}=4.0V and V_{RST}=2.5V)

The LVD block is described in Figure 6-13. It always remains activated except when BGS = 0 and a MiDAS1.0 processor is in power-down mode.

If your system needs to detect the power–fail in power-down mode(PCON[1]=1), you should set the BGS bit (EXIF[0]) to logic 1. Then, the band–gap reference and the power-fail detection circuits will remain activated in power-down mode. As a result, the power consumption of power-down mode increases. If the LVD block is deactivated, less than 1 μA of current will flow through the MiDAS1.0 processor. The current will increase to about 180 μA by activation of the LVD block.

Downloaded from Elcodis.com electronic components distributor

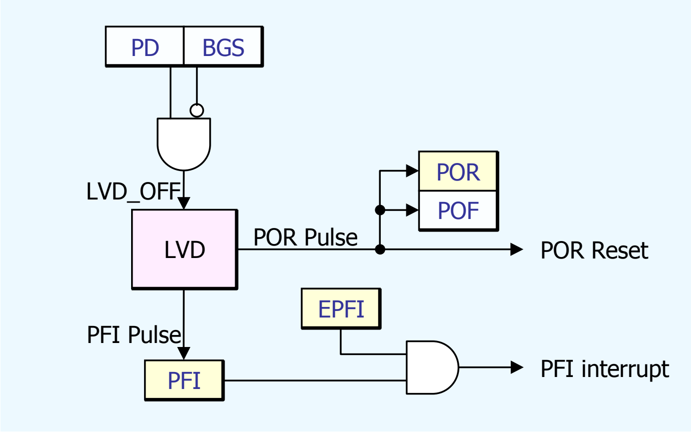


Figure 6-13 LVD Block Diagram

6.2.3 WDT (Watchdog Timer)

The Watchdog timer is a free-running timer with programmable time-out intervals. It is available for a system monitor, a time-base generator or an event timer. It allows an automatic recovery from software upset due to some cause including external noise. It is a kind of counter with the programmable bit length. It counts the system clock pulses. Software can always clear the value of the watchdog timer. When the count rolls over from all 1s to all 0s, the time-out occurs. At that time, the watchdog interrupt flag is set. The watchdog interrupt will occur if EWDT (watchdog interrupt enable) and EA (global enable) are set. If the watchdog reset is enabled by setting the bit EWT to logic 1, the watchdog reset will occur after 512 clock cycles since the watchdog timer interrupt flag is set, However, the watchdog timer can be cleared by setting the bit RWT to logic 1 during the delay of 512 clock cycles. Then the watchdog reset will not occur. The watchdog interrupt and reset may be used independently of each other. So a user can use them separately or together. Figure 6-14 shows the block diagram for Watchdog timer.

Downloaded from Elcodis.com electronic components distributor

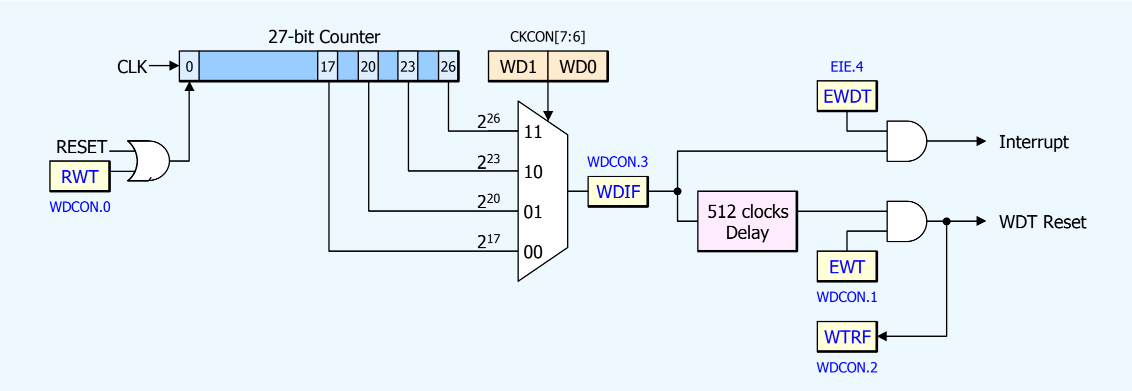


Figure 6-14 Block Diagram for Watchdog Timer

First the Watchdog timer should be cleared by setting the bit RWT (WDCON.0). Then the timer restarts from 0. After clearing the Watchdog timer, the hardware clears the RWT bit automatically. The time-out interval is selected by the values of 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 is enabled by setting the bit EWT to logic 1, the watchdog reset will occur after 512 clock cycles since the watchdog timer interrupt flag is set, However, the watchdog timer can be cleared by setting the bit RWT to logic 1 during the delay of 512 clock cycles. Then the watchdog reset will not occur. The reset process will last for thirty clock cycles, and the Watchdog timer reset flag WTRF (WDCON.2) will be set. This flag indicates to software that the Watchdog time-out was the cause of the reset.

If the reset and the interrupt are disabled, the watchdog timer can be used as a simple timer. Every time the selected time-out occurs, the WDIF flag is set. Software can check the WDIF flag to detect a time-out and set the bit RWT to restart the timer. The Watchdog timer can also be used as a timer with a very long time-out interval. The interrupt is enabled in this case. Every time the time-out occurs, the interrupt service routine will be called if the global interrupt enable bit EA is set.

The main purpose of the watchdog timer is detection of software upset. This is important for real-time control applications. In this case, some power glitches or electro-magnetic interference may cause software upset.

The watchdog time-out interval varies with the clock frequency. The watchdog timer reset will occur after 512 clocks from the end of the time-out interval.

Downloaded from Elcodis.com electronic components distributor

Table 6-6 Time-out values for the Watchdog timer

WD1	WD0	Interrupt time-out (@25MHz)		Reset time-out (@25MHz)	
0	0	2^{17} clocks	5.24ms	2^{17} + 512 clocks	5.26ms
0	1	2^{20} clocks	41.94ms	2^{20} + 512 clocks	41.96ms
1	0	2^{23} clocks	335.54ms	2^{23} + 512 clocks	335.56ms
1	1	2^{26} clocks	2684.35ms	2^{26} + 512 clocks	2684.38ms

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 interval is 2^{17} clocks, which is the shortest time-out period.

6.2.4 Timer 0/1/2

The MiDAS1.0 family has three 16-bit programmable timer/counters.

6.2.4.1 Timer/Counters 0 & 1

Each of these Timer/Counters has two 8-bit registers that form the 16-bit counting register. For Timer/Counter 0, they are the upper 8-bit register TH0 and the lower 8-bit register TL0. Similarly, Timer/Counter 1 has two 8-bit registers, TH1 and TL1. They can be configured to operate either as timers or event counters.

When operating as a timer, the counting registers counts clock cycles. The timer clock frequency can be 1/12 or 1/4 of the system clock frequency. In the “Counter” function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 pin for Timer 0 and T1 pin for Timer 1. The T0 and T1 external inputs are sampled during S3 state of every 4-clock system or every 12-clock system. If the sampled value is high in one machine cycle and low in the next, a valid high-to-low transition on the pin is recognized and the count register is incremented. Since it takes 8 clocks (4-clock system) or 24 clocks (12-clock system) to recognize a negative transition on the pin, the maximum counting rate is 1/8 or 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 “Counter” mode, the recognized negative transition on T0 or T1 pin causes the counting register value to increase in the next machine cycle after the negative edge was detected.

Downloaded from Elcodis.com electronic components distributor

The “Timer” or “Counter” mode is selected by the control bit “C/T” in the TMOD SFR; bit 2 of TMOD selects the mode for Timer/Counter 0 and bit 6 of TMOD the mode for Timer/Counter 1. In addition, each Timer/Counter has four operating modes. The modes are selected by bits M0 and M1 in the TMOD SFR.

6.2.4.1.1 Time-base Selection

The MiDAS1.0 family provides the two time-bases for the timer/counter. Its timers can operate at the same speed with those of the standard 80C52 family, counting at the rate of 1/12 of the clock speed. This can ensure that timer programs of the MiDAS1.0 family have the same timing with those of the standard 80C52 family. This is the default time-base of the GC80C520 timers. However, the timers of the MiDAS1.0 family can also operate in the turbo mode, where they increments at the rate of 1/4 clock speed. This mode is selected by the T0M and T1M bits in CKCON SFR. A reset clears these bits to 0, and the timers then operate in the standard 80C52 mode. The user should set these bits to 1 if the timers are to operate in turbo mode.

6.2.4.1.2 Mode 0

In Mode 0, the timer/counters operate as an 8-bit counter with a divide-by-32 prescaler. This 13-bit counter consists of 8 bits of THx and the lower 5 bits of TLx. The upper 3 bits of TLx are indeterminate and should be ignored.

The value in the TLx register increases at the negative edge of the clock. Every time the value of the fifth bit in TLx changes from 1 to 0, the count in the THx register increases. When the count in THx rolls over from FFh to 00h, the overflow flag TFx in TCON SFR is set. The counted input is enabled only if TRx is set and either GATE=0 or /INTx=1. When C/T is cleared to 0, 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 and T1 (P3.5) for Timer 1.

When the 13-bit counter rolls over from 1FFFh to 000H, the timer overflow flag TFx of the relevant timer is set. If enabled, a timer interrupt will occur. When used as a timer, the time-base is selected as either {clock cycles/12} or {clock cycles/4} by the bits TxM of the CKCON SFR.

6.2.4.1.3 Mode 1

Mode 1 is the same as Mode 0, except that the timer register is being run with all 16 bits counter. This means that all the bits of THx and TLx are used. Roll-over occurs when the timer changes from FFFFh to

Downloaded from Elcodis.com electronic components distributor

0000h. The timer overflow flag TFx of the relevant timer is set and an timer interrupt will occur if enabled. The selection of the time-base in the timer function is the same as Mode 0. The function of GATE bit is the same as Mode 0.

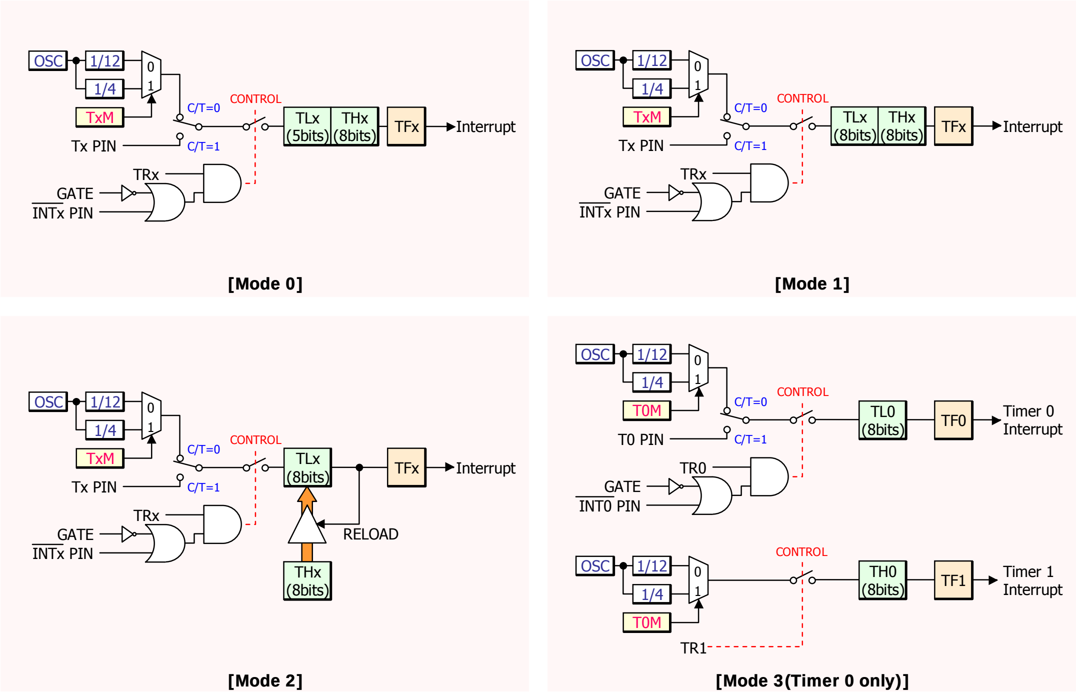
6.2.4.1.4 Mode 2

In Mode 2, the Timer/Counters operate as 8-bit counters with automatic reload. TLx operates as an 8-bit counter, while THx holds the reload value. When the TLx register overflows from FFH to 00H, the TFx bit in TCON is set and TLx is reloaded with the contents of THx, and the counting process continues from reloaded value. The reload operation leaves the contents of the THx register unchanged. Counting is enabled when the TRx = 1 and either GATE = 0 or /INTx =1. As in the other two modes 0 and 1, Mode 2 allows counting of either a timer clock (clock/12 or clock/4) or pulses on pin Tx.

6.2.4.1.5 Mode 3

In Mode 3, Timer/Counters 0 and 1 operate differently from each other. Timer/Counter 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer/Counter 0 in Mode 3, establishes TL0 and TH0 as two separate 8-bit counters. TL0 uses the Timer/Counter 0 control bits: C/T, GATE, TR0, /INT0 and TF0. It is determined by C/T (TMOD.2) whether the TL0 can count clock cycles (clock/12 or clock/4) or 1-to-0 transitions on pin T0. TH0 is locked int a timer function (clock/12 or clock/4) and takes over the use of TR1 and TF1 from Timer/Counter 1. Mode 3 is used when an extra 8-bit timer is needed. With Timer 0 in Mode 3, Timer 1 can still be used in Modes 0, 1 and 2. However, it no longer has control over its overflow flag TF1 and the enable bit TR1. Timer 1 can still be used as a timer/counter and retains the use of GATE and INT1 pin. In this condition, it can be turned on and off by switching it out of and into its own Mode 3. It can also be used as a baudrate generator for the UART.
Figure 6-15 illustrates the various operation modes for Timer/Counter 0/1.

Downloaded from Elcodis.com electronic components distributor



[Mode 0] [Mode 1] [Mode 2] [Mode 3(Timer 0 only)]

Figure 6-15 Timer/Counter 0/1 in Mode 0/1/2/3

6.2.4.2 Timer/Counter 2

Timer/Counter 2 is a 16-bit up/down counter which is configured by the T2MOD SFR and controlled by the T2CON SFR. Like Timers 0 and 1, it can operate either as a timer or as an event counter. Timer/Counter 2 counts either the external pulses of T2 pin (C/T2=1) or the prescaled clock(C/T2=0), which is divided by 12 or 4. If T2M = 0, 1/12 clock will be selected. Timer 2 counts when TR2 is 1, and stops when TR2 is 0.

6.2.4.2.1 Capture Mode

The capture mode is established by setting the CP/RL2 bit in the T2CON SFR to 1. In the capture mode, Timer/Counter 2 operates as a 16-bit up-counter. When the counter rolls over from FFFFh to 0000h, the TF2 bit is set, which will generate an interrupt request. If the EXEN2 bit is set, a negative transition of T2EX pin will cause the value in the TL2 and TH2 register to be captured by the RCAP2L and RCAP2H registers. This action also causes the EXF2 bit in T2CON to be set, which will also generate an interrupt.

Downloaded from Elcodis.com electronic components distributor

6.2.4.2.2 Auto-reload Mode, Counting Up

The auto-reload mode as an up-counter is enabled by clearing the CP/RL2 bit in the T2CON SFR and clearing the DCEN bit in T2MOD SFR. In this mode, Timer/Counter 2 is a 16-bit up-counter. When the counter rolls over from FFFFh to 0000h, TL2 and TH2 are reloaded with the 16-bit value in SFRS RCAP2L and RCAP2H. The TF2 bit is also set. If the EXEN2 bit is set, then a negative transition of T2EX pin will also cause a reload. The EXF2 bit in T2CON is also set.

6.2.4.2.3 Auto-reload Mode, Counting Up/Down

Timer/Counter 2 operates in auto-reload mode as an up/down-counter if CP/RL2 bit in T2CON is cleared and the DCEN bit in T2MOD is set. In this mode, Timer/Counter 2 is an up/down-counter whose direction is controlled by the T2EX pin. When T2EX is high, Timer 2 counts up. Timer overflow occurs at FFFFh which sets the TF2 flag. The overflow also causes the 16-bit value in RCAP2H and RCAP2L registers to be reloaded into the timer registers TH2 and TL2. When T2EX is low, Timer 2 counts down. Timer underflow occurs when the count in the timer registers TH2 and TL2 equals the value stored in RCAP2H and RCAP2L registers. The underflow sets TF2 flag and reloads FFFFh into the timer registers.
The EXF2 bit toggles when Timer 2 overflows or underflows according to the direction of the count. EXF2 does not generate any interrupt in this mode. This bit can be used to provide 17-bit resolution.

6.2.4.2.4 Baud Rate Generator Mode

The baud rate generator mode is selected by setting either the RCLK or TCLK bits in T2CON SFR. In this mode, Timer/Counter 2 is a 16-bit counter with automatic reload when the count rolls over from FFFFh to 0000h. However, rolling-over does not set the TF2 bit. If EXEN2 bit is set, a negative transition of the T2EX pin will set EXF2 bit in the T2CON and cause an interrupt request. Timer 2 interrupt caused by TF2 will be ignored in this mode.

6.2.4.2.5 Programmable Clock-out

In the clock-out mode, Timer 2 operates as a 50% duty cycle, programmable clock generator. The input clock increments TL2 at (Oscillator frequency) / 2. Timer 2 repeatedly counts to overflow from a loaded value. At overflow, the Timer 2 registers TL2 and TH2, will be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. In this mode, Timer 2 overflows do not generate

Downloaded from Elcodis.com electronic components distributor

interrupts as not in the baud rate generator mode. So Timer 2 can be used as a baud rate generator and a clock generator simultaneously. For this configuration, the baud rates and clock frequencies are not independent since both functions use the values in the RCAP2H and RCAP2L registers.

Timer 2 is programmed for the clock-out mode as follows:

- Set T2OE bit in T2MOD register.
- Clear C/T2 bit in T2CON register.
- Determine the 16-bit reload value from the formula and enter it in RCAP2H/RCAP2L registers.
- Enter a 16-bit initial value in timer registers TH2/TL2. It can be the same as the reload value or a different one depending on the application.
- To start the timer, set TR2 run control bit in T2CON register.

The formula gives the clock-out frequency as a function of the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

(Clock-out frequency) = (Oscillator frequency) / [4 * {65536 - (RCAP2H, RCAP2L)}]

The generated clock signal is brought out to T2 pin (P1.0)

Figure 6-16 and Figure 6-17 illustrate the operating modes supported by Timer/Counter 2.

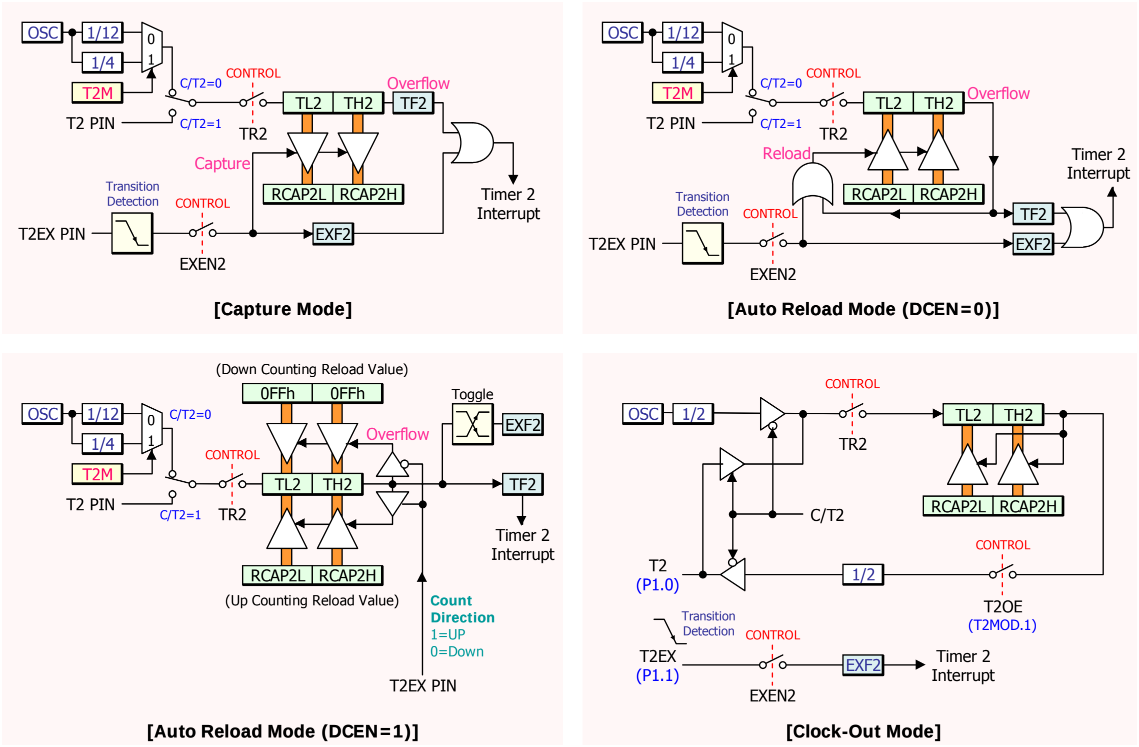


Figure 6-16 Timer/Counter 2 in Capture, Auto reload, and Clock-out Mode

Downloaded from Elcodis.com electronic components distributor

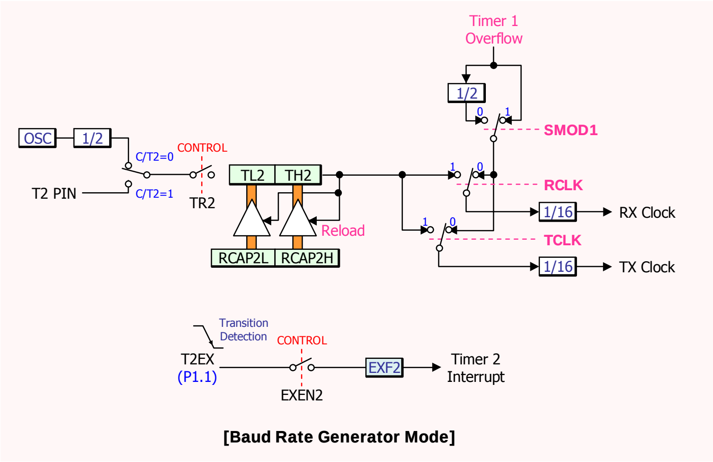


Figure 6-17 Timer/Counter 2 in Baud Rate Generator Mode

Downloaded from Elcodis.com electronic components distributor

6.2.5 UART (Universal Asynchronous Rx/Tx)

Table 6-7 Baudrate Example

Baudrate		UART Mode	FOSC	SMOD1	Timer 1		
T1M=0	T1M=1				C/T	Mode	Reload Value (TH1)
Max: 3 MHz	Max: 3 MHz	Mode 0	12 MHz	X	X	X	X
Max: 750 KHz	Max : 750 KHz	Mode 2	12 MHz	1	X	X	X
62.5 KHz	187.5 KHz	Mode 1 & 3	12 MHz	1	0	2	FFh
19.2 KHz	57.6 KHz		11.0592 MHz	1	0	2	FDh
9.6 KHz	28.8 KHz		11.0592 MHz	0	0	2	FDh
4.8 KHz	14.4 KHz		11.0592 MHz	0	0	2	FAh
2.4 KHz	7.2 KHz		11.0592 MHz	0	0	2	F4h
1.2 KHz	3.6 KHz		11.0592 MHz	0	0	2	E8h
137.5 Hz	412.5 Hz		11.0592 MHz	0	0	2	1Dh
110 Hz	330 Hz		6 MHz	0	0	2	72h
110 Hz	330 Hz		12 MHz	0	0	1	FEEBh

UART in the MiDAS1.0 family is a full duplex port. It includes additional features such as the Automatic Address Recognition. The UART port provides both synchronous and asynchronous communication modes. In the synchronous mode, the MiDAS1.0 family generates the communication clock and operates in a half duplex mode. In the asynchronous mode, the UART port operates as a full duplex port. This means that it can transmit and receive data simultaneously. The transmit buffer and the receive buffer are both addressed as SBUF SFR. However, writing to SBUF loads the transmit buffer, and reading SBUF accesses a physically separate receive buffer. The UART port can operate in four different modes as described below.

