MX10E8050I
MX10E8050X
MX10E8050IA

Major Difference

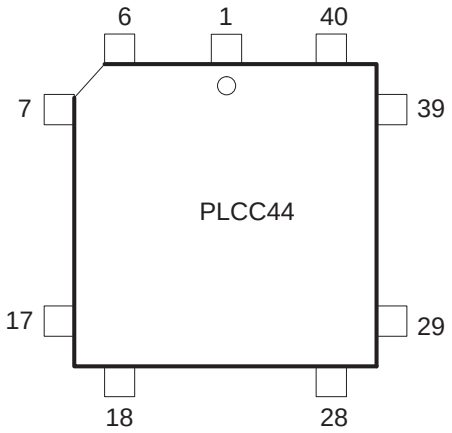
Product	Feature			
	Default Clock mode	ISP	IAP	Package
MX10E8050IPC MX10E8050IQC MX10E8050IUC	x 12	YES	YES	40 Pin PDIP 44 Pin PLCC 44 Pin LQFP
MX10E8050XPC MX10E8050XQC MX10E8050XUC	x 6	NO	NO	40 Pin PDIP 44 Pin PLCC 44 Pin LQFP
MX10E8050IAQC	x 12	Yes	Yes	44 Pin PLCC

This datasheet has been downloaded from http://www.digchip.com at this page

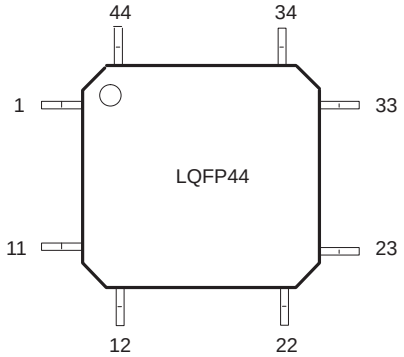
FEATURES

- 80C51 CPU core
- 3.0 ~ 3.6V voltage range
- On-chip Flash program memory with in-system programming (ISP)
- Operating frequency up to 40MHz (12x),20MHz(6x)
- 64K bytes Flash memory for code memory
- 1280 bytes internal data RAM
- Low power consumption
- Code and data memory expandable to 64K Bytes
- Four 8 bit and one 4 bit general purpose I/O ports
- Three standard 16-bit Timers
- In - Application Programming(IAP) capability
- On-chip Watch Dog Timer
- Four channel PWM outputs/4bit general purpose I/O ports (PLCC & LQFP only)
- UART
- 7 interrupt sources with four priority level
- 5 volt tolerant input
- 400kb/s I^2C
- 6x / 12x clock mode

PIN Configurations



Pin	Function	Pin	Function	Pin	Function
1	P4.2/PWM2	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	$\overline{PSEN}$
3	P1.1/T2EX	18	P3.6/$\overline{WR}$	33	ALE
4	P1.2	19	P3.7/$\overline{RD}$	34	P4.1/PWM1
5	P1.3	20	$\overline{XTAL2}$	35	$\overline{EA}$
6	P1.4	21	XTAL1	36	P0.7/AD7
7	P1.5	22	V_{SS}	37	P0.6/AD6
8	P1.6/SCL	23	P4.0/PWM0	38	P0.5/AD5
9	P1.7/SDA	24	P2.0/A8	39	P0.4/AD4
10	RST	25	P2.1/A9	40	P0.3/AD3
11	P3.0/RxD	26	P2.2/A10	41	P0.2/AD2
12	P4.3/PWM3	27	P2.3/A11	42	P0.1/AD1
13	P3.1/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/$\overline{INT0}$	29	P2.5/A13	44	V_{CC}
15	P3.3/$\overline{INT1}$	30	P2.6/A14		



Pin	Function	Pin	Function	Pin	Function
1	P1.5	16	V_{SS}	31	P0.6/AD6
2	P1.6/SCL	17	P4.0/PWM0	32	P0.5/AD5
3	P1.7/SDA	18	P2.0/A8	33	P0.4/AD4
4	RST	19	P2.1/A9	34	P0.3/AD3
5	P3.0/RxD	20	P2.2/A10	35	P0.2/AD2
6	P4.3/PWM3	21	P2.3/A11	36	P0.1/AD1
7	P3.1/TxD	22	P2.4/A12	37	P0.0/AD0
8	P3.2/$\overline{INT0}$	23	P2.5/A13	38	V_{CC}
9	P3.3/$\overline{INT1}$	24	P2.6/A14	39	P4.2/PWM2
10	P3.4/T0	25	P2.7/A15	40	P1.0/T2
11	P3.5/T1	26	$\overline{PSEN}$	41	P1.1/T2EX
12	P3.6/$\overline{WR}$	27	ALE	42	P1.2
13	P3.7/$\overline{RD}$	28	P4.1/PWM1	43	P1.3
14	$\overline{XTAL2}$	29	$\overline{EA}$	44	P1.4
15	XTAL1	30	P0.7/AD7		