6.2.5.1 Mode 0

This mode provides synchronous communication with external devices. Serial data enters and exits

Downloaded from Elcodis.com electronic components distributor

through the RxD. TxD outputs the shift clock. This mode, therefore, supports a half duplex mode of serial communication. 8 bits are transmitted/received: 8 data bits (LSB first). Because the baud rate is fixed at 1/4 of the oscillator frequency, SM2 bit (SCON.5) should be 0 in mode 0.

The UART port sends and receives data through the RxD line. The TxD line outputs the shift clock. The clock shifts data bits into and out of the devices at the either end of the line. Transmission is initiated by any instruction that uses SBUF as a destination buffer. The shift clock will be activated and data will be shifted out to the RxD pin until all 8 bits are transmitted. The data on RxD will appear four clock periods at the center of the falling edge of TxD pin. This ensures that at the receiving end the data on RxD line can either be clocked on the rising edge of the shift clock on TxD.

The TI flag is set high in S1 state following the end of transmission of the last bit. The UART port will receive data when REN is 1 and RI is zero. The shift clock (TxD) will be activated and the serial port will latch data at the low state of shift clock. The external device should therefore present data at the rising edge on the shift clock. This process will be repeated until all the 8 bits have been received. The RI flag is set in S1 state following the last rising edge of the shift clock on TxD. This will stop reception and hold stop state until the RI is cleared by software.

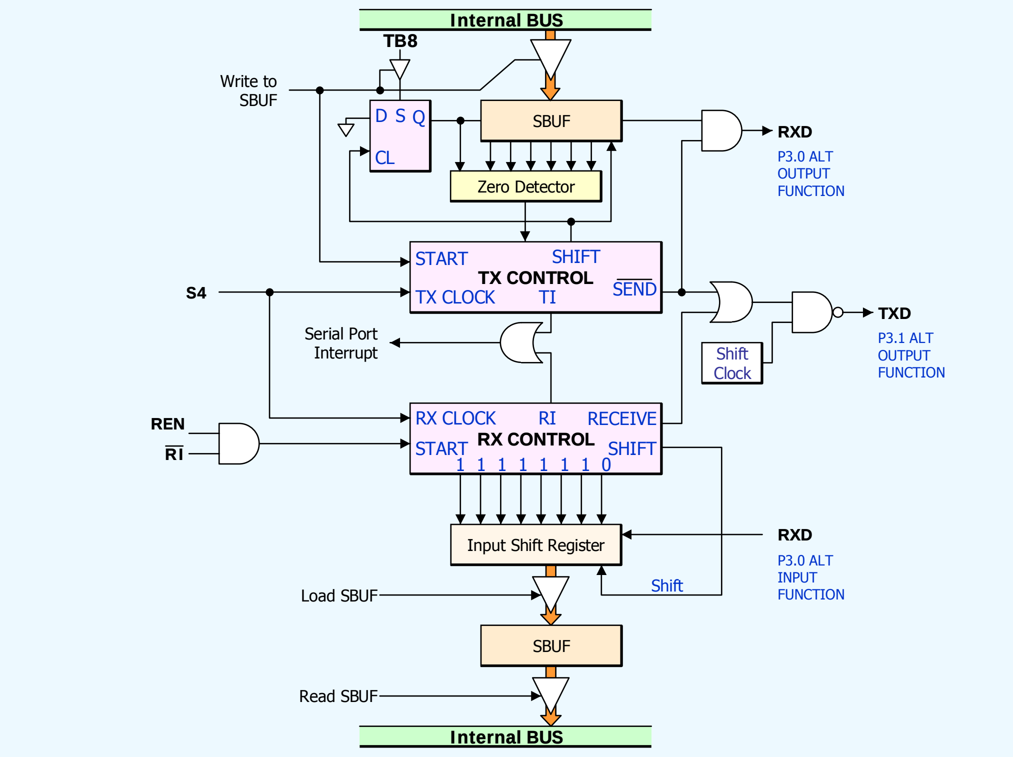


Figure 6-18 UART Mode 0

Downloaded from Elcodis.com electronic components distributor

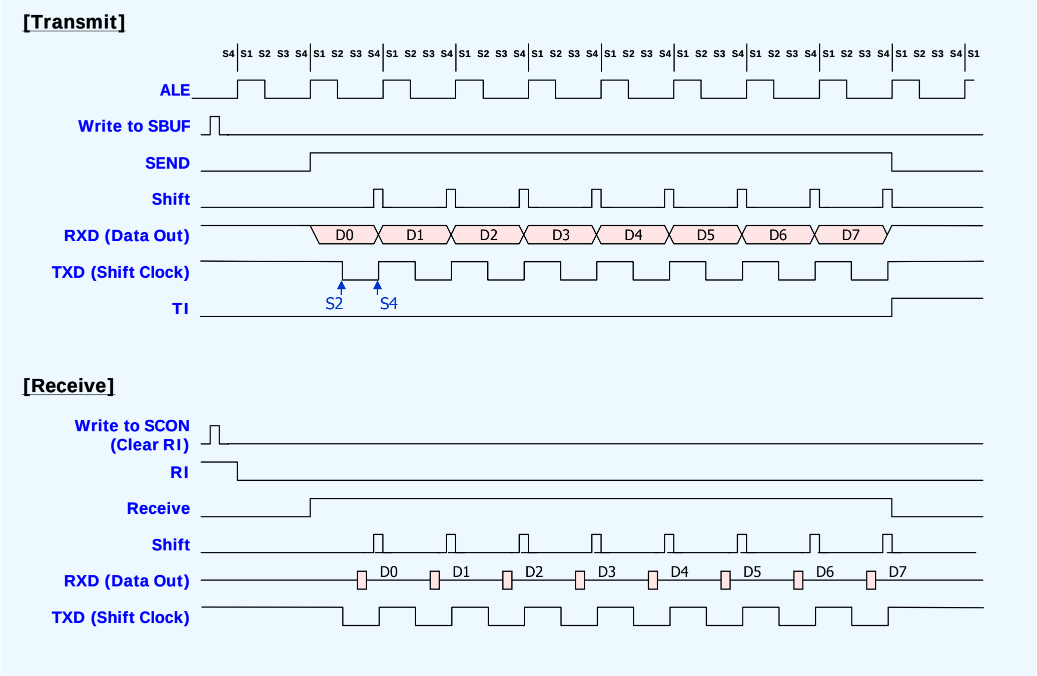


Figure 6-19 UART Mode 0 Timing

6.2.5.2 Mode 1

In Mode 1, the UART communicates asynchronously in the full duplex. Ten bits are transmitted (through TXD), or received (through RXD): 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 can be programmed to be 1/16 or 1/32 of the Timer 1 overflow or 1/16 of the Timer 2 overflow rate. It varies widely according to the automatic reload value of Timer 1 or Timer 2.

Transmission is initiated by any instruction that uses SBUF as a destination register. The serial data is sent to TxD pin at S1 state after the first roll-over of the divide-by-16 counter. The next bit is placed on TxD pin at S1 state after the next roll-over of the divide-by-16 counter. Thus, the transmission is synchronized to the divide-by-16 counter, not directly to the “write to SBUF” signal. After transmission of all 8-bit data, the stop bit is transmitted. The TI flag is set in the S1 state after the stop bit has been sent to TxD pin. This will occur at the 10th roll-over of the divide-by-16 counter after a write to SBUF.

Reception is enabled only if REN is “1”. It is initiated by detecting 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate established. When a transition is detected, the divide-by-16 counter is immediately reset. This aligns its roll-overs with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16 ths. At the 8th, 9th, and 10th counter states of each

Downloaded from Elcodis.com electronic components distributor

bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of three samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

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. Either SM2 = 0, or 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 in RxD.

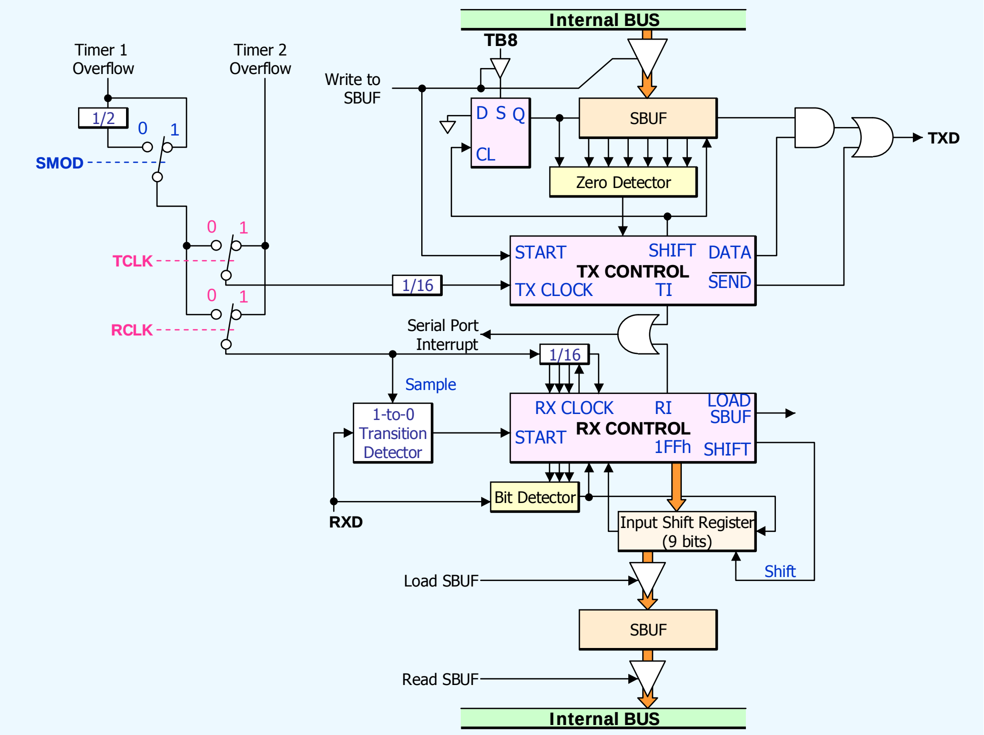


Figure 6-20 UART Mode 1

Downloaded from Elcodis.com electronic components distributor

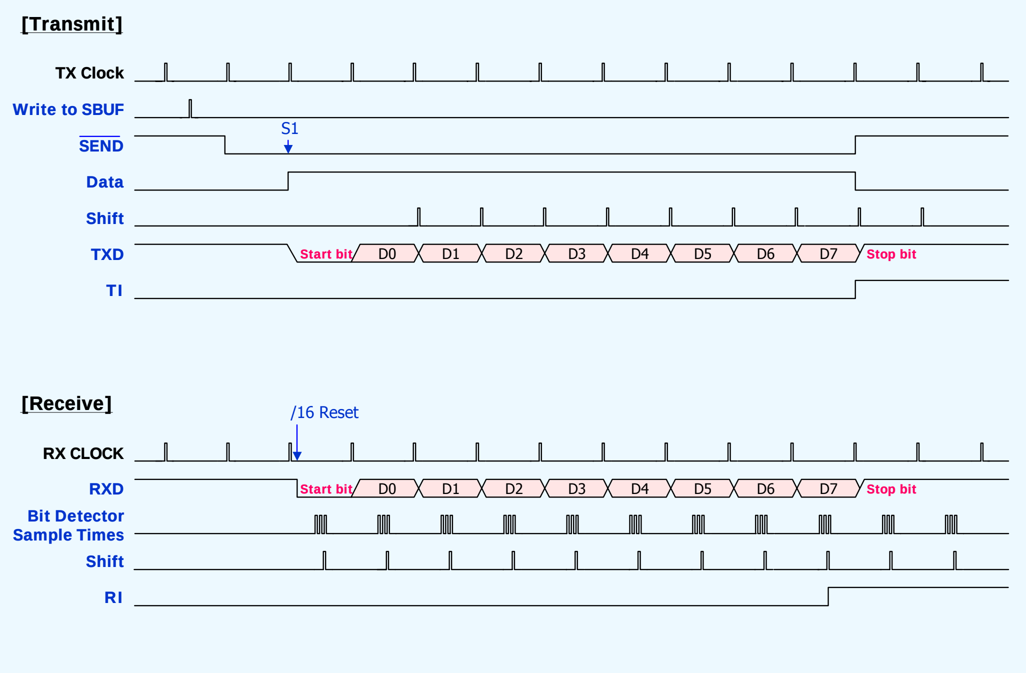


Figure 6-21 UART Mode 1 Timing

6.2.5.3 Mode 2

In Mode 2, the UART communicates asynchronously in full-duplex mode. Eleven bits are transmitted (through TXD), or received (through RXD): start bit (0), 8 data bits (LSB first), a programmable 9th bit (TB8) and a stop bit (0). On transmission, the 9th data bit (TB8) can be assigned the value of 0 or 1. On reception, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to 1/32 or 1/64 of the oscillator frequency, which is determined by the SMOD1 bit (PCON.7). Transmission is initiated by any instruction that uses SBUF as a destination register. The serial data is sent to TxD at S1 state after the first roll-over of the divide-by-16 counter. The next bit is placed on TxD at S1 state after the next roll-over of the divid-by-16 counter. Thus the transmission is synchronized to the divide-by0-16 counter, and not directly to the write to SBUF signal. After transmission of all 9 bits of data, the stop bit is transmitted. The TI flag is set at the S1 state after the stop bit has been put out on TxD. This will occur at the 11th roll-over of the divide-by-16 counter after a write to SBUF. Reception is enabled only if REN is “1”. Reception is initiated by detecting 1-to-0 transition of RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a 1-to-0 transition of RXD is detected, the divide-by-16 counter is immediately reset. This aligns its roll-overs with the boundaries of the incoming bit times. The 16 states of the counter divide each bit time into 16 ths.

Downloaded from Elcodis.com electronic components distributor

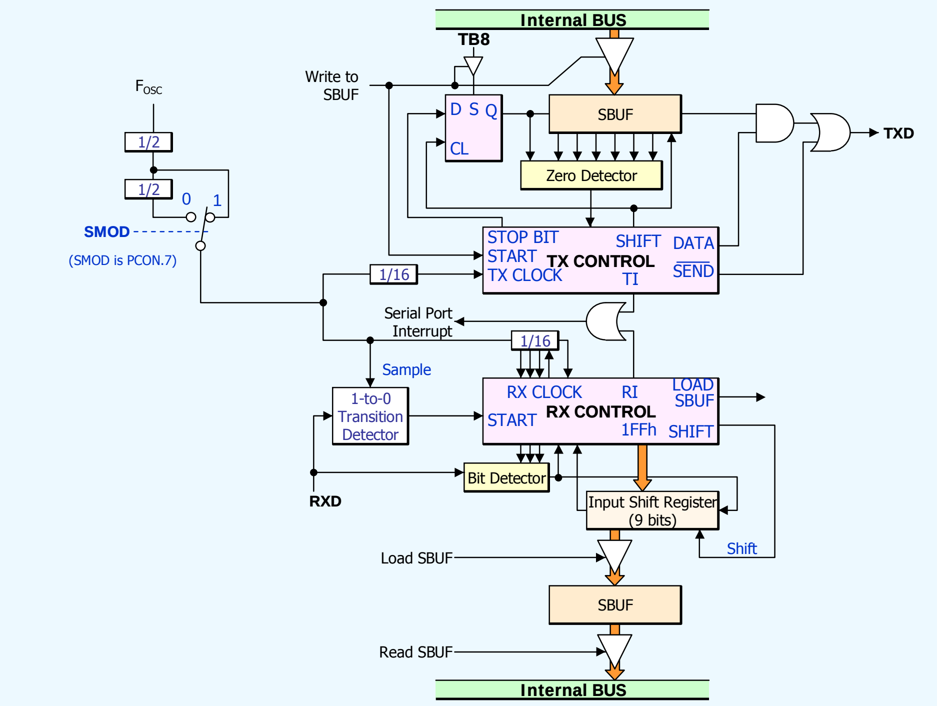


Figure 6-22 UART Mode 2

The 16 states of the counter divide each bit time into 16 ths. At the 8th, 9th, and 10th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of three samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

After shifting in 9 data bits, 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. Either SM2=0, or the received stop bit=1

If these conditions are met, then the stop bit goes to RB8, the 8 data bits 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.

Downloaded from Elcodis.com electronic components distributor

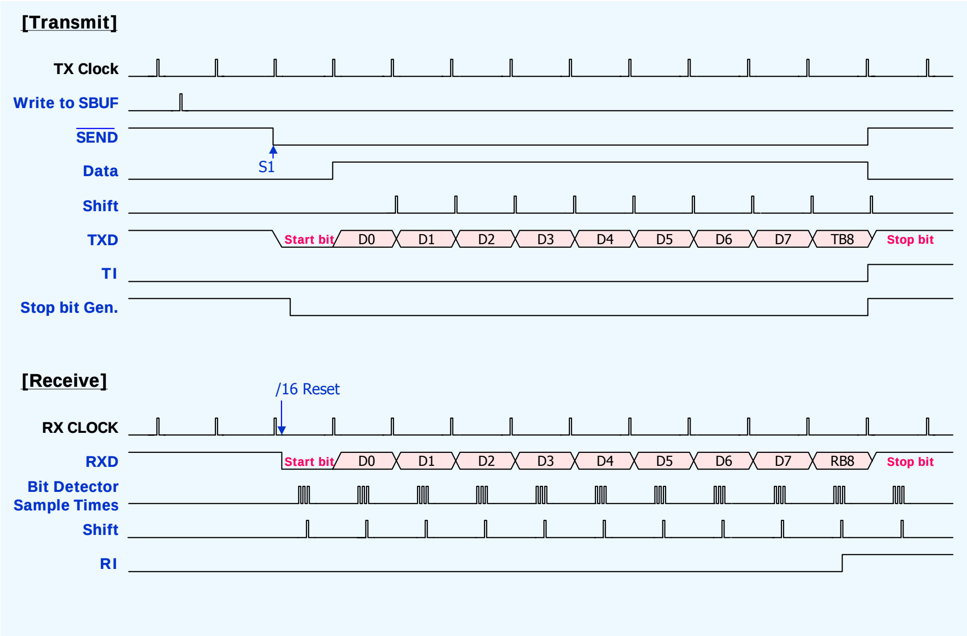


Figure 6-23 UART Mode 2 Timing

6.2.5.4 Mode 3

This mode is the same as Mode 2, except that the baud rate is programmable. Mode 3 may have a variable baud rate generated from either Timer 1 or 2 depending on the state of TCLK and RCLK. Timer 1 should also be initialized if Mode 1 and 3 are used. In all four modes, transmission is started by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 when RI=0 and REN=1. This will shift in 8 bits on the RxD pin. Reception is initiated in the other modes by the incoming start bit if REN=1. The external device will start the communication by transmitting the start bit.

Downloaded from Elcodis.com electronic components distributor

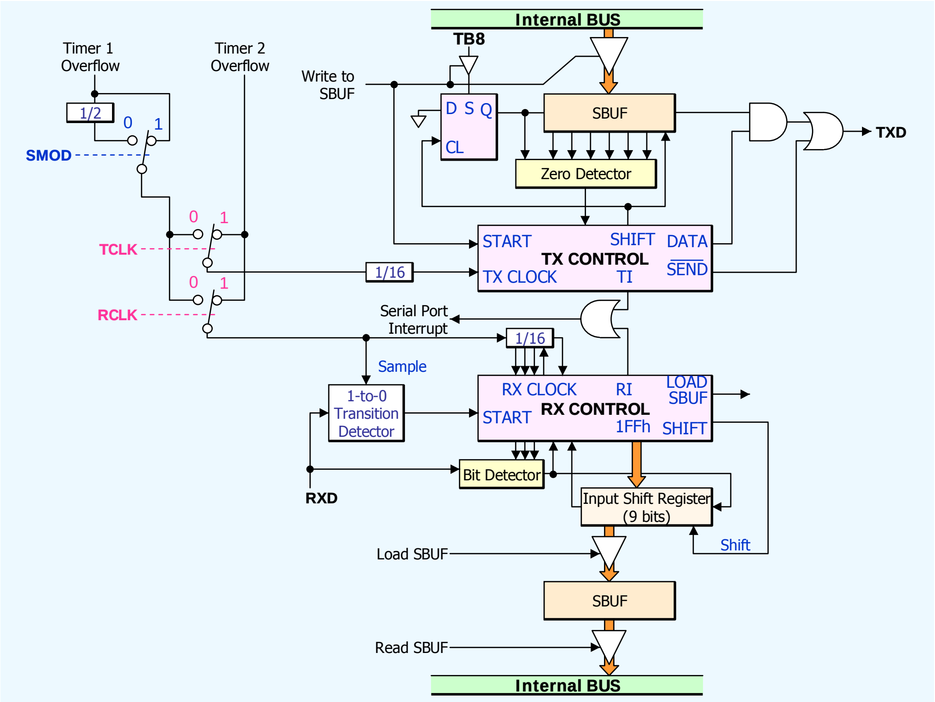


Figure 6-24 UART Mode 3

Table 6-8 Summary of Modes in UART

SM1	SM0	Mode	Data Size	Data Format	Baud Rate
0	0	0	8 bit	8 data bit	1/4 * oscillator clock
0	1	1	10 bit	Start bit, 8 data bit, stop bit	1/32 * Timer 1 overflow (SMOD1=0) 1/16 * Timer 1 overflow (SMOD1=1) 1/16 * Timer 2 overflow rate
1	0	2	11 bit	Start bit, 8 data bit, programmable bit, stop bit	1/32 * oscillator clock (SMOD1=0) 1/16 * oscillator clock (SMOD1=1)
1	1	3	11 bit	Start bit, 8 data bit, programmable bit, stop bit	1/32 * Timer 1 overflow (SMOD1=0) 1/16 * Timer 1 overflow (SMOD1=1) 1/16 * Timer 2 overflow rate

Downloaded from Elcodis.com electronic components distributor

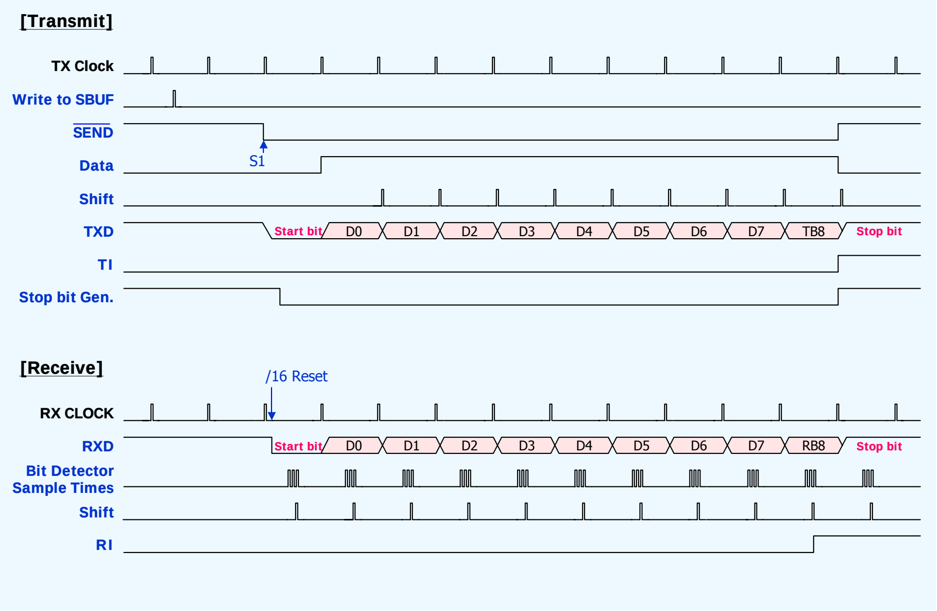


Figure 6-25 UART Mode 3 Timing

6.2.5.5 Automatic Address Recognition

Automatic address recognition makes use of the 9th data bit in Mode 2 and 3. In the MiDAS1.0 family, the RI flag is set only if the received byte corresponds to the given or broadcast address. This hardware feature eliminates the software overhead required for checking every received byte, and greatly reduces the software programmer task.

In the multiprocessor communication, the address byte is distinguished from the data byte by setting its 9th bit to 1. When the master processor wants to transmit data to a slave, first it sends the address byte of the targeted slave (or slaves). All the slave processors have their SM2 bit set high when waiting for its address byte. So they will be interrupted only by the reception of their address. By automatic address recognition feature, only the addressed slave will be interrupted. The address was recognized by hardware, not software.

The addressed slave clears the SM2 bit and waits for data bytes. If SM2=0, the slave will be interrupted every time it receive a complete frame of data. The unaddressed slaves will be still waiting for their address. If SM2 is 1, RI is set only if a valid frame is received and the received byte matches the given or broadcast address.

The master processor can selectively communicate with a group of slaves by using a given address. The address for each slave is formed with the SADDR and SADEN SFRs. The slave address is 8-bit value specified in the SADDR. The SADEN is actually a mask byte for the byte value in SADDR. If a bit in

Downloaded from Elcodis.com electronic components distributor

SADEN is 0, the corresponding bit in SADDR is don't care. By only the bits in SADDR whose corresponding bits in SADEN are 1, the given address is determined. So a user can address many slaves flexibly without changing the slave address in SADDR.

The following example shows how to address different slaves with a given address.

Slave 1:
SADDR 1010 0100
SADEN 1111 1010
Given 1010 0X0X

Slave 2:
SADDR 1010 0111
SADEN 1111 1001
Given 1010 0XX1

The given addresses for slave 1 and 2 have the different LSB. For slave 1: Don't care for Slave 1, 1 for Slave 2. Thus to communicate only with slave 1, the master must send 0 for the LSB of address value (1010 0000). Similarly the bit 1 is 0 for slave 1 and Don't care for slave 2. Hence to communicate only with slave 2, the master has to transmit 1 for the bit 1 of address value (1010 0011). If the master wishes to communicate with both slaves simultaneously, then the LSB two bits of address must be set to '01'. The bit 3 position is a don't care for both the slaves. So the two different addresses (1010 0001 and 1010 0101) can be selected.

The master can communicate with all the slaves simultaneously with the broadcast address. This address is formed from the logical OR of the SADDR and SADEN registers with zeros defined as don't care bits. In most applications, a broadcast address is FFh. In the previous example, the broadcast address is (1111 111X) for slave 1 and (1111 1111) for slave 2.

The SADDR and SADEN are located at address A9H and B9H, respectively. By reset, these two SFRs are initialized to 00H. The given address and broadcast address are set to XXXX XXXX (i.e. all bits are don't care). As a result, the master cannot communicate with the slaves selectively.

6.2.6 PWM (Pulse Width Modulator)

The MiDAS1.0 family has two channels of 8-bit pulse width modulated output channels. Their pulse width is programmable. The pulse width is adjusted by the value of SFRs; PWM0CON and PWM1CON.

Downloaded from Elcodis.com electronic components distributor

The PWM0 and PWM1 registers operate as 8-bit or 6-bit incrementing counters. The counter values increment until they reach to overflow. To operate the counter and the PWM block, set PWM0CON[0]/PWM1CON[0] to 1. If PWM0CON[0]/PWM1CON[0] is cleared to 0, the counter will stop and retain its value; when re-started, it will resume counting from the retained value. To clear the value of the counter to 0, set the bits PWM0CON[1]/PWM1CON[1] to 1. The PWM supports two modes: an 8-bit counter mode; a 6-bit counter and 2-bit extension mode. If PWM0CON[3] or PWM1OCN[3] is set to 1, a PWM block operates in the 8-bit counter mode. Otherwise, it operates in (2+6)-bit mode.

The prescale factor of the input clock frequency for a PWM counter, is determined by the value of PWM0CON[6:4] or PWM1CON[6:4]. The selectable input frequencies (F_{OSC} * prescale factor) 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

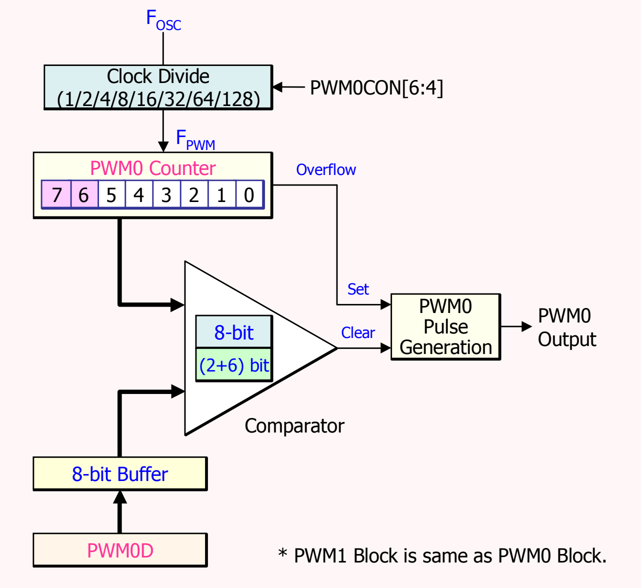


Figure 6-26 Functional block diagram

6.2.6.2 PWM components

The 8-bit PWM circuits consist of the following components:

- A 6-bit/8-bit comparator and a 2-bit extension effect implementation circuit
- A 6-bit/8-bit duty data register (PWM0D/PWM1D/PWM0D[5:0]/PWM1D[5:0])
- A 2-bit extension data register (PWM0D[7:6]/PWM1D[7:6])

Downloaded from Elcodis.com electronic components distributor

- A PWM output pin (P3.4/PWM0 or P1.0/PWM1)

6.2.6.3 PWM counter

A PWM counter operates in the two modes. One is the 8-bit counter mode. In this mode, the PWM counter operates as an 8-bit counter. A period of PWM output starts at the overflow of the counter. The duty rate of the PWM output is determined by comparing the value of the 8-bit counter with that of PWM0D or PWM1D. The mode is used for low frequency applications. The other is the 6-bit counter and 2-bit extension mode. In this mode, the period of the PWM output starts at the overflow of the lower 6-bits of the counter. The duty rate of the PWM output is determined by comparing the value of the lower 6-bits with that of PWM0D[5:0] or PWM1D[5:0]. In order to enhance resolutions, the pulse width of the PWM output can lengthen by one clock cycle according to the rule between the 2-bit extension and PWM0D[7:6]/PWM1D[7:6]. The mode is used for high frequency applications.

6.2.6.4 PWM frequency

The period of the PWM output are based on the prescaled PWM clock frequency. The PWM counter clock frequency is determined by the value of PWM0CON[6:4] or PWM1CON[6:4]. Table 6-9 shows the clock prescaling rate determined by the value of PWM0CON[6:4] or PWM1CON[6:4].

PWM0CON[6:4]/PWM1CON[6:4]	PWM clock prescaling rate
000	$F_{OSC}/1$
001	$F_{OSC}/2$
010	$F_{OSC}/4$
011	$F_{OSC}/8$
100	$F_{OSC}/16$
101	$F_{OSC}/32$
110	$F_{OSC}/64$
111	$F_{OSC}/128$

Table 6-9 PWM clock scaling rate determined by PWM0CON[6:4]/PWM1CON[6:4]

Downloaded from Elcodis.com electronic components distributor

6.2.6.5 PWM function Description

The PWM output signal drops to low level when the 8 bits of counter has the same value with the duty data register (PWM0D/PWM1D) in 8-bit counter mode, or the lower 6-bits of counter has the same value with the duty data register (PWM0D[5:0]/PWM1D[5:0]) in the 6-bit counter mode. In the 8-bit counter mode, if the values in the PWM0D/PWM1D register are not zero, an overflow of the counter causes the PWM output to rise from low level to high level. In 6-bit counter mode, if the values in the PWM0D[5:0]/PWM1D[5:0] register are not zero, an overflow of the lower 6-bits of the counter causes the PWM output to rise to high level. In this way, the value of the duty data register determines the duty rate of the PWM output.

To generate the PWM output of the (2+6)-bit mode, set PWM0CON.3 or PWM1CON.3 to 0. In the mode, pulse width of the PWM output can be lengthened according to the rule between the value of the upper 2-bits of the PWM counter and that of the extension data register. The upper 2-bits of the PWM counter counts the periods of the PWM output with the cycle of 4 from 1 to 4 (see Table 6-10). If, for example, the value in the extension data register is ‘01b', the 2nd period will be one input cycle longer than the other 3 periods. If the base duty rate is 50 %, the duty rate of the 2nd period will increase approximately to 51%. And if ‘10b’ is stored in the extension data register, all odd-numbered pulses will be one cycle longer. If ‘11h’ is stored to the extension data register, all pulses except the 4th period pulse will be extended by one cycle. The PWM signal will be sent to the corresponding PWM output pin through an output buffer. In this way, high output resolution can be obtained for high frequencies.

PWM0D[7:6]/PWM1D[7:6]	"Extended" Cycle Number
00	-
01	2
10	1, 3
11	1, 2, 3

Table 6-10 Output "extended" values for PWM data register (PWM0D[7:6]/PWM1D[7:6])

Downloaded from Elcodis.com electronic components distributor

6.2.6.6 PWM Timing Diagram

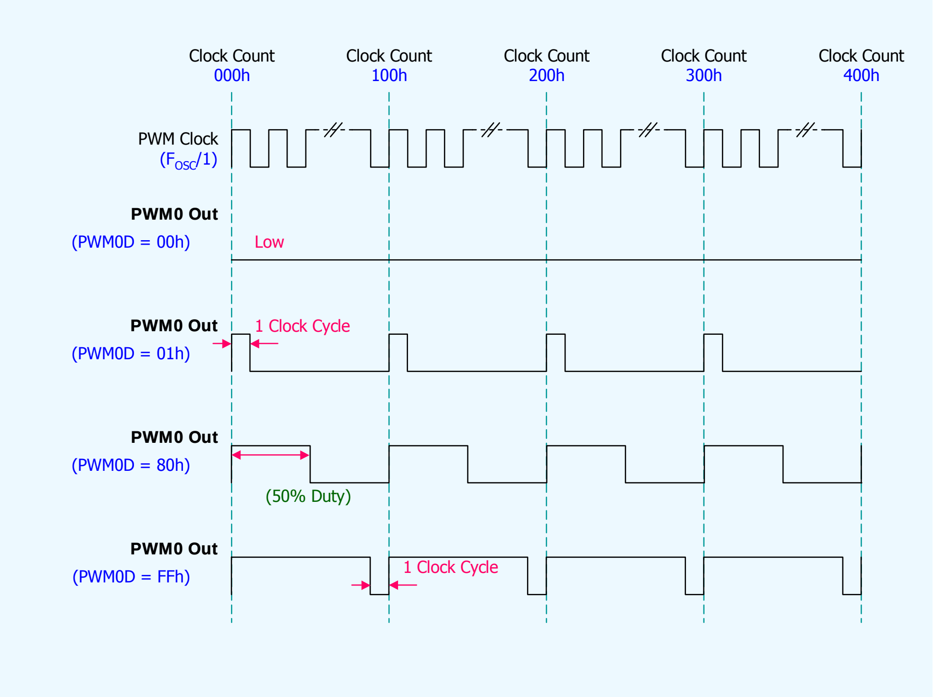


Figure 6-27 In 8bit counter mode, PWM basic timing diagram

Downloaded from Elcodis.com electronic components distributor

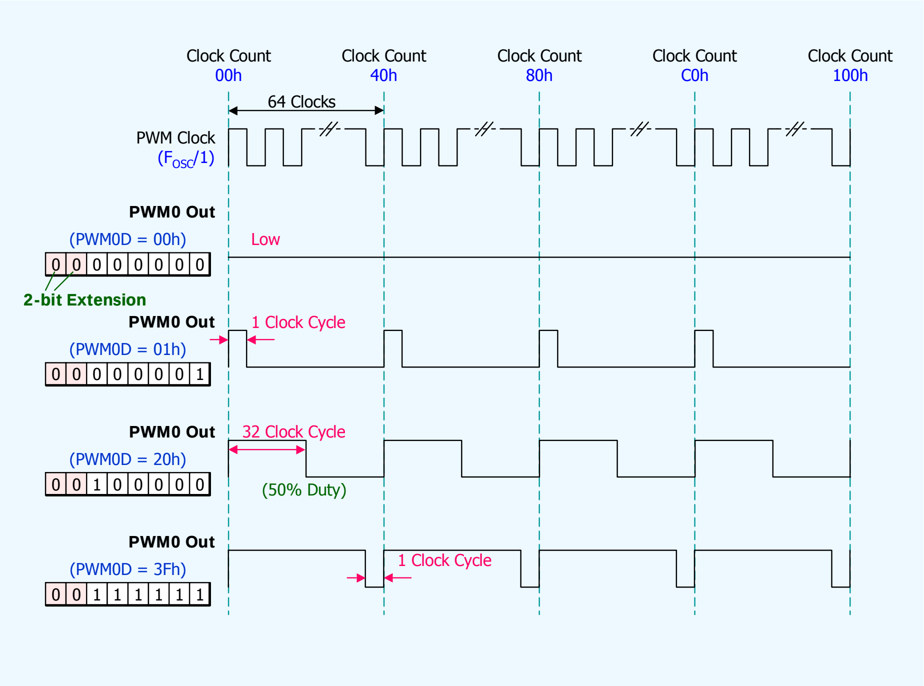


Figure 6-28 In 2+6bit count mode, PWM basic timing diagram

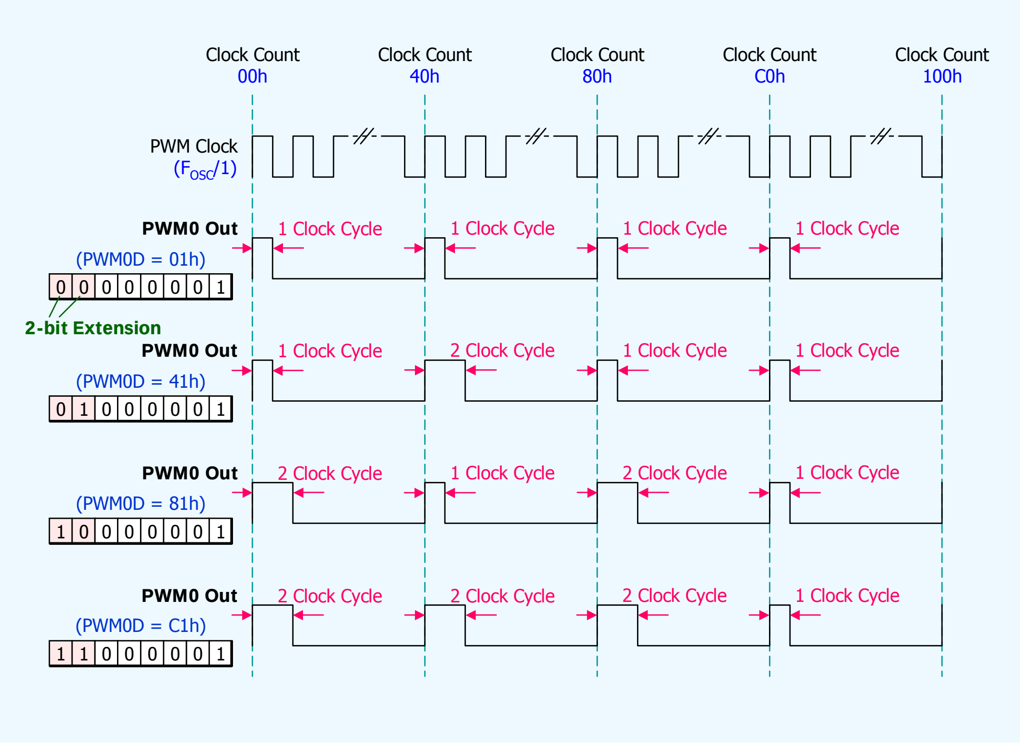


Figure 6-29 In 2+6bit counter mode, PWM extension cycle timing diagram

Downloaded from Elcodis.com electronic components distributor

6.2.7 ADC (Analog to Digital Converter)

The 9-bit A/D converter (ADC) converts analog input voltages to 9-bit digital values with successive approximation logic. The analog input voltages can come in through four input channels. The analog input voltage must be in the range between the VDD and VSS.

The A/D converter consists of the following components:

- An analog comparator with successive approximation logic
- A D/A converter logic
- An A/D converter control register (ADCON)
- Four multiplexed analog data input pins (ADC0–ADC3)
- 9-bit A/D conversion result data register (ADCR[7:0] and ADCON[0])

An analog-to-digital conversion is started as follows:

- Write the channel selection data to ADCON register in order to select one analog input channel among four analog pins (ADCn, n = 0 ~ 3).
- Set AD_EN bit and AD_REQ bit in ADCON register.

At first, A/D converter logic sets the successive approximation register to 000h. This register will be updated automatically for every conversion step. The successive approximation block performs 9-bit conversions for one input channel at a time. You can select different channels by manipulating the analog input selection value in ADCSEL[3:0] and the channel selection value (ADCON[3:2]) in ADCON register.