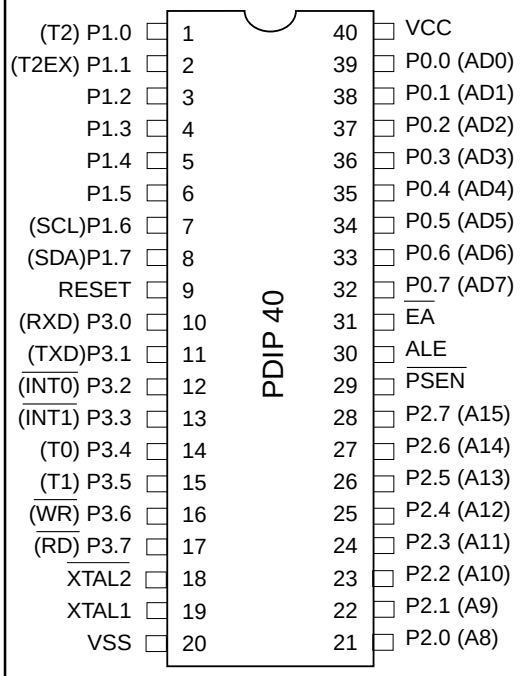


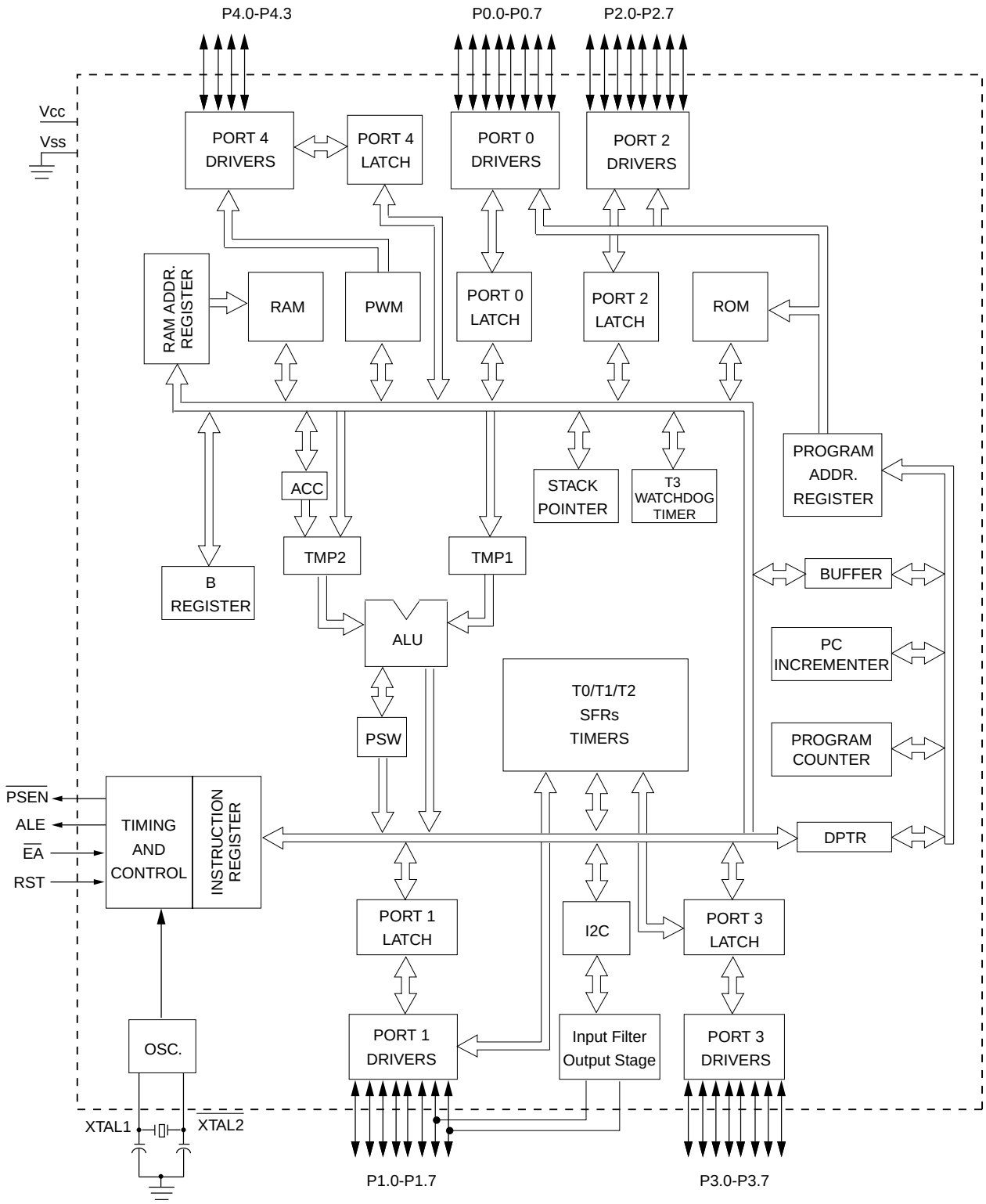
Table. 1 Pin Description

	Package Type	PDIP	PLCC	LQFP	
I/O	SYMBOL	PIN	PIN	PIN	DESCRIPTION
I/O	P0.0-P0.7	39-32	43-36	37-30	Port:8-bit open drain bidirectional I/O Port
I/O	P2.0-P2.7	21-28	24-31	18-25	Port: 8-bit quasi-bidirectional I/O Port with internal pull-up
I/O	P1.0-P1.7	1-8	2-9	40-44,1-3	Port: 8-bit quasi-bidirectional I/O Port with internal pull-up , except P1.6 and P1.7
I/O	P3.0-P3.7	10-17	11,13-19	5,7-13	Port: 8-bit quasi-bidirectional I/O Port with internal pull-up
I/O	P4.0~P4.3/ PWM0~PWM3	NA	23,34,1,12	17,28,39,6	4bit Quasi-bidirectional I/O port or PWM
I	RESET	9	10	4	reset input
I	VCC	40	44	38	Positive power supply
I	VSS	20	22	16	Ground
I	XTAL1	19	21	15	XTAL connection input
O	$\overline{XTAL2}$	18	20	14	XTAL connection output
O	$\overline{PSEN}$	29	32	26	Program store enable output
O	ALE	30	33	27	Address latch enable output
I	$\overline{EA}$	31	35	29	External access input

Mnemonic	Pin Number			Type	Name and Function
	PDIP	PLCC	LQFP		
V_{ss}	20	22	16	I	Ground: 0 volt reference
V_{cc}	40	44	38	I	Power Supply: This is the power supply voltage for normal, idle and power-down operation
P0.0 ~ 0.7	39-32	43-36	37-30	I/O	Port 0: Port 0 is an open drain, bi-directional I/O port. Port 0 pins have 1s written to them float and can be used as high impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accessed to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0~1.7	1-8	2-9	40-44 1-3	I/O	Port1: Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current because of the internal pull-ups. Note that P1.6 and P1.7 are open drain pins for I^2C function. Alternate functions for port 1 include:
	1	2	40	I/O	T2(P1.0): Timer/Counter 2 external count input/clock out
	2	3	41	I	T2EX(P1.1): Timer/Counter 2 Reload / Capture / Direction control
	3	4	42	I	SDA (P1.7): Data line for I^2C
	4	5	43	I/O	SCL (P1.6): Clock line for I^2C
	5	6	44	I/O	
	6	7	1	I/O	
	7	8	2	I/O	
	8	9	3	I/O	
P2.0~2.7	21-28	24-31	18-25	I/O	Port 2 : Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current because of the internal pull-ups. Port 2 emits the high ordered address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory using 8-bit addresses (MOVX@R_I), port 2 emits the contents of P2 special `function register.
P3.0~3.7	10-17	11, 13-19	5, 7-13	I/O	Port 3: Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high with the internal pull-ups and can be used as inputs. As