To start the A/D conversion, you should set AD_EN bit (ADCON[7]) and AD_REQ bit (ADCON[6]) to 1. 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]. ADCF (ADCON[4]) is used for interrupt flag. Conversion result is sent to the ADCR[7:0] and ADCON[0] register. Finally the A/D converter 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 voltage of the ADC0–ADC3 input pins should be kept in the absolute minimum range during a conversion procedure. Any change in the input voltage, perhaps due to circuit noise, will invalidate the result.

Downloaded from Elcodis.com electronic components distributor

The ADC input pins are alternately used for digital input in P0[0], P1[1], P1[2], and P1[3].

When used for ADC inputs, they must be configured as analog input. Also, you should switch off the pull-up resistors. To switch off the pull-up resistors, set the pull-up control bits: the P0SEL[0], P1SEL[1], P1SEL[2] and P1SEL[3].

6.2.7.1 INTERNAL REFERENCE VOLTAGE LEVELS

In the ADC block diagram (Figure 6-30), the analog input voltage is compared to the reference voltage. The analog input voltage must be within the range from VSS to VDD.

The resistor tree generated different reference internally during the analog-to-digital conversion process for each conversion step. The reference voltage level for the first bit conversion is always 1/2 VDD.

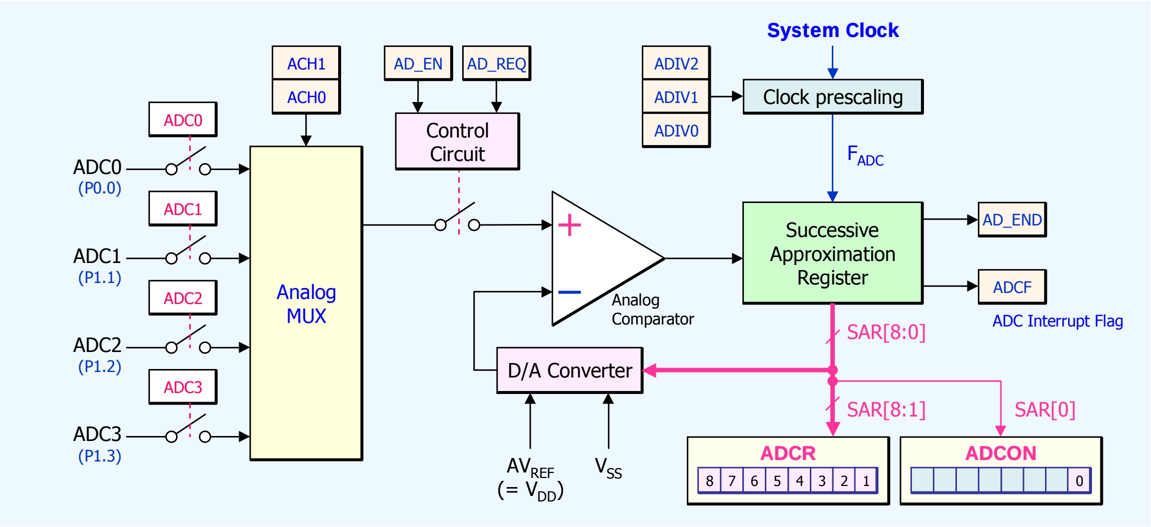


Figure 6-30 Analog-to-Digital Converter block Diagram

6.2.7.2 CONVERSION TIMING

The A/D conversion process requires total 88 clock cycles.

- Setup time: 8 clock cycles
- Conversion time: 9bit x 8 clock cycles = 72 clock cycles
- Hold time: 8 clock cycles

Downloaded from Elcodis.com electronic components distributor

Each conversion time for each system clock frequency is presented in Table 6-11.

Table 6-11 ***Example of conversion time vs. clock frequency***

OSC	Divide (ADCSEL[7:5])	F_{ADC}	T (1/F_{ADC})	1 sample conversion time
20 MHz (Vdd=5V)	000 (OSC/2)	10 MHz	100 ns	8.8 us
	001 (OSC/4)	5 MHz	200 ns	17.6 us
	010 (OSC/8)	2.5 MHz	400 ns	35.2 us
	011 (OSC/16)	1.25 MHz	800 ns	70.4 us
	100 (OSC/32)	625 KHz	1.6 us	140.8 us
10 MHz (Vdd=5V/3V)	000 (OSC/2)	5 MHz	200 ns	17.6 us
	001 (OSC/4)	2.5 MHz	400 ns	35.2 us
	010 (OSC/8)	1.25 MHz	800 ns	70.4 us
	011 (OSC/16)	625 KHz	1.6 us	140.8 us
	100 (OSC/32)	312.5 KHz	3.2 us	281.6 us

6.2.7.3 INTERNAL A/D CONVERSION PROCEDURE

The following sequences is the way to execute ADC conversion as described in Figure 6-31.

1. Analog input voltage must be in the voltage range between V_{SS} and V_{DD}.
2. Configure the analog input pins to analog input mode by setting the bits of ADCSEL[3:0] to 1.
3. Switch off a pull-up resistor by setting P0SEL[0] or P1SEL[3:1] to 1.
4. Before the conversion starts, you must select one of the four input pins (ADC0–ADC3) by setting ADCON[3:2] to the appropriate value.
5. To enable A/D conversion, set ADCON[7] to 1.
6. To start A/D conversion, set ADCON[6] to 1.
7. When the conversion is completed, the AD_END will be set to 0 and ADCF flag to "1".
8. Check AD_END or ADCF flag to verify if the conversion is finished or not.
9. The converted digital value is loaded to the ADC result register, ADCR[7:0] and ADCON[0]. Then the ADC enters an idle state.
10. The A/D conversion result can now be read from the ADCR and ADCON register.

Downloaded from Elcodis.com electronic components distributor

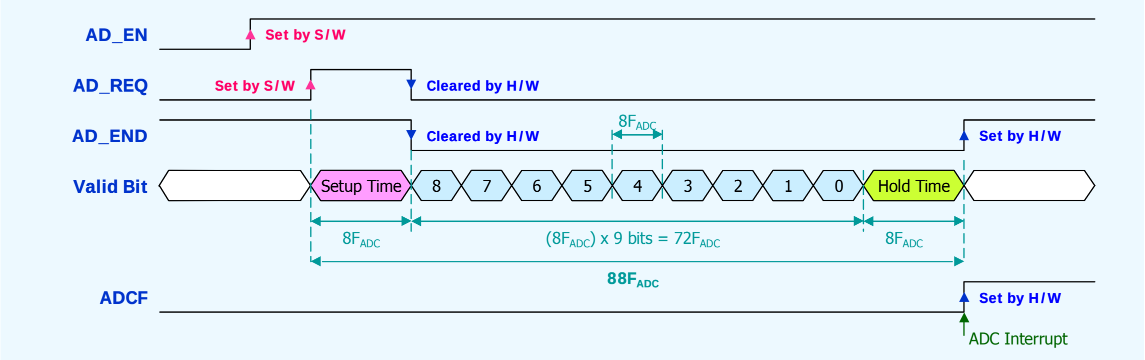


Figure 6-31 A/D Converter Timing Diagram

6.2.8 Interrupt

The MiDAS1.0 family has 13 interrupt sources with a four or two priority level interrupt structure. Each interrupt source has an individual priority bit, flag, interrupt vector and enable bit. In addition, All interrupts can be globally enabled or disabled.

6.2.8.1 Interrupt Sources

The two external interrupts /INT0 and /INT1 can be either edge or level triggered, depending on bits IT0 and IT1 in the TCON register. The bits IE0 and IE1 in the TCON register are the interrupt request bits. The other four external interrupts of INT2, /INT3, INT4 and /INT5 are only edge triggered. INT2 and INT4 are positive edge triggered but /INT3 and /INT5 are negative edge triggered. The negative interrupt inputs, /INT3 and /INT5, are sampled in every machine cycle. If the sampling result is high in one cycle and low in the next, a high to low transition is detected and the interrupt request flag IEx in TCON or EXIF is set. Similarly, a low-to-high transition of the positive interrupt inputs is detected and the interrupt request flag is set. Since the external interrupt inputs are sampled every machine cycle, they have to be held high or low for at least one complete machine cycle. The IEx in TCON is cleared automatically when the service routine is called. But, IEx in EXIF must be cleared by software. For a level triggered interrupt, the interrupt input has to be kept low till the interrupt is serviced. The request bits, IE0 and IE1, are not cleared by the hardware for the level triggered interrupt when the interrupt is serviced. If the interrupt input continues to be held low even after the service routine is completed, the processor may acknowledge another interrupt request from the same source. Note that the external interrupts, INT2 to /INT5, are edge triggered only. The individual interrupt flag corresponding to external interrupt 2 to 5 must be cleared manually by software.

Downloaded from Elcodis.com electronic components distributor

The TF0 and TF1 flags generate the Timer 0 and 1 interrupts. These flags are set by the overflow in the Timer 0 and 1. The TF0 and TF1 flags are cleared automatically by the hardware when the timer interrupt is serviced. The Timer 2 interrupt is generated by a logical OR of the TF2 and the EXF2 flags. These flags are set by overflow or capture/reload events in the Timer 2 operation. The hardware does not clear these flags when a Timer 2 interrupt is executed. Software has to resolve the cause of the interrupt between TF2 and EXF2 and clear the appropriate flag.

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

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 cleared automatically by the hardware, and a user will have to clear these bits using software.

ADCF flag and AD_END flag are set to 1 when A/D conversion is finished. Then the ADC interrupt will be generated if it is enabled. The ADCF flag must be cleared by software in ADC interrupt routine.

PFI flag is set when Vdd drops below V_{PFI}. The flag generates LVD interrupt if EPFI flag is set. This interrupt is not disabled by EA that is a global interrupt enable bit.

Each individual interrupt can be enabled or disabled by setting or clearing a bit in the IE. IE also has a global enable/disable bit EA that can be cleared to disable all interrupts.

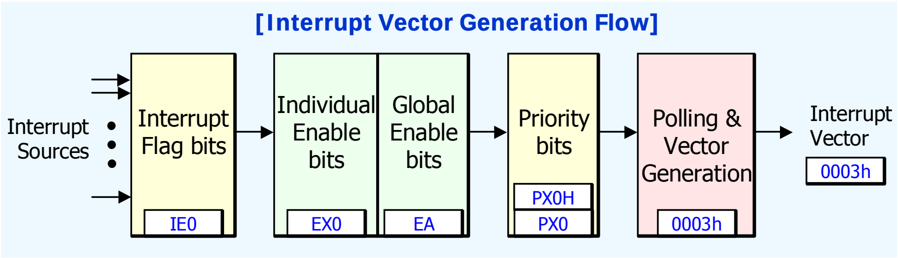


Figure 6-32 Interrupt Vector Generation Flow

6.2.8.2 Priority Level Structure

There are four or 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.

Downloaded from Elcodis.com electronic components distributor

Table 6-12 Priority Structure of Interrupts

Hierarchy	Sources	Vector Address	Priority Level
1 (highest)	LVD	0033h	
2	/INT0	0003h	4 levels
3	TF0	000Bh	4 levels
4	/INT1	0013h	4 levels
5	TF1	001Bh	4 levels
6	RI + TI	0023h	4 levels
7	TF2	002Bh	4 levels
8	ADC	003Bh	4 levels
9	INT2	0043h	2 levels
10	/INT3	004Bh	2 levels
11	INT4	0053h	2 levels
12	/INT5	005Bh	2 levels
13	WDT	0063h	2 levels

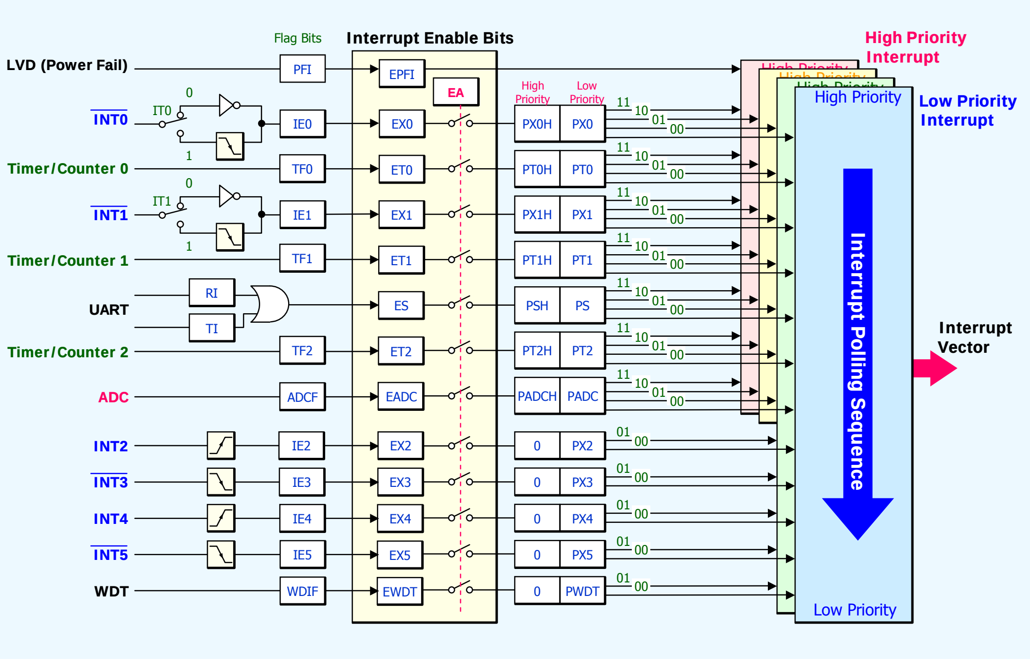


Figure 6-33 Hierarchy of Interrupt Priority

The interrupt flags are sampled at every machine cycle. In the same machine cycle, the sampled

Downloaded from Elcodis.com electronic components distributor

interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction that will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are

1. Neither an equal priority nor a higher priority interrupt is currently being serviced.
2. The current polling cycle is the last machine cycle of the instruction currently being executed.
3. The instruction in progress is neither RETI nor any write to IP, IE, EIP,EIE, IPH or EXIF.

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. 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 that 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 interrupts, /INT0 and /INT1, the flags are cleared only if they are edge triggered. In case of UART interrupts, the flags are not cleared by hardware. In the case of Timer 2 interrupt, 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 Program Counter is reloaded with the vector address of that interrupt which caused the LCALL.

Execution proceeds from the vector address until an RETI instruction is encounter. RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off. Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking that the service routine was still in progress.

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 in progress. The external interrupt inputs, /INT0 to /INT5, are sampled at S3 state of every machine cycle. After that, their interrupt flags IEx will be set. 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 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 takes four machine cycles to be completed. Thus there is a

Downloaded from Elcodis.com electronic components distributor

minimum of five complete machine cycles between activation of an external interrupt request and the beginning of execution of the service routine's first instruction.

A longer response time would result if any of the three conditions are not met. If a higher or equal priority level is already in progress, the additional wait time obviously depends on the nature of the currently executed service routine. If the polling cycle is not in the last machine cycle of the instruction in progress, then an additional delay is introduced. The maximum response time (if no other interrupt is in service) occurs if the MiDAS1.0 family is performing any 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 cycles is 44 clock cycles. Note that in the standard 80C52, the maximum latency is 8 machine cycles that equals 96 clock cycles. This is more than 50% reduction in terms of clock periods.

6.2.9 Reset Circuit

The user has several hardware related options for placing the MiDAS1.0 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 find 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 6-34.

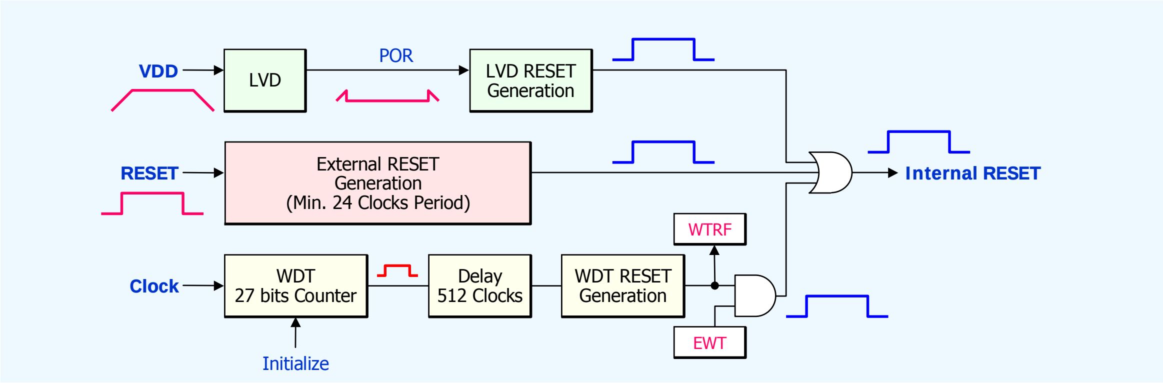


Figure 6-34 Three Reset Resources

Downloaded from Elcodis.com electronic components distributor

6.2.9.1 Power On/Fail Reset

The MiDAS1.0 family has a precision band-gap voltage reference to recognize that V_{CC} is out of tolerance. When power is turned on, its internal circuit detects the rise of V_{CC} above V_{RST}, the reset threshold (2.5V) voltage. Once V_{CC} is above this voltage, its oscillator starts oscillation. Then its internal reset circuit waits for 65,536 clock cycles until the power supply and the oscillator are stabilized. Next, the MiDAS1.0 processor will exit the reset state. No external components are needed to generate a power on reset. During power-down or during a severe power glitch, if V_{CC} falls below V_{RST}, the processor will also set Power On Reset flag to high. The reset state is maintained as long as the power voltage remains below the threshold. This will occur automatically, needing no action from the user or from the software.

6.2.9.2 External Reset

The reset input is the RST pin. The external input signal is asynchronous to the internal clock. The RST pin is sampled during state S4 of every machine cycle. The RST pin must be held for at least 6 machine cycles (24 clocks) to accomplish an external reset. The reset circuitry synchronously generates the internal reset signal. Thus the reset procedure operates synchronously, while the clock is running.

Once the device is in reset state, it will remain so as long as RST is 1. Even after RST 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 512 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.

Downloaded from Elcodis.com electronic components distributor

6.2.10 Clock Circuit (On-chip oscillators)

The MiDAS1.0 family supports two clock configurations as shown in Figure 6-35: 1) crystal oscillator, and 2) oscillator module.

The MiDAS1.0 family has the on-chip oscillator circuitry, which is a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator. Software can turn off its oscillator by writing a 1 to the PD bit in PCON as shown in Figure 6-36. Typically, C_L = 30 pF when the feedback element is a quartz crystal, and C_L = 47 pF when a ceramic resonator is used.

The load capacitance of the crystal oscillator is dependent on the operating frequency as shown in Figure 6-37. Table 6-13 shows the recommended load capacitance for the various frequencies.

Table 6-13 Recommended load capacitance in the crystal oscillator (V_{DD} = 5V)

	Crystal Oscillator [MHz]		
	~ 11.0592	22.1184	40.0000
Load Capacitance C_L	47pF	20pF	10pF

To drive the processor with an external clock source, apply the external clock signal to XTAL1 and leave XTAL2 float, as shown in Figure 6-36.

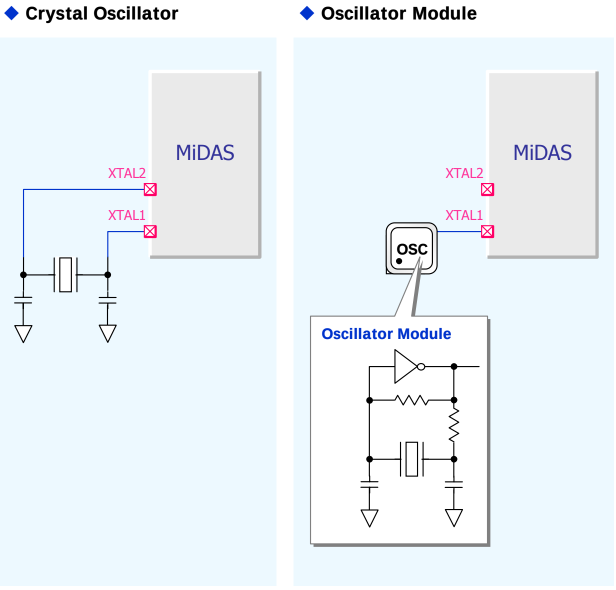


Figure 6-35 Configuration for Clock Generators

Downloaded from Elcodis.com electronic components distributor

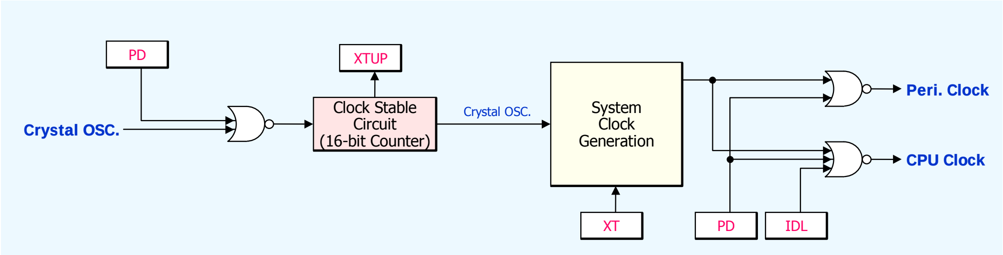


Figure 6-36 Clock Circuit

An external oscillator may encounter as much as a 100pF load at XTAL1 pin 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.

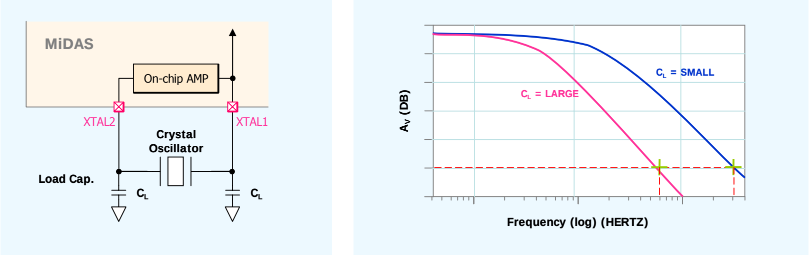
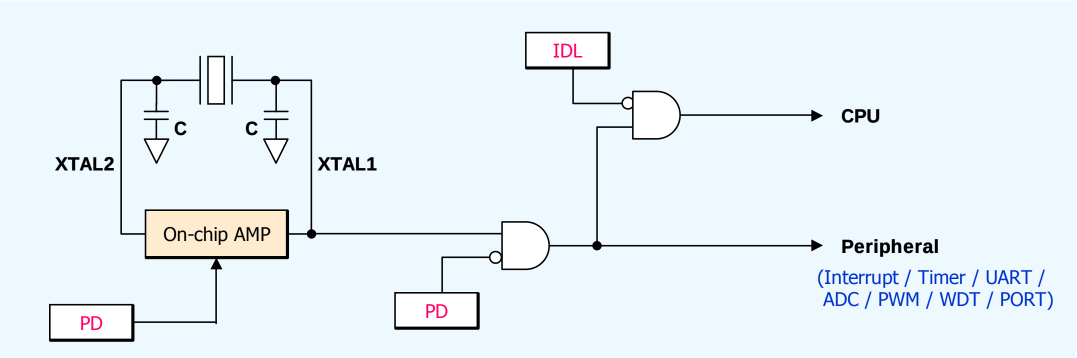


Figure 6-37 The Load Capacitor vs. Operating Frequency

6.2.11 Power Management

The MiDAS1.0 family has two power saving features (POWER DOWN mode and IDLE mode) that help the user to modify the power consumption of the device.



Downloaded from Elcodis.com electronic components distributor

Figure 6-38 Power Management Circuit

6.2.11.1 IDLE Mode

The user can put the processor 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 Stack 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 the IDLE state had started. 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 that put the processor into IDLE mode.

The IDLE mode can also be exited by activating the reset. The processor can be put into the reset by applying a high on the external RST pin, a Power-On-Rest condition or a Watchdog timer reset. The external reset pin has to be held high 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 has been already running, without delay 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 that will wake up the processor. The software must reset the Watchdog timer in order to preempt the reset that will occur after 512 clock periods of the time-out. When the MiDAS1.0 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 processor 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 processor goes into Power Down mode. In the Power Down mode, all the clocks are stopped and the processor 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 MiDAS1.0 family will exit the Power Down mode with a reset or by the two external interrupts, /INT0 and /INT1, enabled as level sensitive configuration (0 or 1). An external reset can be used to exit the

Downloaded from Elcodis.com electronic components distributor

Power Down state. The high on RST pin terminates the Power Down mode, and restarts the clock. The program execution will restart from 0000H. Because the clock stops in the Power Down mode, the Watchdog timer cannot be used for exiting the Power Down mode.

The MiDAS1.0 family can be woken from the Power Down mode by activating an enabled external, when the global enable (EA) bit is set to 1 and the external interrupt was set to the level sensitive mode. If these conditions are satisfied, the low level on the external interrupt pin restarts the oscillator. The external interrupt pin should be held to “0” for at least 65,536clocks. Because of crystal stabilization time, 65,536 clocks must be needed to wake-up from the power down mode. After 65,536 clocks, system clock will be running normally and interrupt logic will start the external interrupt. Then the processor will execute the interrupt service routine for the corresponding external interrupt. After the interrupt service routine is completed, the processor will go beyond the instruction that put it into the Power Down mode and it will start execution from the next instruction.

Table 6-14 Status of external pins during IDLE and Power Down modes

Mode	Program Memory	ALE	/PSEN	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

6.2.12 EPROM

The programming and verifying timing is very similar to that of Intel 80C52. The MiDAS1.0 family programs at V_{PP}=11.5V±0.5V using five /PROG(ALE) pulses per byte programmed (typ. 100us).

To erase the EPROM, the MiDAS1.0 family must be exposed for 20 minutes in UV-light (15mW/cm^2).

A 44-PLCC or 44-MQPF device may require an additional adaptor to program the EPROM.

In some applications, it is desirable that the Program Memory be secure from software piracy. The MiDAS1.0 family has varying degrees of program protection using Encryption array or lock bits.

6.2.12.1 Encryption Array

Within the EPROM is an array of encryption bytes that are initially unprogrammed (all 1’s). For EPROM

Downloaded from Elcodis.com electronic components distributor

devices, a user can program the encryption array to encrypt the program code bytes during EPROM verification. For ROM devices, a user submits the encryption array to be programmed by the factory. If an encryption array is submitted, Lock Bit 1 (LB1) will also be programmed by the factory. The encryption array is not available without the Lock Bit. Program code verification is performed as usual, except that each code byte comes out exclusive-NOR'ed (XNOR) with one of the key bytes. Therefore, to read the ROM/EPROM code, the user has to know the encryption key bytes in their proper sequence.

Unprogrammed bytes have the value FFh. If the Encryption Array is left unprogrammed, all the key bytes have the value FFh. Since any code byte XNOR'ed with FFh leaves the byte unchanged, leaving the Encryption Array unprogrammed in effect bypasses the encryption feature.

When using the encryption array feature, one important factor should be considered. If a code byte has the value FFh, verifying the byte will product the encryption byte value. It is recommended that all unused code bytes should be programmed with some value other than FFh, and not all of then the same value. This will ensure maximum program protection.

6.2.12.2 Program Lock Bits

Also included in the Program Lock scheme are Lock Bits which can be enabled to provide varying degrees of protection. Table 6-15 lists the Lock Bits and their corresponding influence on the device. A user is responsible for programming the Lock Bits on EPROM devices. On ROM devices, Lock Bit 1 is automatically set by the factory when the encryption array is submitted. The Lock Bit is not available without the encryption array on ROM devices.

Erasing the EPROM also erases the Encryption Array and the Lock Bits, returning the part to full functionality.

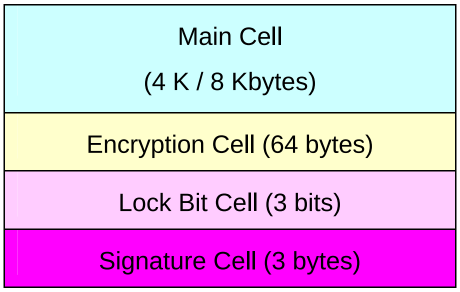


Figure 6-39 The EPROM Cell Configuration In Physical Layout

Downloaded from Elcodis.com electronic components distributor

Table 6-15 Conditions to control EPROM in ROM Writer

ROM Writer Interface											Functional Description
Mode		RESET	PSEN	ALE	VPP	P2.6	P2.7	P3.3	P3.6	P3.7	
Program code data		H	L	⊔	11.5V	L	H	H	H	H	V_{PP}=11.5V and ALE low pulse
Verify code data		H	L	H	5V	L	L	L	H	H	Code=(main byte) XNOR (Enc. Byte)
Program Encryption Array		H	L	⊔	11.5V	L	H	H	L	H	V_{PP}=11.5V and ALE low pulse
Program Lock Bits	Bit1	H	L	⊔	11.5V	H	H	H	H	H	Lock1=1: MOVC instruction disable. Further programming disable.
	Bit2	H	L	⊔	11.5V	H	H	H	L	L	Lock2=1: same as Lock1. And verify disable.
	Bit3	H	L	⊔	11.5V	H	L	H	H	L	Lock3=1: same as bit1, bit2. External code execution disable.
Read Signature		H	L	H	5V	L	L	L	L	L	30h=C0h (CORERIVER) 31h=87h(Product) 60h=52h(EPROM size)

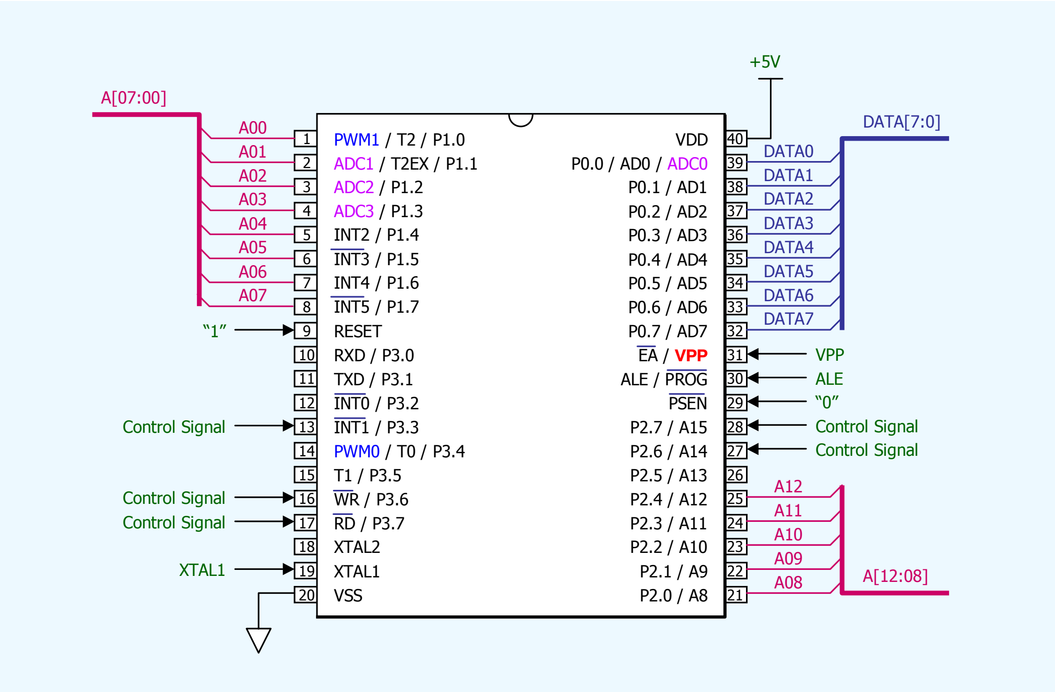


Figure 6-40 Pin Configuration for EPROM programming

Downloaded from Elcodis.com electronic components distributor

Table 6-16 EPROM Characteristics

Parameter	Symbol	Min	Max	Units
Programming supply voltage	Vpp	11.0	12.0	V
Programming supply current	Ipp		75	MA
Address setup to ALE low	T_{AVGL}	$48T_{CLCL}$		
Address hold after ALE high	T_{GHAX}	$48T_{CLCL}$		
Data setup to ALE low	T_{DVGL}	$48T_{CLCL}$		
Data hold after ALE high	T_{GHDX}	$48T_{CLCL}$		
ENABLE High to Vpp	T_{EHSH}	$48T_{CLCL}$		
Vpp setup to ALE low	T_{SHGL}	10		Us
Vpp hold after ALE high	T_{GHSL}	10		Us
ALE low width	T_{GLGH}	90	110	Us
Address to data valid	T_{AVQV}		$48T_{CLCL}$	
ENABLE low to data valid	T_{ELQV}		$48T_{CLCL}$	
Data float after EA	T_{EHQZ}	0	$48T_{CLCL}$	
ALE high to ALE low	T_{GHGL}	10		Us

* Oscillator frequency: Freq. = 4 ~ 6MHz (1 T_{CLCL} = 1/Freq.)

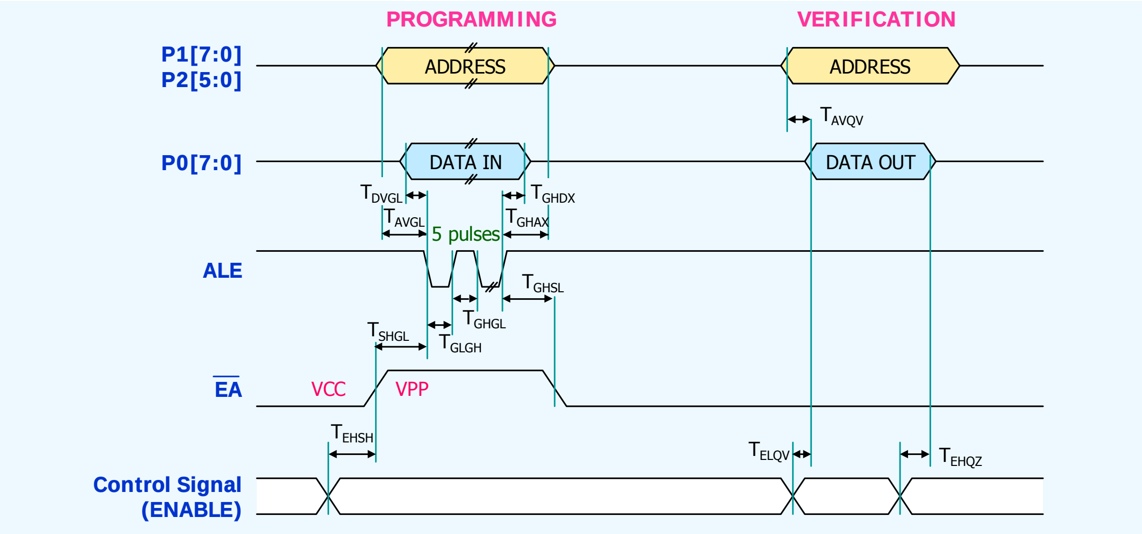


Figure 6-41 Timing of EPROM programming and Verification

Downloaded from Elcodis.com electronic components distributor

7 Strong Points

7.1 Reduction of noise influence

The MiDAS1.0 family can reduce noise by inhibiting the ALE signal. To do so, set the ALEOFF bit(PMR.2) to 1.

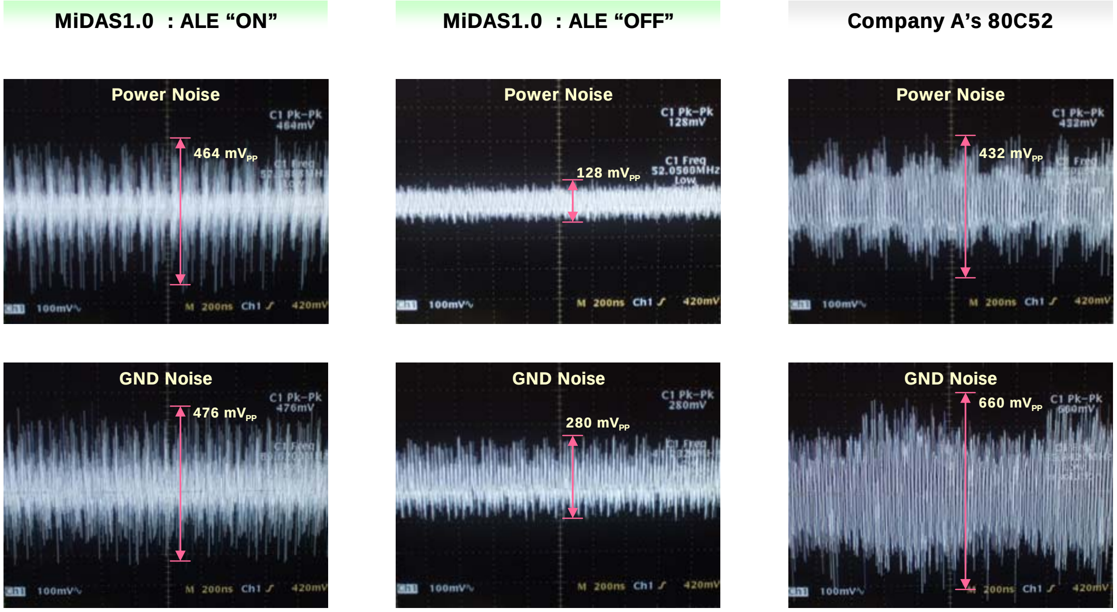


Figure 7-1 Noise reduction

Table 7-1 The comparison table of noise reduction

System Clock [MHz]	Noise	**MiDAS1.0**		**Company A's 80C52**
		ALE ON	ALE OFF	ALE ON (always)
11.0592	Power	410 mV_{PP}	**170 mV_{PP}**	360 mV_{PP}
	Ground	550 mV_{PP}	**330 mV_{PP}**	500 mV_{PP}
22.1184	Power	464 mV_{PP}	**128 mV_{PP}**	432 mV_{PP}
	Ground	476 mV_{PP}	**280 mV_{PP}**	640 mV_{PP}
6	Power	360 mV_{PP}	**170 mV_{PP}**	380 mV_{PP}
	Ground	500 mV_{PP}	**330 mV_{PP}**	480 mV_{PP}

Downloaded from Elcodis.com electronic components distributor

If the ALE signal is enabled, the noise amplitude of the MiDAS1.0 family is similar with that of Company A's 80C52. However, Table 7-1 shows that the MiDAS1,0 family reduce noise largely by inhibiting the ALE pin.

7.2 On-chip POR(Power-ON-Reset)

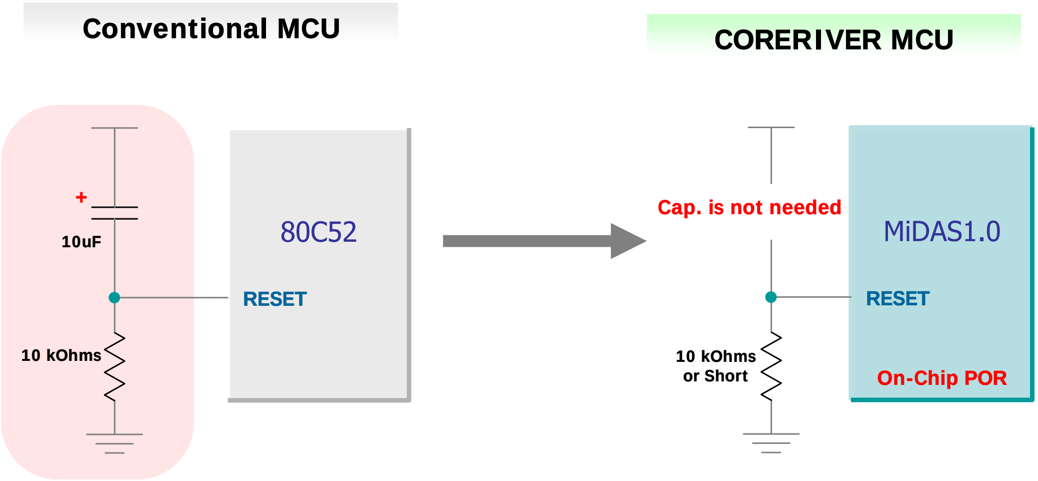


Figure 7-2 Comparison of POR

The MiDAS1.0 family contains the on-chip POR circuit. As a result, no resistor and capacitor is needed for POR. So we can reduce system cost.

For correct operation of the on-chip POR circuit, the supply voltage slope must be in the range from 0.0V/us to 0.5V/us. That is, the supply voltage should be increasing monotonically until it reaches to the normal range.