					inputs, Port 3 pins that are externally pulled low will source current because of the internal pull-ups. Port 3 also serves the special features of MX10E8050I family, as listed below:
	10	11	5	I	RxD (P3.0) : Serial input port
	11	13	7	O	TxD (P3.1) : Serial output port
	12	14	8	I	INT0 (P3.2) : External interrupt 0
	13	15	9	I	INT1 (P3.3) : External interrupt 1
	14	16	10	I	T0 (P3.4) : Timer 0 external input
	15	17	11	I	T1 (P3.5) : Timer 1 external input
	16	18	12	O	WR (P3.6) : External data memory write strobe
	17	19	14	O	RD (P3.7) : External data memory read strobe
P4.0~P4.3				I/O	Port 4: Port 4 is an 4-bit bi-directional I/O port with internal pull-ups. Port 4 pins that have 1s written to them are pulled high with the internal pull-ups and can be used as inputs. As inputs, Port 4 pins that are externally pulled low will source current because of the internal pull-ups. Port 4 also serves the special features of MX10E8050I family, as listed below:
P4.0		23	17	I	PWM0 (P4.0) : PWM module output 0
P4.1		34	28	I	PWM1 (P4.1) : PWM module output 1
P4.2		1	39	I	PWM2 (P4.2) : PWM module output 2
P4.3		12	6	I	PWM3 (P4.3) : PWM module output 3
RST	9	10	4	I	Reset : A high on this pin for eight machine cycles while the oscillator is running, reset the devices. An internal diffused resistor to V_{ss} permits a power-on reset using only an external capacitor to V_{cc}
ALE	30	33	27	O	Address Latch Enable: Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at constant rate of 1/6 the oscillator frequency in 12x clock mode. 1/3 the oscillator frequency in 6x clock mode, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
$\overline{PSEN}$	29	32	26	O	Program Strobe Enable: The read strobe to external program memory. When executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle., except the two $\overline{PSEN}$ activation are skipped during each access to external data memory. $\overline{PSEN}$ is not activated during fetch from internal program memory.
$\overline{EA}$	31	35	15	I	External Access Enable/ Programming Supply Voltage: $\overline{EA}$ must be external held low to enable the device to fetch code from external program memory locations 0000H and FFFFH for 64 K devices.
XTAL 1	19	21	15	I	Crystal 1: Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL 2	18	20	14	O	Crystal 2: Output from the inverting oscillator amplifier.

BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

General

The MX10E8050I Serial is a stand-alone high-performance and low power microcontroller designed for use in many applications which need code programmability.

The Flash EPROM offers customers to program the device themselves. This feature increases the flexibility in many applications, not only in development stage, but also in mass production stage.

In addition to the 80C51 standard functions, the MX10E8050I Serial provides a number of dedicated hardware functions. MX10E8050I Serial is a control-oriented CPU with on-chip program and data memory. It can execute program with internal memory up to 64k bytes. MX10E8050I Serial has two software selectable modes of reduced activity for power reduction Idle, and Power-down. The idle mode freezes the CPU while allowing the RAM, Timers, serial ports, interrupt system and other peripherals to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative. Power-down mode can be terminated by an external reset ,and in addition , by either of the two external interrupts can be terminated as the power down mode does.

MEMORY ORGANIZATION

The Central Processing Unit (CPU) manipulates operands in three memory spaces; these are the 256 bytes internal data memory (RAM), 1k byte auxiliary data memory (AUX-RAM) and 64k byte internal MTP program memory (FLASH ROM).

Program Memory

The program memory address space of the MX10E8050I Serial comprises an internal and an external memory space. The MX10E8050I Serial has 64k byte of program memory on-chip.

Program Protection

If the user choose to set security lock in MTP memory, the program content is protected from reading out of chip.

Internal Data Memory

The internal data memory is divided into three physically separated parts: 256 byte of RAM, 1k bytes of AUX-RAM, and 128 bytes special function register area (SFR). These parts can be addressed as follows (see Fig.1 and Table. 2)

- RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- RAM 128 to 255 can only be addressed indirectly . Address pointers are R0 and R1 of the selected register bank.
- AUX-RAM 0 to 1023 is indirectly addressable as the external data memory locations 0 to 1023 by the MOVX instructions. Address pointers are R0 and R1 of the selected register bank and DPTR. SFRs can only be addressed directly in the address range from 128 to 255.

Table. 2 Internal data memory access

LOCATION	ADDRESSED
RAM 0 to 127	DIRECT and INDIRECT
RAM 128 to 255	INDIRECT only
AUX-RAM 0 to 1023	INDIRECT only with MOVX
Special Function Register (SFR) 128 to 255	DIRECT only

Fig. 1 shows the internal memory address space. Table 3 shows the Special Function Register (SFR) memory map. Location 0 to 31 at the lower RAM area can be devided into four 8-bit register banks. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations.

The stack can be located anywhere in the internal 256 byte RAM. The stack depth is only limited by the available internal RAM space of 256 bytes. All registers except the Program Counter and the four 8-byte register banks reside in the SFR address space.

Five methods to access memory space are as floww :
- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register plus Index-Register-Indirect.

The first three methods can be used for addressing destination operands. Most instructions have a 'destination / source' field that specifies the data type, addressing methods and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addresses is as follows:
- Register in one of the four 8-byte register banks through Direct or Register-Indirect addressing.
- 256 bytes of internal RAM through Direct or Register-Indirect addressing. Bytes 0-127 of internal RAM may be only be addressed indirectly as data RAM.
- SFR through direct addressing at address location 128-255.