Downloaded from Elcodis.com electronic components distributor

8 Absolute Maximum Ratings

Table 8-1 Absolute Maximum Ratings

Items	Ranges
Voltage on any pin relative to ground	-0.3V to (V_{DD} + 0.5V)
Voltage on V_{DD} relative to ground	-0.3V to 6.0V
Output Voltage	-0.3V to (V_{DD} + 0.3V)
Operating Temperature	-25°C to +85°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	160°C for 10 seconds

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

9 DC Characteristics

Table 9-1 DC Characteristics

(T_A= -20°C ~ +85°C, V_{DD}=2.7V ~ 5.5V unless otherwise specified)

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Input low voltage	V_{IL}	Except EA	V_{DD}=2.7V~5.5V	-0.5	-	0.2V_{DD}-0.1	V
	V_{IL1}	EA		-0.5	-	0.2V_{DD}-0.3	
Input high voltage	V_{IH}	ALE,PSEN,P0,P1, P2,P3,P4	V_{DD}=2.7V~5.5V	2.0	-	V_{DD}+0.5	V
	V_{IH1}	EA,RESET,XTAL1		0.7V_{DD}		V_{DD}+0.5	
Output low voltage	V_{OL}	P1,P2,P3,P4	V_{DD}=2.7V~5.5V (I_{OL}=1.6mA)	-	-	0.45	V
	V_{OL1}	P0,ALE,PSEN	V_{DD}=2.7V~5.5V (I_{OL}=3.2mA)	-	-	0.45	
Output high voltage	V_{OH}	P1,P2,P3,P4	V_{DD}=2.7V~5.5V (I_{OH}=-60uA)	0.75V_{DD}	-	-	V
			V_{DD}=2.7V~5.5V (I_{OH}=-25uA)	0.76V_{DD}			
			V_{DD}=2.7V~5.5V (I_{OH}=-10uA)	0.78V_{DD}			
	V_{OH1}	ALE,PSEN, P0(external access)	V_{DD}=2.7V~5.5V (I_{OH}=-800uA)	0.75V_{DD}			
			V_{DD}=2.7V~5.5V (I_{OH}=-300uA)	0.78V_{DD}			
			V_{DD}=2.7V~5.5V (I_{OH}=-80uA)	0.8V_{DD}			
Logical 0 input current	I_{IL}	P0,P1,P2,P3,P4	V_{DD}=5.5V (V_{IN}=0.45V)	-	-	-1	uA
Logic 1 to 0 transition current	I_{TL}	P0,P1,P2,P3,P4	V_{DD}=5V ± 10% (V_{IN}=2V)	-	-	-650	uA
Input leakage current	I_{IL}	All pins except XTAL1,XTAL2	V_{IN} = V_{IH} or V_{IL}	-	-	±1	uA
Reset pull-down resistor	R_{RST}	RESET	-	25	-	100	kΩ
Pin capacitance	C_{IO}	All	V_{DD} = 5V	-	10	-	pF

Downloaded from Elcodis.com electronic components distributor

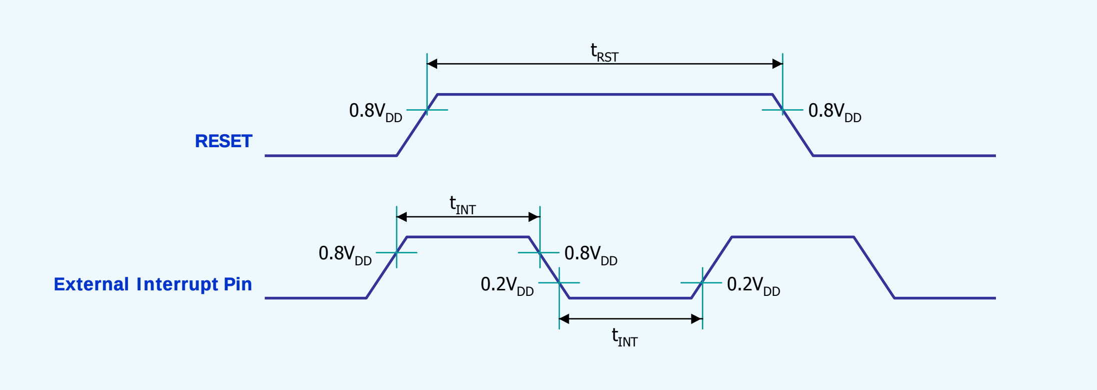
Downloaded from Elcodis.com electronic components distributor

10 AC Characteristics

Table 10-1 AC Characteristics

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

Parameter	Symbol	Pin	Conditions	Value			Unit
				Min.	Typ.	Max.	
Operating Frequency	F_{OSC}	XTAL1, XTAL2	V_{DD}=5V±10%	1	-	40	MHz
			V_{DD}=3V±10%	1	-	20	
RESET Input Width	t_{RST}	RESET	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	-	-	



Downloaded from Elcodis.com electronic components distributor

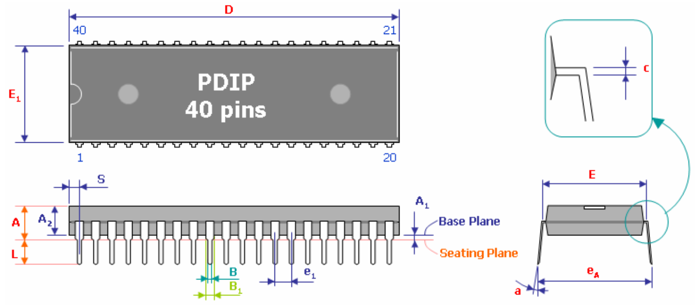
11 ADC Specifications

Table 11-1 ADC Specifications

Parameter		Symbol	Conditions	Value			Unit
				Min.	Typ.	Max.	
Supply voltage		V_{DDADC}	-	2.7	-	5.5	V
Input voltage		V_{INADC}	-	V_{SS}	-	V_{DD}	V
Resolution		RES_{ADC}	-	-	9	-	bit
Operating frequency		F_{ADC}	V_{DD}=4.5~5.5V V_{DD}=2.7~3.3V		-	10 5	MHz
Conversion time		t_{ADC}	-	-	88/F_{ADC}	-	s
Overall accuracy		OA_{ADC}	V_{DD}=5V , F_{ADC} = 10MHz V_{DD}=3V , F_{ADC} = 5MHz	-	±2	±4	LSB
Integral nonlinearity		INL_{ADC}	V_{DD}=5V , F_{ADC} = 10MHz V_{DD}=3V , F_{ADC} = 5MHz	-	±2	±4	LSB
Differential nonlinearity		DNL_{ADC}	V_{DD}=5V , F_{ADC} = 10MHz V_{DD}=3V , F_{ADC} = 5MHz	-	±0.5	±1	LSB
Zero input error		ZIE_{ADC}	V_{DD}=5V , F_{ADC} = 10MHz V_{DD}=3V , F_{ADC} = 5MHz	-	±2	±4	LSB
Full scale error		FSE_{ADC}	V_{DD}=5V , F_{ADC} = 10MHz V_{DD}=3V , F_{ADC} = 5MHz	-	±2	±4	LSB
Analog input capacitance		C_{INADC}	-	-	10	15	pF
ADC current	Active	I_{ADC}	V_{DD} = 5V, F_{ADC} = 10MHz	-	1	2	mA
			V_{DD} = 3V, F_{ADC} = 5MHz	-	0.3	0.6	
	Power down		V_{DD} = 5V	-	-	500	nA

Downloaded from Elcodis.com electronic components distributor

12 Package Dimension

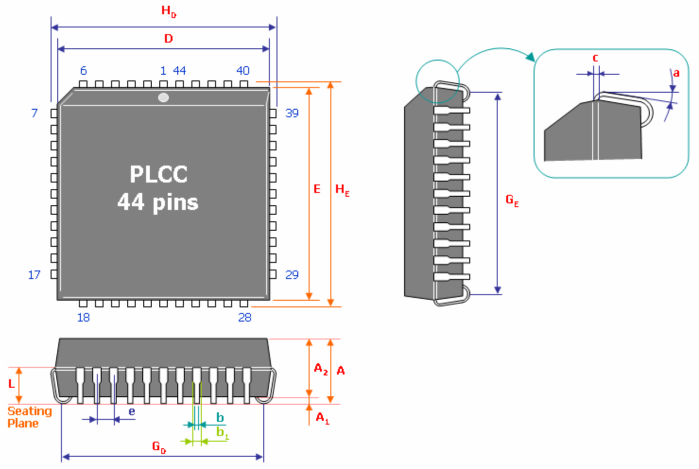


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	-	-	0.200	-	-	5.080
A_1	0.015	-	-	0.381	-	-
A_2	0.150	0.155	0.160	3.810	3.937	4.064
B	0.016	0.018	0.022	0.406	0.457	0.559
B_1	0.045	0.055	0.065	1.143	1.397	1.651
c	0.008	0.010	0.012	0.203	0.254	0.356
D	2.045	2.055	2.075	51.943	52.197	52.705
E	0.590	0.600	0.610	14.986	15.240	15.494
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	0.630	0.650	0.670	16.000	16.510	17.010
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 12-1 PDIP 40-pin Package Dimension



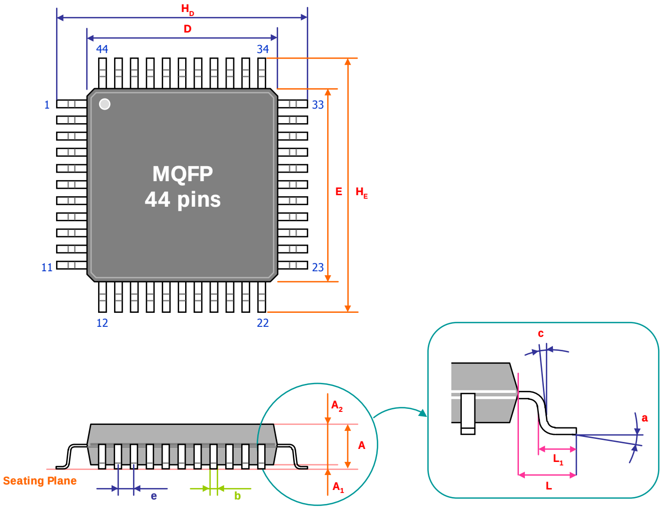
Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.165	-	0.180	4.191	-	4.572
A_1	0.020	-	-	0.508	-	-
A_2	0.145	0.150	0.155	3.683	3.810	3.937
b_1	0.026	0.028	0.032	0.660	0.711	0.813
b	0.013	0.017	0.021	0.330	0.432	0.533
c	0.008	0.010	0.014	0.203	0.254	0.356
D	0.648	0.650	0.658	16.460	16.510	16.710
E	0.648	0.650	0.658	16.460	16.510	16.710
e	0.050 BSC			1.27 BSC		
G_D	0.590	0.610	0.630	14.986	15.494	16.002
G_E	0.590	0.610	0.630	14.986	15.494	16.002
H_D	0.680	0.690	0.700	17.272	17.526	17.780
H_E	0.680	0.690	0.700	17.272	17.526	17.780
L	0.090	0.100	0.120	2.296	2.540	3.048

Notes:

1. Dimension D * E do not include interlead flash.
2. Dimension b_1 dose not include dambar protrusion/intrusion.
3. Controlling dimension: Inches
4. General appearance spec. should be based on final visual inspection spec.

Figure 12-2 PLCC 44-pin Package Dimension

Downloaded from Elcodis.com electronic components distributor

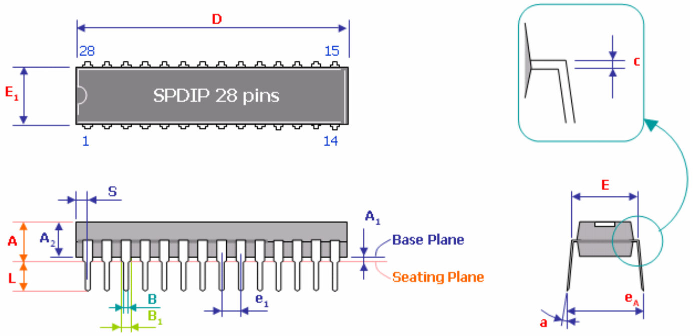


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	–	–	0.091	–	–	2.30
A_1	0.002	–	0.006	0.05	–	0.15
A_2	0.077	0.081	0.085	1.95	2.05	2.15
b	0.012	0.015	0.018	0.30	0.37	0.45
D	0.394 BSC			10.00 BSC		
E	0.394 BSC			10.00 BSC		
e	0.031 BSC			0.80 BSC		
H_D	0.520 BSC			13.20 BSC		
H_E	0.520 BSC			13.20 BSC		
L	–	0.063	–	–	1.60	–
L_1	0.024	0.031	0.039	0.60	0.80	1.00
a	0°	–	8°	0°	–	8°
c	0°	–	–	0°	–	–

Notes:

1. Dimension D * E do not include interlead flash.
2. Controlling dimension: Inches
3. General appearance spec. should be based on final visual inspection spec.

Figure 12-3 MQFP 44-pin Package Dimension

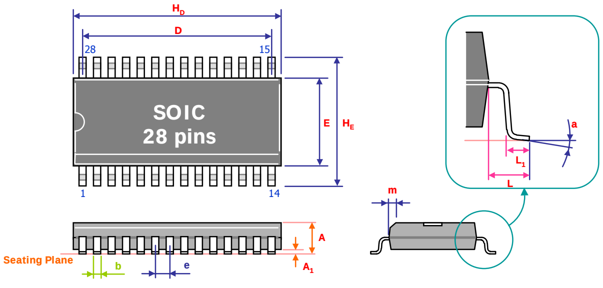


Symbol	Dimension in Inches			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	-	-	0.200	-	-	5.080
A_1	0.015	-	-	0.381	-	-
A_2	0.150	0.155	0.160	3.810	3.937	4.064
B	0.016	0.018	0.022	0.406	0.457	0.559
B_1	0.045	0.055	0.065	1.143	1.397	1.651
c	0.008	0.010	0.012	0.203	0.254	0.356
D	1.445	1.455	1.475	36.703	36.907	37.465
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 12-4 SPDIP 28-pin 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	-	0.65	-	-	16.51	-
E	0.291	0.295	0.299	7.40	7.50	7.60
H_D	0.597	0.655	0.713	17.70	17.90	18.10
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 12-5 SOIC 28-pin Package Dimension

Downloaded from Elcodis.com electronic components distributor

13 Product Numbering System

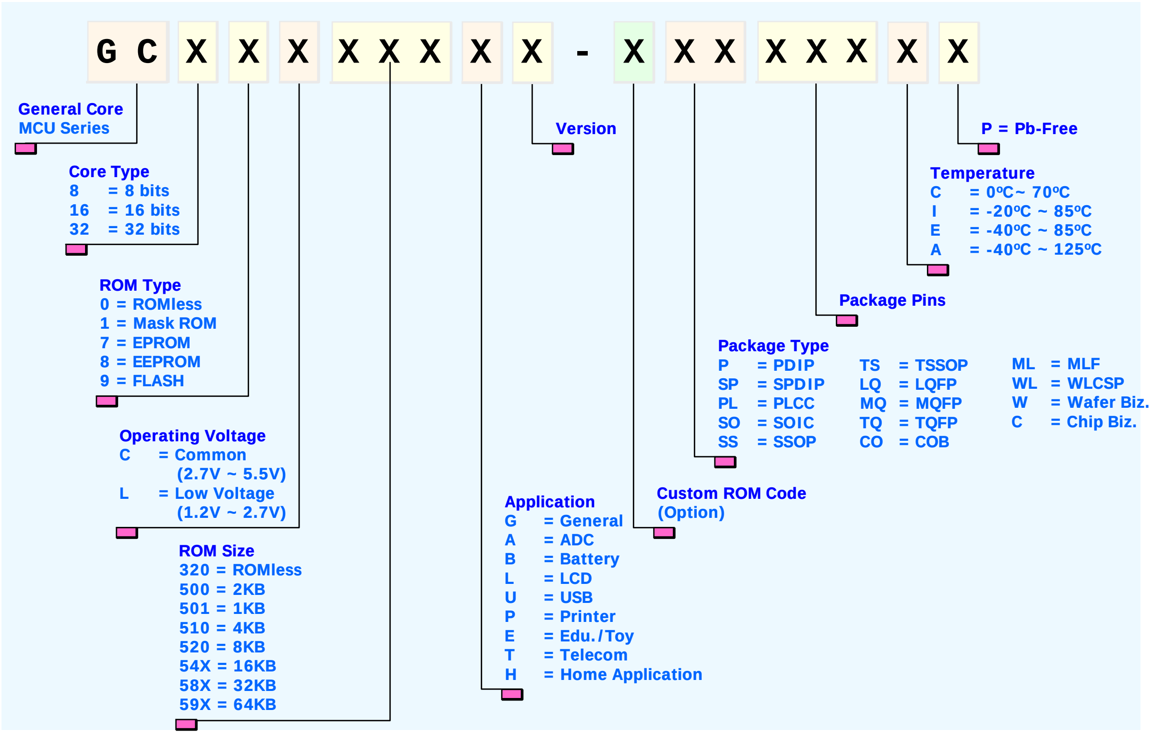


Figure 13-1 Product Numbering System

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

14 Appendix A: Instruction Set

Table 14-1 Note on instruction set and addressing modes

Notation	Description
Rn	Register R0-R7 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an IRAM location (0-127) or a SFR (128-255).
@Ri	8-bit IRAM location (0-255) 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. A branch can be anywhere within the 64Kbyte Program Memory address space.
Addr11	11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2Kbyte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
bit	Direct addressed bit in IRAM or SFR.

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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)

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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)

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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
($PC_{10\text{-}0}$) ← page address

Preliminary

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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,

Preliminary

```
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
```

Downloaded from Elcodis.com electronic components distributor

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
```

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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)

Preliminary

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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.

Preliminary

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.

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Operation: DEC
(DPTR) ← (DPTR) - 1

Downloaded from Elcodis.com electronic components distributor

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

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

Downloaded from Elcodis.com electronic components distributor

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.

Preliminary

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.

Downloaded from Elcodis.com electronic components distributor

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
```

Downloaded from Elcodis.com electronic components distributor

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, repectively.

INC A

Bytes: 1

Cycles: 1

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Downloaded from Elcodis.com electronic components distributor

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
```

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Operation:

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

Downloaded from Elcodis.com electronic components distributor

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
```

Downloaded from Elcodis.com electronic components distributor

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.

Preliminary

Bytes: 1

Cycles: 2

Encoding:

0	1	1	1	0	0	1	1

Operation: JMP
(PC) ← (A) + (DPTR)

Downloaded from Elcodis.com electronic components distributor

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
```

Downloaded from Elcodis.com electronic components distributor

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
```

Preliminary

Downloaded from Elcodis.com electronic components distributor

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
```

Downloaded from Elcodis.com electronic components distributor

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
```

Preliminary

Downloaded from Elcodis.com electronic components distributor

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

$(PC) \leftarrow (PC) + 3$

$(SP) \leftarrow (SP) + 1$

$((SP)) \leftarrow (PC_{7-0})$

$(SP) \leftarrow (SP) + 1$

$((SP)) \leftarrow (PC_{15-8})$

$(PC) \leftarrow (PC) + 3$

$(PC) \leftarrow addr_{15-0}$

Downloaded from Elcodis.com electronic components distributor

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) ← $addr_{15-0}$

Preliminary

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

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))

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Downloaded from Elcodis.com electronic components distributor

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))

Downloaded from Elcodis.com electronic components distributor

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.

Preliminary

Downloaded from Elcodis.com electronic components distributor

MOVX <dest-byte>, <src-byte>

Function: Move External

Description: The MOVX instructions transfer data between the Accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.

In the first type the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.

In the second type of MOVX instruction, the Data Pointer generates a sixteen-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64Kbytes), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example: An external 256byte RAM using multiplexed address/data lines (e.g., an Intel 8155 RAM / I/O / TIMER) is connected to the 8052 Port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12h and 34h. Location 34h of the external RAM holds the value 56h. The instruction sequence,

```
MOVX   A, @R1
MOVX   @R0, A
```

copies the value 56h into both the Accumulator and external RAM location 12h.

MOVX A, @Ri

Bytes: 1

Cycles: 3

Encoding:

1	1	1	0	0	0	1	i

Operation: MOVX
(A) ← ((Ri))

Downloaded from Elcodis.com electronic components distributor

MOVX A, @DPTR

Bytes: 1

Cycles: 3

Encoding:

1 1 1 0	0 0 0 0

Operation: MOVX
(A) ← ((DPTR))

MOVX @Ri, A

Bytes: 1

Cycles: 3

Encoding:

1 1 1 1	0 0 1 i

Operation: MOVX
((Ri)) ← (A)

MOVC @DPTR, A

Bytes: 1

Cycles: 3

Encoding:

1 1 1 1	0 0 0 0

Preliminary

Operation: MOVX
(DPTR) ← (A)

Downloaded from Elcodis.com electronic components distributor

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The low-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
$(A)_{7-0}$, $(B)_{15-8}$ ← (A) X (B)

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Downloaded from Elcodis.com electronic components distributor

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
$(A) \leftarrow (A) \vee (Rn)$

ORL A, direct

Bytes: 2

Cycles: 2

Encoding:

0 1 0 0	0 1 1 i	direct address

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Encoding:

1 1 0 1	0 0 0 0	direct address

Operation: POP
(direct) ← ((SP))
(SP) ← (SP) – 1

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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 location 0123h.

Bytes: 1

Cycles: 2

Encoding:

0 0 1 0	0 0 1 0

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

Preliminary

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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
$(A0) \leftarrow (A7)$

Preliminary

Downloaded from Elcodis.com electronic components distributor

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)

Downloaded from Elcodis.com electronic components distributor

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
$(A7) \leftarrow (A0)$

Preliminary

Downloaded from Elcodis.com electronic components distributor

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
$(A_n) \leftarrow (A_n + 1)$ $n = 0 \sim 6$
$(A7) \leftarrow (C)$
$(C) \leftarrow (A0)$

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

SETB bit

Bytes: 2

Cycles: 2

Encoding:

1 1 0 1	0 0 1 0	bit address

Operation: SETB
(bit) ← 1

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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.

Preliminary

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)

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

Downloaded from Elcodis.com electronic components distributor

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))

Downloaded from Elcodis.com electronic components distributor

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

$(A_{3-0}) \leftrightarrow ((Rn_{3-0}))$

Preliminary

Downloaded from Elcodis.com electronic components distributor

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

Downloaded from Elcodis.com electronic components distributor

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

Preliminary

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

Downloaded from Elcodis.com electronic components distributor

15 Appendix B: SFR description

15.1 Port 0 Register (P0)

Bit No.	7	6	5	4	3	2	1	0	
80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	P0
	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 port functions as a multiplexed address/data bus during external memory access, and as a general purpose I/O port on devices which incorporate internal program memory. During external memory cycles, this port will contain the LSB of the address when ALE is high, and data when ALE is low. When used as a general purpose I/O, this port is open-drain and can select pull-up resistors optionally. If SFR P0SEL is set to “1”, writing a 1 to this port will place it in a high impedance mode, which is necessary if the pin is to be used as an input, and if not, pull-up resistor in port 0 is on. Pull-up resistors are not required when used as a memory interface.

Only P0.0 can be configurable as analog input (ADC input channel 0).

15.2 Stack Pointer Register (SP)

Bit No.	7	6	5	4	3	2	1	0	
81h	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0	SP
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(1)	R/W(1)	R/W(1)	

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.

15.3 Data Pointer Low Register (DPL)

Bit No.	7	6	5	4	3	2	1	0	
82h	DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0	DPL
	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.

Downloaded from Elcodis.com electronic components distributor

15.4 Data Pointer High Register (DPH)

Bit No.	7	6	5	4	3	2	1	0	
83h	DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0	DPH
	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.

15.5 Power Control Register (PCON)

Bit No.	7	6	5	4	3	2	1	0	
87h	SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL	PCON
	R/W(0)	R(0)		R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Symbol	Function
SMOD1	This bit doubles the serial port baud rate in mode 1, 2, and 3 when set to 1.
SMOD0	0: SCON.7 controls the SM0 function defined for the SCON register.
-	Not implemented, reserved for future use. **Note**: User software should not write 1s to reserved bits. These bits may be used in future products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.
POF	Power Off flag. When power-on, this bit will be set by H/W.
GF1	General purpose flag bit
GF0	General purpose flag bit
PD	Power down (Stop) mode enable. Setting this bit causes the MiDAS1.0 family to go into the POWER DOWN mode. In this mode all the clocks are stopped and program execution is frozen.
IDL	Idle mode enable. Setting this bit causes the MiDAS1.0 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.

15.6 Timer 0/1 Control Register (TCON)

Downloaded from Elcodis.com electronic components distributor

Bit No.	7	6	5	4	3	2	1	0	
88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON
	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	Function
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 and is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine. 0: No Timer 1 overflow has been detected. 1: Timer 1 has overflowed its maximum count
TR1	Timer 1 Run Control. This bit enables/disables the operation of Timer 1. Halting this timer will preserve the current count in TH1 and TL1. 0: Timer 1 is halted. 1: Timer 1 is enabled.
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 and is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine. 0: No Timer 0 overflow has been detected. 1: Timer 0 has overflowed its maximum count
TR0	Timer 0 Run Control. This bit enables/disables the operation of Timer 0. Halting this timer will preserve the current count in TH0 and TL0. 0: Timer 0 is halted. 1: Timer 0 is enabled.
IE1	Interrupt 1 Edge Detect. This bit is set when an edge/level of the type defined by IT1 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 /INT1 pin.
IT1	Interrupt 1 Type Select. This bit selects whether the /INT1 pin will detect edge or level triggered interrupts. 0: /INT1 is level triggered. 1: /INT1 is edge triggered.
IE0	Interrupt 0 Edge Detect. This bit is set when an edge/level of the type defined by IT0 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 /INT0 pin.

Downloaded from Elcodis.com electronic components distributor

IT0	Interrupt 0 Type Select. This bit selects whether the /INT0 pin will detect edge or level triggered interrupts. 0: /INT0 is level triggered. 1: /INT0 is edge triggered.

15.7 Timer Mode Control Register (TMOD)

Bit No.	7	6	5	4	3	2	1	0	
89h	GATE	C/T	M1	M0	GATE	C/T	M1	M0	TMOD
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

<table>
<tr><th>Symbol</th><th colspan="3">Function</th></tr>
<tr><td>GATE</td><td colspan="3">Timer 1 Gate Control. This bit enables/disables the ability of Timer 1 to increment.
0: Timer 1 will clock when TR1=1, regardless of the state of /INT1.
1: Timer 1 will clock only when TR1=1 and /INT1=1.</td></tr>
<tr><td>C/T</td><td colspan="3">Timer 1 Counter/Timer Select.
0: Timer 1 is incremented by internal clocks.
1: Timer 1 is incremented by pulses on T1 when TR1 (TCON.6) is 1.</td></tr>
<tr><td rowspan="6">M1, M0</td><td colspan="3">Timer 1 Mode Select. These bits select the operating mode of Timer 1.</td></tr>
<tr><td>M1</td><td>M0</td><td>Mode</td></tr>
<tr><td>0</td><td>0</td><td>Mode 0: 8 bits with 5-bit prescale</td></tr>
<tr><td>0</td><td>1</td><td>Mode 1: 16 bits</td></tr>
<tr><td>1</td><td>0</td><td>Mode 2: 8 bits with auto-reload</td></tr>
<tr><td>1</td><td>1</td><td>Mode 3: Timer 1 is halted, but holds its count.</td></tr>
<tr><td>GATE</td><td colspan="3">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 /INT0.
1: Timer 0 will clock only when TR0=1 and /INT0=1.</td></tr>
<tr><td>C/T</td><td colspan="3">Timer 0 Counter/Timer Select.
0: Timer 0 is incremented by internal clocks.
1: Timer 0 is incremented by pulses on T0 when TR0 (TCON.4) is 1.</td></tr>
<tr><td rowspan="3">M1, M0</td><td colspan="3">Timer 0 Mode Select. These bits select the operating mode of Timer 0. When Timer 0 is in mode 3, TL0 is started/stopped by TR0 and TH0 is started/stopped by TR1. Run control for Timer 1 is then provided via the Timer 1 mode selection.</td></tr>
<tr><td>M1</td><td>M0</td><td>Mode</td></tr>
<tr><td>0</td><td>0</td><td>Mode 0: 8 bits with 5-bit prescale</td></tr>
</table>

Downloaded from Elcodis.com electronic components distributor

	0	1	Mode 1: 16 bits
	1	0	Mode 2: 8 bits with auto-reload
	1	1	Mode 3: Timer 1 is halted, but holds its count.

15.8 Timer 0 Low Byte Register (TL0)

Bit No.	7	6	5	4	3	2	1	0	
8Ah	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0	TL0
	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.

15.9 Timer 1 Low Byte Register (TL1)

Bit No.	7	6	5	4	3	2	1	0	
8Bh	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0	TL1
	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.

15.10 Timer 0 High Byte Register(TH0)

Bit No.	7	6	5	4	3	2	1	0	
8Ch	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0	TH0
	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.

15.11 Timer 1 High Byte Register (TH1)

Bit No.	7	6	5	4	3	2	1	0	
8Dh	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0	TH1
	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.

Downloaded from Elcodis.com electronic components distributor

15.12 Clock Control Register (CKCON)

Bit No.	7	6	5	4	3	2	1	0	
8Eh	WD1	WD0	T2M	T1M	T0M	-	-	-	CKCON
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	-	-	-	

<table>
<tr><th>Symbol</th><th colspan="4">Function</th></tr>
<tr><td rowspan="6">WD1, WD0</td><td colspan="4">Watchdog Timer Mode Select. These bits determine the Watchdog timer time-out period. The timer divides the crystal frequency by a programmable value as shown below. The divider value is expressed in clock (crystal) cycles. Note that the reset time-out is 512 clocks longer than the interrupt, regardless of whether the interrupt is enabled.</td></tr>
<tr><td>WD1</td><td>WD0</td><td>Interrupt Divider</td><td>Reset Divider</td></tr>
<tr><td>0</td><td>0</td><td>2^{17}</td><td>2^{17} + 512</td></tr>
<tr><td>0</td><td>1</td><td>2^{20}</td><td>2^{20} + 512</td></tr>
<tr><td>1</td><td>0</td><td>2^{23}</td><td>2^{23} + 512</td></tr>
<tr><td>1</td><td>1</td><td>2^{26}</td><td>2^{26} + 512</td></tr>
<tr><td>T2M</td><td colspan="4">Timer 2 Clock Select. This bit controls the division of the system clock that drives Timer 2. This bit has no effect when the timer is in baud rate generator or clock output modes. Clearing this bit to 0 maintains 80C52 compatibility. This bit has no effect on instruction cycle timing.
0: Timer 2 uses a divide by 12 of the crystal frequency.
1: Timer 2 uses a divide by 4 of the crystal frequency.</td></tr>
<tr><td>T1M</td><td colspan="4">Timer 1 Clock Select. This bit controls the division of the system clock that drives Timer 1. Clearing this bit to 0 maintains 80C52 compatibility. This bit has no effect on instruction cycle timing.
0: Timer 1 uses a divide by 12 of the crystal frequency.
1: Timer 1 uses a divide by 4 of the crystal frequency.</td></tr>
<tr><td>T0M</td><td colspan="4">Timer 0 Clock Select. This bit controls the division of the system clock that drives Timer 0. Clearing this bit to 0 maintains 80C52 compatibility. This bit has no effect on instruction cycle timing.
0: Timer 0 uses a divide by 12 of the crystal frequency.
1: Timer 0 uses a divide by 4 of the crystal frequency.</td></tr>
<tr><td>-</td><td colspan="4">Reserved. These bits will read 0</td></tr>
</table>

Downloaded from Elcodis.com electronic components distributor

15.13 Port 1 Register (P1)

Bit No.	7	6	5	4	3	2	1	0	
90h	P1.7 /INT5	P1.6 INT4	P1.5 /INT3	P1.4 INT2	P1.3 ADC3	P1.2 ADC2	P1.1 T2EX ADC1	P1.0 T2 PWM1	P1
	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. In addition, all the pins have an alternative function listed below. Each of the functions is controlled by several other SFRs. The associated Port 1 latch bit must contain a logic one before the pin can be used in its alternative function capacity.

Symbol	Function
/INT5	External Interrupt 5. A falling edge on this pin will cause an external interrupt 5 if enabled.
INT4	External Interrupt 4. A rising edge on this pin will cause an external interrupt 4 if enabled.
/INT3	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.
ADC3	ADC Analog Input 3.
ADC2	ADC Analog Input 2.
ADC1	ADC Analog Input 1
T2EX	Timer 2 Capture/Reload Trigger. A 1-to-0 transition on this pin will cause the value in the T2 registers to be transferred into the capture registers if enabled by EXEN2 (T2CON.3). When in auto-reload mode, a 1-to-0 transition on this pin will reload the Timer 2 register with the value in RCAP2L and RCAP2H if enabled by EXEN2.
T2	Timer 2 External Input. A 1-to-0 transition on this pin will cause Timer 2 to increment or decrement depending on the timer configuration.
PWM1	PWM Waveform Output 1.

Downloaded from Elcodis.com electronic components distributor

15.14 External Interrupt Flag Register (EXIF)

Bit No.	7	6	5	4	3	2	1	0	
91h	IE5	IE4	IE3	IE2	XT	-	-	BGS	EXIF
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R(1)			R/W(1)	

Symbol	Function
IE5	External Interrupt 5 Flag. This bit will be set when a falling edge is detected on /INT5. This bit must be cleared manually by software. Setting this bit in software will cause an interrupt if enabled.
IE4	External Interrupt 4 Flag. This bit will be set when a rising edge is detected on INT4. This bit must be cleared manually by software. Setting this bit in software will cause an interrupt if enabled.
IE3	External Interrupt 3 Flag. This bit will be set when a falling edge is detected on /INT3. 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	Crystal Source Select. Don't touch this bit. This bit is set to "1" after a power-on reset, and unchanged by all other forms of reset. 1: The crystal oscillator is selected as the clock source.
BGS	Band-Gap Select. This bit enables/disables the band-gap reference during Stop mode. Disabling the band-gap reference provides significant power savings in Stop mode, but sacrifices the ability to perform a power fail interrupt or 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 Stop mode but will function during normal operation. 1: The band-gap reference will operate in Stop mode.

Downloaded from Elcodis.com electronic components distributor

15.15 Serial Port Control Register (SCON)

Bit No.	7	6	5	4	3	2	1	0	
98h	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON
	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	Function			
SM0, SM1	Serial Port Mode. These bits control the mode of serial port.			
	SM0	SM1	Mode	Period
	0	0	0	8-bit shift register (OSC/4)
	0	1	1	8-bit UART (Variable)
	1	0	2	9-bit UART (OSC/32 or OSC/16)
	1	1	3	9-bit UART (Variable)
SM2	Automatic Address Recognition. The function of this bit is dependent on the serial port mode. Mode 1: When set, reception is ignored if invalid stop bit received. Mode 2/3: When this bit is set, automatic address recognition is enabled in modes 2 and 3. This will prevent the RI bit from being set, and an interrupt being asserted, if the 9th bit received is not 1.			
REN	Receive Enable. This bit enables/disables the serial port receiver shift register. 0: serial port reception disabled. 1: serial port receiver enabled (modes 1, 2 and 3). Initiate synchronous reception (mode 0).			
TB8	9th Transmission Bit State. This bit defines the state of the 9th transmission bit in serial port modes 2 and 3.			
RB8	9th Received Bit State. This bit identifies the state of the 9th reception bit of received data in serial port modes 2 and 3. In serial port mode 1, when SM2=0, RB8 is the state of the stop bit. RB8 is not used in mode 0.			
TI	Transmitter Interrupt Flag. This bit indicates that data in the serial port buffer has been completely shifted out. In serial port mode 0, TI is set at the end of the 8th data bit. In all other modes, this bit is set at the end of the last data bit. This bit must be manually cleared by software.			
RI	Receiver Interrupt Flag. This bit indicates that a byte of data has been received in the serial port buffer. In serial port mode 0, RI is set at the end of the 8th bit. In serial port mode 1, RI is set after the last sample of the incoming stop bit subject to the state of			

Downloaded from Elcodis.com electronic components distributor

	SM2. In modes 2 and 3, RI is set after the last sample of RB8. This bit must be manually cleared by software.

15.16 Serial Data Buffer Register (SBUF)

Bit No.	7	6	5	4	3	2	1	0	
99h	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0	SBUF
	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 serial transmit and receive buffers are separate registers, but both are addressed at the same location.

15.17 Port 2 Register (P2)

Bit No.	7	6	5	4	3	2	1	0	
A0h	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	P2
	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 port functions as an address bus during external memory access, and as a general purpose I/O port on devices which incorporate internal program memory. During external memory cycles, this port will contain the MSB of the address.

15.18 Port 4 Register (P4)

Bit No.	7	6	5	4	3	2	1	0	
A5h	-	-	-	-	P4.3	P4.2	P4.1	P4.0	P4
	-	-	-	-	R/W(1)	R/W(1)	R/W(1)	R/W(1)	

This register functions as a general purpose I/O port in the 44-PLCC/44-LQFP package.

15.19 Port 4 Pull-up Control Register (P4SEL)

Bit No.	7	6	5	4	3	2	1	0	
A6h	-	-	-	-	P4SEL.3	P4SEL.2	P4SEL.1	P4SEL.0	P4SEL
	-	-	-	-	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Downloaded from Elcodis.com electronic components distributor

This register enables/disables the pull-up resistors. If bit with 0 indicates the pull-up resistor is on.

15.20 Interrupt Enable Register (IE)

Bit No.	7	6	5	4	3	2	1	0	
A8h	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0	IE
	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	Function
EA	Global Interrupt Enable. This bit controls the global masking of all interrupts except Power-Fail interrupt, which is enabled by the EPFI bit (WDCON.5). 0: Disable all interrupt sources. This bit overrides individual interrupt mask settings. 1: Enable all individual interrupt masks. Individual interrupts will occur if enabled.
EADC	Enable ADC Interrupt. This bit controls the masking of the ADC interrupt. 0: Disable all ADC interrupts. 1: Enable interrupt requests generated by the ADCF flag (ADCON.4)
ET2	Enable Timer 2 Interrupt. This bit controls the masking of the Timer 2 interrupt. 0: Disable all Timer 2 interrupts. 1: Enable interrupt requests generated by the TF2 flag (T2CON.7)
ES0	Enable Serial port Interrupt. This bit controls the masking of the serial port interrupt. 0: Disable all serial port interrupts. 1: Enable interrupt requests generated by the RI (SCON.0) or TI (SCON.1) flags
ET1	Enable Timer 1 Interrupt. This bit controls the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupts. 1: Enable interrupt requests generated by the TF1 flag (TCON.7).
EX1	Enable External Interrupt 1. This bit controls the masking of external interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the /INT1 pin.
ET0	Enable Timer 0 Interrupt. This bit controls the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupts. 1: Enable interrupt requests generated by the TF0 flag (TCON.5).
EX0	Enable External Interrupt 0. This bit controls the masking of external interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the /INT0 pin.

Downloaded from Elcodis.com electronic components distributor

15.21 Slave Address Register (SADDR)

Bit No.	7	6	5	4	3	2	1	0	
A9h	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0	SADDR
	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 register is programmed with the given or broadcast address assigned to serial port.

15.22 Port 3 Register (P3)

Bit No.	7	6	5	4	3	2	1	0	
B0h	P3.7 /RD	P3.6 /WR	P3.5 T1	P3.4 T0	P3.3 /INT1	P3.2 /INT0	P3.1 TXD	P3.0 RXD	P3
	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. In addition, all the pins have an alternative function listed below. Each of the functions is controlled by other SFRs. The associated Port 3 latch bit must contain in a logic one before the pin can be used in its alternative function capacity.

Symbol	Function
/RD	External Data Memory Read Strobe. This pin provides an active low read strobe to an external memory device.
/WR	External Data Memory Write Strobe. This pin provides an active low write strobe to an external memory device.
T1	Timer/Counter 1 External Input. A 1-to-0 transition on this pin will increment Timer 1.
T0	Timer/Counter 0 External Input. A 1-to-0 transition on this pin will increment Timer 0.
/INT1	External Interrupt 1. A falling edge/low level on this pin will cause an external interrupt 1 if enabled.
/INT0	External Interrupt 0. A falling edge/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 1,2,3 and emits the synchronizing clock in serial port mode 0.
RXD	Serial Port Receive. This pin receives the serial port data in serial port modes 1,2,3 and is a bi-directional data transfer pin in serial port mode 0.

Downloaded from Elcodis.com electronic components distributor

15.23 Interrupt Priority High (IPH) & Interrupt Priority Register (IP)

Bit No.	7	6	5	4	3	2	1	0	
B7h	-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	IPH
	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Bit No.	7	6	5	4	3	2	1	0	
B8h	-	PADC	PT2	PS	PT1	PX1	PT0	PX0	IP
	R(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

<table>
<tr><th>Symbol</th><th colspan="3">Function</th></tr>
<tr><td>-</td><td colspan="3">Reserved.</td></tr>
<tr><td rowspan="6">PADCH
PADC</td><td colspan="3">ADC Interrupt Priority. These bits control the priority of the ADC interrupt up to four levels.</td></tr>
<tr><td>PADCH</td><td>PADC</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>Priority level 0</td></tr>
<tr><td>0</td><td>1</td><td>Priority level 1</td></tr>
<tr><td>1</td><td>0</td><td>Priority level 2</td></tr>
<tr><td>1</td><td>1</td><td>Priority level 3</td></tr>
<tr><td rowspan="6">PT2H
PT2</td><td colspan="3">Timer 2 Interrupt Priority. These bits control the priority of the Timer 2 interrupt up to four levels.</td></tr>
<tr><td>PT2H</td><td>PT2</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>Priority level 0</td></tr>
<tr><td>0</td><td>1</td><td>Priority level 1</td></tr>
<tr><td>1</td><td>0</td><td>Priority level 2</td></tr>
<tr><td>1</td><td>1</td><td>Priority level 3</td></tr>
<tr><td rowspan="6">PSH
PS</td><td colspan="3">Serial port Interrupt Priority. These bits control the priority of the serial port interrupt up to four levels.</td></tr>
<tr><td>PSH</td><td>PS</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>Priority level 0</td></tr>
<tr><td>0</td><td>1</td><td>Priority level 1</td></tr>
<tr><td>1</td><td>0</td><td>Priority level 2</td></tr>
<tr><td>1</td><td>1</td><td>Priority level 3</td></tr>
</table>

Downloaded from Elcodis.com electronic components distributor

PT1H PT1	Timer 1 Interrupt Priority. These bits control the priority of the Timer 1 interrupt up to four levels.		
	PT1H	PT1	Description
	0	0	Priority level 0
	0	1	Priority level 1
	1	0	Priority level 2
	1	1	Priority level 3
PX1H PX1	External Interrupt 1 Priority. These bits control the priority of the external interrupt 1 up to four levels.		
	PX1H	PX1	Description
	0	0	Priority level 0
	0	1	Priority level 1
	1	0	Priority level 2
	1	1	Priority level 3
PT0H PT0	Timer 0 Interrupt Priority. These bits control the priority of the Timer 0 interrupt up to four levels.		
	PT0H	PT0	Description
	0	0	Priority level 0
	0	1	Priority level 1
	1	0	Priority level 2
	1	1	Priority level 3
PX0H PX0	External Interrupt 0 Priority. These bits control the priority of the external interrupt 0 up to four levels.		
	PX0H	PX0	Description
	0	0	Priority level 0
	0	1	Priority level 1
	1	0	Priority level 2
	1	1	Priority level 3

15.24 Slave Address Mask Enable Register (SADEN)

Bit No.	7	6	5	4	3	2	1	0	
B9h	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0	SADEN
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Downloaded from Elcodis.com electronic components distributor

This register functions as a mask when comparing serial port addresses for automatic address recognition. When a bit in this register is set, the corresponding bit location in the SADDR will be exactly compared with the incoming serial port data to determine if a receiver interrupt should be generated. When a bit in this register is cleared, the corresponding bit in the SADDR becomes a don’t care and is not compared against the incoming data. All incoming data will generate a receiver interrupt when this register is cleared.

15.25 Power Management Register (PMR)

Bit No.	7	6	5	4	3	2	1	0	
C4h	-	-	-	-	-	ALEOFF	-	-	PMR
	-	-	-	-	-	R/W(0)	-	-	

Symbol	Function
-	Reserved.
ALEOFF	ALE Disable. This bit disables the expression of the ALE signal on the device pin during all on-board program and data memory accesses. External memory accesses will automatically enable ALE independent of ALEOFF. 0: ALE expression is enabled. 1: ALE expression is disabled.

15.26 Status Register (STATUS)

Bit No.	7	6	5	4	3	2	1	0	
C5h	-	-	-	XTUP	-	-	-	-	STATUS
	-	-	-	R(0)	-	-	-	-	

Symbol	Function
-	Reserved.
XTUP	Crystal Oscillator Warm-up Status. This bit indicates whether the CPU crystal oscillator has completed the 65,536 cycle warm-up and is ready to operate from the external crystal or oscillator. This bit is cleared by H/W when Power-on Reset, during Power Down wake-up. This bit is set to 1 following a XTAL stabilization by H/W.

Downloaded from Elcodis.com electronic components distributor

15.27 Timer 2 Control Register (T2CON)

Bit No.	7	6	5	4	3	2	1	0	
C8h	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	T2CON
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

<table>
<tr><th>Symbol</th><th colspan="4">Function</th></tr>
<tr><td>TF2</td><td colspan="4">Timer 2 Overflow Flag. This flag will be set when Timer 2 overflow from FFFFh to 0000h or the counter equal to the capture register in down count mode. It must be cleared by software. TF2 will only be set if RCLK and TCLK are both cleared to 0.</td></tr>
<tr><td rowspan="7">EXF2</td><td colspan="4">Timer 2 External Flag. A negative transition on the T2EX pin (P1.1) or Timer 2 underflow/overflow will cause this flag to set based on the CP/RL2 (T2CON.0), EXEN2 (T2CON.3), and DCEN (T2MOD.0) bits. If set by a negative transition, this flag must be cleared to 0 by software. Setting this bit in software or detection of a negative transition on the T2EX pin will force a timer 2 interrupt if enabled.</td></tr>
<tr><td>CP/RL2</td><td>EXEN2</td><td>DCEN</td><td>Result</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>Negative transition on P1.1 will not affect EXF2 bit.</td></tr>
<tr><td>1</td><td>1</td><td>X</td><td>Negative transition on P1.1 will set EXF2 bit.</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Negative transition on P1.1 will not affect EXF2 bit.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Negative transition on P1.1 will set EXF2 bit.</td></tr>
<tr><td>0</td><td>X</td><td>1</td><td>EXF2 toggles whenever Timer 2 underflow/overflows and can be used as a 17th bit of resolution. In this mode, EXF2 will not cause an interrupt.</td></tr>
<tr><td>RCLK</td><td colspan="4">Receive Clock Flag. This bit determines the serial port timebase when receiving data in serial modes 1 or 3.
0: Timer 1 overflow is used to determine receiver baud rate for serial port.
1: Timer 2 overflow is used to determine receiver baud rate for serial port. Setting this bit will force Timer 2 into baud rate generation mode. The timer will operate from a divide by 2 of the external clock.</td></tr>
</table>

Downloaded from Elcodis.com electronic components distributor

TCLK	Transmit Clock Flag. This bit determines the serial port timebase when transmitting data in serial modes 1 or 3. 0: Timer 1 overflow is used to determine transmitter baud rate for serial port. 1: Timer 2 overflow is used to determine transmitter baud rate for serial port. Setting this bit will force Timer 2 into baud rate generation mode. The timer will operate from a divide by 2 of the external clock.
EXEN2	Timer 2 External Enable. This bit enables the capture/reload function on the T2EX pin if Timer 2 is not generating baud rates for the serial port. 0: Timer 2 will ignore all external events at T2EX. 1: Timer 2 will capture or reload a value if a negative transition is detected on the T2EX pin.
TR2	Timer 2 Run Control. This bit enables/disables the operation of Timer 2. Halting this timer will preserve the current count in TH2 and TL2. 0: Timer 2 is halted. 1: Timer 2 is enabled.
C/T2	Counter/Timer Select. This bit determines whether Timer 2 will function as a timer or counter. Independent of this bit, Timer 2 runs at 2 clocks per tick when used in either baud rate generator or clock output mode. 0: Timer 2 functions as a timer. The speed of Timer 2 is determined by the T2M bit (CKCON.5). 1: Timer 2 will count negative transitions on the T2 pin (P1.0).
CP/RL2	Capture/Reload Select. This bit determines whether the capture or reload function will be used for Timer 2. If either RCLK or TCLK is set, this bit will not function and the timer will function in an auto-reload mode following each overflow. 0: Auto-reloads will occur when Timer 2 overflows or a falling edge is detected on T2EX if EXEN2=1. 1: Timer 2 captures will occur when a falling edge is detected on T2EX if EXEN2=1.

15.28 Timer 2 Mode Register (T2MOD)

Bit No.	7	6	5	4	3	2	1	0	
C9h	-	-	-	-	-	-	T2OE	DCEN	T2MOD
	-	-	-	-	-	-	R/W(0)	R/W(0)	

Downloaded from Elcodis.com electronic components distributor

Symbol	Function
-	Reserved. Read data will be indeterminate.
T2OE	Timer 2 Output Enable. This bit enables/disables the clock output function of the T2 pin (P1.0). 0: The T2 pin functions as either a standard port pin or as a counter input for Timer 2. 1: Timer 2 will drive the T2 pin with a clock output if C/T2=0. Also, Timer 2 roll-overs will not cause interrupts.
DCEN	Down Count Enable. This bit, in conjunction with the T2EX pin, controls the direction that Timer 2 counts in 16-bit auto-reload mode.

15.29 Timer 2 Capture/Reload Low Byte Register (RCAP2L)

Bit No.	7	6	5	4	3	2	1	0	
CAh	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0	RCAP2L
	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 used to capture the TL2 value when Timer 2 is configured in capture mode. RCAP2L is also used as the low byte of a 16-bit reload value when Timer 2 is configured in auto-reload mode.

15.30 Timer 2 Capture/Reload High Byte Register (RCAP2H)

Bit No.	7	6	5	4	3	2	1	0	
CBh	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0	RCAP2H
	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 used to capture the TH2 value when Timer 2 is configured in capture mode. RCAP2H is also used as the high byte of a 16-bit reload value when Timer 2 is configured in auto-reload mode.

15.31 Timer 2 Low Byte Register(TL2)

Bit No.	7	6	5	4	3	2	1	0	
CCh	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0	TL2
	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 low byte of Timer 2.

Downloaded from Elcodis.com electronic components distributor

15.32 Timer 2 High Byte Register (TH2)

Bit No.	7	6	5	4	3	2	1	0	
CDh	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0	TH2
	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 high byte of Timer 2.

15.33 Program Status Word Register (PSW)

Bit No.	7	6	5	4	3	2	1	0	
D0h	CY	AC	F0	RS1	RS0	OV	F1	P	PSW
	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 colspan="4">Function</th></tr>
<tr><td>CY</td><td colspan="4">Carry Flag. This bit is set when if the last arithmetic operation resulted in a carry (during addition) or a borrow (during subtraction). Otherwise it is cleared to 0 by all arithmetic operations.</td></tr>
<tr><td>AC</td><td colspan="4">Auxiliary Carry Flag. This bit is set to 1 if the last arithmetic operation resulted in a carry into (during addition), or a 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 colspan="4">User Flag 0. This is a general purpose flag for software control.</td></tr>
<tr><td rowspan="6">RS1, RS0</td><td colspan="4">Register Bank Select. These bits select which register bank is addressed during register accesses.</td></tr>
<tr><td>RS1</td><td>RS0</td><td>Register Bank</td><td>Address</td></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>
<tr><td>OV</td><td colspan="4">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 colspan="4">User Flag 1. This is a general purpose flag for software control.</td></tr>
</table>

Downloaded from Elcodis.com electronic components distributor

P	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.

15.34 Watchdog Control & Power Status Register (WDCON)

Bit No.	7	6	5	4	3	2	1	0	
D8h	-	POR	EPFI	PFI	WDIF	WTRF	EWT	RWT	WDCON
	-	R/W(1)	R/W(0)	R/W(1)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Symbol	Function
-	Reserved.
POR	Power-On-Reset Flag. This bit indicates whether the last reset was a power-on-reset. This bit is set following a power-on-reset and unaffected by all other resets. 0: Last reset was from a source other than a power-on-reset. 1: Last reset was a power-on-reset.
EPFI	Enable Power fail Interrupt. This bit enables/disables the ability of the internal band-gap reference to generate a power-fail interrupt when V_{CC} falls below approximately 4.0V. While in Stop mode, both this bit and the band-gap select bit BGS (EXIF.0) must be set to enable the power-fail interrupt. 0: Power-fail interrupt disabled. 1: Power-fail interrupt enabled during normal operation. Power-fail interrupt enabled in Stop mode if BGS is set.
PFI	Power Fail Interrupt Flag. When set, this bit indicates that a power-fail interrupt has occurred. 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 power-fail interrupt, if enabled.
WDIF	Watchdog Interrupt Flag. This bit, in conjunction with the Watchdog timer interrupt enable bit EWDI (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	EWDI	WDIF	Result
X	X	0	No watchdog event has occurred.

Downloaded from Elcodis.com electronic components distributor

	0	0	1	Watchdog time-out has expired. No interrupt has been generated.
	0	1	1	Watchdog interrupt has occurred.
	1	0	1	Watchdog time-out has expired. No interrupt has been generated. Watchdog timer reset will occur in 512 cycles if RWT is not strobed.
	1	1	1	Watchdog interrupt has occurred. Watchdog timer reset will occur in 512 cycles if RWT is not set.
WTRF	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.			
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	Reset 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.			

15.35 PWM 0 Control Register (PWM0CON)

Bit No.	7	6	5	4	3	2	1	0	
DCh	P0SEL	PS2_P0	PS1_P0	PS0_P0	MODE_P0	RL_P0	CLR_P0	RUN_P0	PWM0CON
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Downloaded from Elcodis.com electronic components distributor

Symbol	Function			
P0SEL	PWM0 Waveform Output. This bit enables/disables the PWM0 waveform output to Port 3.4.			
PS2_P0 PS1_P0 PS0_P0	Prescaled Clock Selection			
	PS2_P0	PS1_P0	PS0_P0	Result
	0	0	0	F_{OSC}/1 divide
	0	0	1	F_{OSC}/2 divide
	0	1	0	F_{OSC}/4 divide
	0	1	1	F_{OSC}/8 divide
	1	0	0	F_{OSC}/16 divide
	1	0	1	F_{OSC}/32 divide
	1	1	0	F_{OSC}/64 divide
	1	1	1	F_{OSC}/128 divide
MODE_P0	Counter Mode Selection. 0: 8-bit counter 1: (2+6)-bit counter. 6-bit compare & 2-bit extended-compare			
RL_P0	Reload Mode Selection. 0: 6-bit counter overflow/reload. 1: 8-bit counter overflow/reload.			
CLR_P0	Counter Reset Enable. This bit is cleared by hardware.			
RUN_P0	Counter Start Enable.			

15.36 PWM 1 Control Register (PWM1CON)

Bit No.	7	6	5	4	3	2	1	0	
DDh	P1SEL	PS2_P1	PS1_P1	PS0_P1	MODE_P1	RL_P1	CLR_P1	RUN_P1	PWM1CON
	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	Function			
P1SEL	PWM1 Waveform Output. This bit enables/disables the PWM1 waveform output to Port 1.0.			
PS2_P1 PS1_P1 PS0_P1	Prescaled Clock Selection			
	PS2_P1	PS1_P1	PS0_P1	Result
	0	0	0	F_{OSC}/1 divide

Downloaded from Elcodis.com electronic components distributor

	0	0	1	F_{OSC}/2 divide
	0	1	0	F_{OSC}/4 divide
	0	1	1	F_{OSC}/8 divide
	1	0	0	F_{OSC}/16 divide
	1	0	1	F_{OSC}/32 divide
	1	1	0	F_{OSC}/64 divide
	1	1	1	F_{OSC}/128 divide
MODE_P1	Counter Mode Selection. 0: 8-bit counter 1: (2+6)-bit counter. 6-bit compare & 2-bit extended-compare			
RL_P1	Reload Mode Selection. 0: 6-bit counter overflow/reload. 1: 8-bit counter overflow/reload.			
CLR_P1	Counter Reset Enable. This bit is cleared by hardware.			
RUN_P1	Counter Start Enable.			

15.37 PWM0 Duty Data Register (PWM0D)

Bit No.	7	6	5	4	3	2	1	0	
DEh	PWM0D.7	PWM0D.6	PWM0D.5	PWM0D.4	PWM0D.3	PWM0D.2	PWM0D.1	PWM0D.0	PWM0D
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

PWM0D duty data register, located in DEh, determines the duty of the output value generated by each 8-bit PWM circuit. In 8-bit counter mode, to program the required PWM output, you load the appropriate initialization values into the 8-bit data register (PWM0D), and in (6+2)-bit counter mode, you load the appropriate initialization values into the 6-bit reference data register (PWM0D[5:0]) and the 2-bit extension data register (PWM0D[7:6]).

15.38 PWM1 Duty Data Register (PWM1D)

Bit No.	7	6	5	4	3	2	1	0	
DFh	PWM1D.7	PWM1D.6	PWM1D.5	PWM1D.4	PWM1D.3	PWM1D.2	PWM1D.1	PWM1D.0	PWM1D
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

PWM1D duty data register, located in DFh, determines the duty of the output value generated by each 8-

Downloaded from Elcodis.com electronic components distributor

bit PWM circuit. In 8-bit counter mode, to program the required PWM output, you load the appropriate initialization values into the 8-bit data register (PWM1D), and in (6+2)-bit counter mode, you load the appropriate initialization values into the 6-bit reference data register (PWM1D[5:0]) and the 2-bit extension data register (PWM1D[7:6]).

15.39 Accumulator (ACC/A)

Bit No.	7	6	5	4	3	2	1	0	
E0h	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	ACC/A
	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.

15.40 ADC Clock and Port Control Register (ADCSEL)

Bit No.	7	6	5	4	3	2	1	0	
E2h	ADIV2	ADIV1	ADIV0	-	ADC3	ADC2	ADC1	ADC0	ADCSEL
	R/W(0)	R/W(0)	R/W(0)	-	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Symbol	Function			
ADIV2 ADIV1 ADIV0	ADC Input Clock Divide			
	ADIV2	ADIV1	ADIV0	Result
	0	0	0	2 divide
	0	0	1	4 divide
	0	1	0	8 divide
	0	1	1	16 divide
	1	0	0	16 divide
-	Reserved.			
ADC3	ADC3 Analog Input Selection Enable. 0: Disables the ADC3 analog input to Port 1.3. 1: Enables the ADC3 analog input to Port 1.3.			
ADC2	ADC2 Analog Input Selection Enable. 0: Disables the ADC2 analog input to Port 1.2. 1: Enables the ADC2 analog input to Port 1.2.			

Downloaded from Elcodis.com electronic components distributor

ADC1	ADC1 Analog Input Selection Enable. 0: Disables the ADC1 analog input to Port 1.1. 1: Enables the ADC1 analog input to Port 1.1.
ADC0	ADC0 Analog Input Selection Enable. 0: Disables the ADC0 analog input to Port 0.0. 1: Enables the ADC0 analog input to Port 0.0.

15.41 Alternative Function Selection Register (ALTSEL)

Bit No.	7	6	5	4	3	2	1	0	
E3h	-	-	EAINT0	P0INT1	P0INT0	TX	RX	PWM0	ALTSEL
			R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

This register is only available in 8/10/20 DIP/SOP package. Don't use this register in 40-DIP/44-PLCC/44-LQFP package.

Symbol	Function
-	Reserved.
EAINT0	INT0 Input Selection Enable. 0: Disables the INT0 input to EA pin. 1: Enables the INT0 input to EA pin.
P0INT1	INT1 Input Selection Enable. 0: Disables the INT1 input to Port 0.2. 1: Enables the INT1 input to Port 0.2.
P0INT0	INT0 Input Selection Enable. 0: Disables the INT0 input to Port 0.1. 1: Enables the INT0 input to Port 0.1.
TX	TX Input/Output Selection Enable. 0: Disables the TX input/output to Port 0.1. 1: Enables the TX input/output to Port 0.1.
RX	RX Output Selection Enable. 0: Disables the RX output to Port 0.0. 1: Enables the RX output to Port 0.0.

Downloaded from Elcodis.com electronic components distributor

PWM0	PWM0 Output Selection Enable. 0: Disables the PWM0 output to Port 0.1. 1: Enables the PWM0 output to Port 0.1.

15.42 Port 0 Pull-up Control Register (P0SEL)

Bit No.	7	6	5	4	3	2	1	0	
E4h	P0SEL.7	P0SEL.6	P0SEL.5	P0SEL.4	P0SEL.3	P0SEL.2	P0SEL.1	P0SEL.0	P0SEL
	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 enables/disables the internal pull-up resistors in Port 0.

0: The internal pull-up resistors in Port 0 are ON.

1: The internal pull-up resistors in Port 0 are OFF.

15.43 Port 1 Pull-up Control Register (P1SEL)

Bit No.	7	6	5	4	3	2	1	0	
E5h	P1SEL.7	P1SEL.6	P1SEL.5	P1SEL.4	P1SEL.3	P1SEL.2	P1SEL.1	P1SEL.0	P1SEL
	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 enables/disables the internal pull-up resistors in Port 1.

0: The internal pull-up resistors in Port 1 are ON.

1: The internal pull-up resistors in Port 1 are OFF.

15.44 Port 2 Pull-up Control Register (P2SEL)

Bit No.	7	6	5	4	3	2	1	0	
E6h	P2SEL.7	P2SEL.6	P2SEL.5	P2SEL.4	P2SEL.3	P2SEL.2	P2SEL.1	P2SEL.0	P2SEL
	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 enables/disables the internal pull-up resistors in Port 2.

0: The internal pull-up resistors in Port 2 are ON.

1: The internal pull-up resistors in Port 2 are OFF.

Downloaded from Elcodis.com electronic components distributor

15.45 Port 3 Pull-up Control Register (P3SEL)

Bit No.	7	6	5	4	3	2	1	0	
E7h	P3SEL.7	P3SEL.6	P3SEL.5	P3SEL.4	P3SEL.3	P3SEL.2	P3SEL.1	P3SEL.0	P3SEL
	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 enables/disables the internal pull-up resistors in Port 3.

0: The internal pull-up resistors in Port 3 are ON.

1: The internal pull-up resistors in Port 3 are OFF.

15.46 External Interrupt Enable Register (EIE)

Bit No.	7	6	5	4	3	2	1	0	
E8h	-	-	-	EWDT	EX5	EX4	EX3	EX2	EIE
	-	-	-	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Symbol	Function
-	Reserved.
EWDT	Watchdog timer interrupt Enable. This bit enables/disables the watchdog timer interrupt. 0: Disable the watchdog timer interrupt. 1: Enable the interrupt requests generated by the watchdog timer.
EX5	External Interrupt 5 Enable. This bit enables/disables external interrupt 5. 0: Disable external interrupt 5. 1: Enable the interrupt requests generated by the /INT5 pin.
EX4	External Interrupt 4 Enable. This bit enables/disables external interrupt 4. 0: Disable external interrupt 4. 1: Enable the interrupt requests generated by the INT4 pin.
EX3	External Interrupt 3 Enable. This bit enables/disables external interrupt 3. 0: Disable external interrupt 3. 1: Enable the interrupt requests generated by the /INT3 pin.
EX2	External Interrupt 2 Enable. This bit enables/disables external interrupt 2. 0: Disable external interrupt 2. 1: Enable the interrupt requests generated by the INT2 pin.

Downloaded from Elcodis.com electronic components distributor

15.47 ADC Result Value Register (ADCR)

Bit No.	7	6	5	4	3	2	1	0	
EEh	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1	ADCR
	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

[8:1] of ADC conversion result will be stored into ADCR. LSB of conversion result will be stored into ADCON[0].

15.48 ADC Control Register (ADCON)

Bit No.	7	6	5	4	3	2	1	0	
EFh	AD_EN	AD_REQ	AD_END	ADCF	ACH1	ACH0	-	SAR0	ADCON
	R/W(0)	R/W(0)	R(1)	R/W(0)	R/W(0)	R/W(0)		R/W(0)	

<table>
<tr><th>Symbol</th><th colspan="3">Function</th></tr>
<tr><td>AD_EN</td><td colspan="3">ADC Ready Enable.</td></tr>
<tr><td>AD_REQ</td><td colspan="3">ADC Start Enable. Cleared by H/W when AD_END goes to 1 from 0.</td></tr>
<tr><td>AD_END</td><td colspan="3">Cleared by H/W when ADC conversion start.
Set by H/W when ADC conversion has been finished.
0: Data conversion is now running.
1: ADC is idle state.</td></tr>
<tr><td>ADCF</td><td colspan="3">ADC Interrupt Flag. This bit must be cleared by S/W.</td></tr>
<tr><td rowspan="6">ACH1
ACH0</td><td colspan="3">ADC Channel Selection</td></tr>
<tr><td>ACH1</td><td>ACH0</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>ADC0 (P0.0) input selection</td></tr>
<tr><td>0</td><td>1</td><td>ADC1 (P1.1) input selection</td></tr>
<tr><td>1</td><td>0</td><td>ADC2 (P1.2) input selection</td></tr>
<tr><td>1</td><td>1</td><td>ADC3 (P1.3) input selection</td></tr>
<tr><td>-</td><td colspan="3">Reserved.</td></tr>
<tr><td>SAR0</td><td colspan="3">LSB of ADC result value</td></tr>
</table>

Downloaded from Elcodis.com electronic components distributor

15.49 B Register (B)

Bit No.	7	6	5	4	3	2	1	0	
F0h	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	B
	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.

15.50 Extended Interrupt Priority Register (EIP)

Bit No.	7	6	5	4	3	2	1	0	
F8h	-	-	-	PWDT	PX5	PX4	PX3	PX2	EIP
	-	-	-	R/W(0)	R/W(0)	R/W(0)	R/W(0)	R/W(0)	

Symbol	Function
-	Reserved.
PWDT	Watchdog timer Interrupt Priority. This bit controls the priority of the watchdog timer interrupt. 0: The Watchdog timer interrupt is a low priority interrupt. 1: The Watchdog timer interrupt is a high priority interrupt.
PX5	External Interrupt 5 Priority. This bit controls the priority of the external interrupt 5. 0: The external interrupt 5 is a low priority interrupt. 1: The external interrupt 5 is a high priority interrupt.
PX4	External Interrupt 4 Priority. This bit controls the priority of the external interrupt 4. 0: The external interrupt 4 is a low priority interrupt. 1: The external interrupt 4 is a high priority interrupt.
PX3	External Interrupt 3 Priority. This bit controls the priority of the external interrupt 3. 0: The external interrupt 3 is a low priority interrupt. 1: The external interrupt 3 is a high priority interrupt.
PX2	External Interrupt 2 Priority. This bit controls the priority of the external interrupt 2. 0: The external interrupt 2 is a low priority interrupt. 1: The external interrupt 2 is a high priority interrupt.

Downloaded from Elcodis.com electronic components distributor