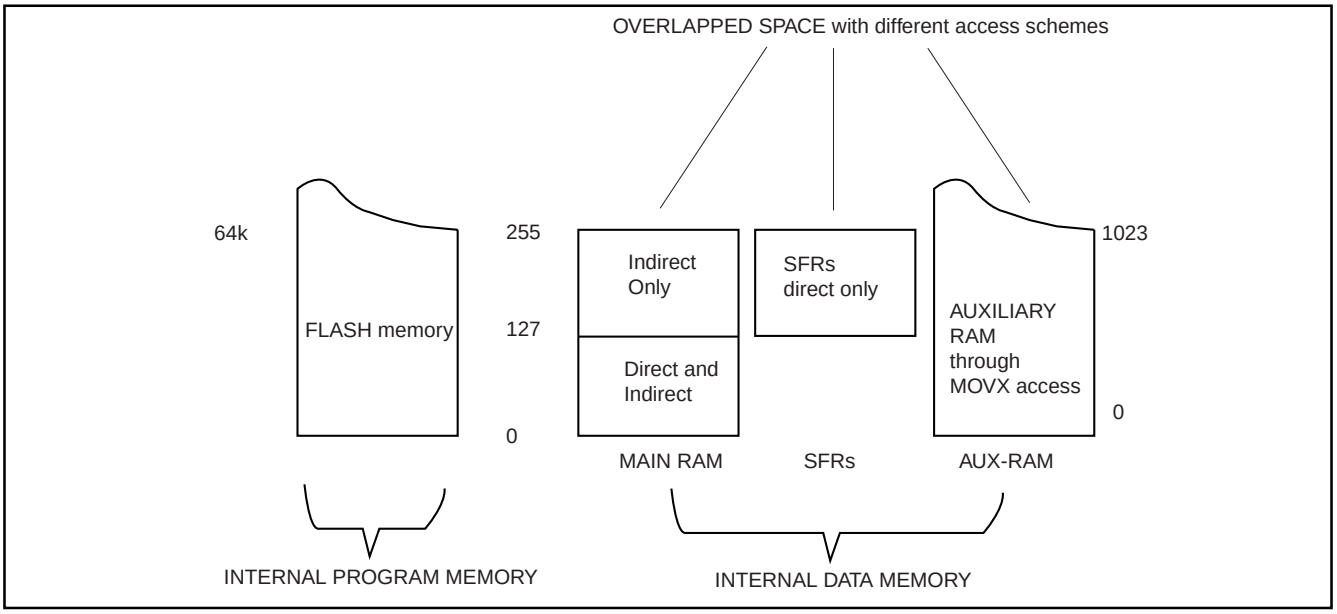


Fig.1 Internal program and data memory address space

Table. 3 SFR Register Map

HIGH NIBBLE OF SFR ADDRESS

LOW	8	9	A	B	C	D	E	F
0	P0% 11111111	P1% 11111111	P2% 11111111	P3% 11111111	P4% 11111111	PSW% 00000000	ACC% 00000000	B% 00000000
1	SP 00000111							PWMC 10000000
2	DPL 00000000		AUXR1 00000000					
3	DPH 00000000							PWMP3 00000000
4							FMCON 00000001	PWM2 00000000
5							FMDATA 00000000	PWM3 00000000
6								PWMP2 00000000
7	PCON 00000000			IPH 00000000				
8	TCON% 00000000	SCON% 00000000	IE% 00000000	IP% 00000000	T2CON% 00000000	S1CON 00000000		PDCON 00000000
9	TMOD 00000000	SBUF XXXXXXXX	SADDR 00000000	SADEN 00000000	T2MOD 11111110	S1STA 11111000		
A	TL0 00000000				RCAP2L 00000000	S1DAT 00000000		
B	TL1 00000000				RCAP2H 00000000	S1ADR 00000000	EBTCON XXXXXX1X	PWMP1 00000000
C	TH0 00000000				TL2 00000000			PWM0 00000000
D	TH1 00000000				TH2 00000000			PWM1 00000000
E	AUXR 00000000							PWMP0 00000000
F								T3 11111111

NOTES :

% = Bit addressable register

x = Undefined

Special Function Registers

Symbol	Description	Direct Address	Bit Address, Symbol, or Alternative Port Function MSB							LSB	Reset Function
ACC	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR	Auxiliary	8EH	-	-	-	-	-	-	EXTRAM	AO	00000000B
AUXR1	Auxiliary1	A2H	-	-	ENBOOT	-	-	0	-	DPS	00000000B
B	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR	Data pointer(2-byte)										
	Data pointer high										
DPH	Data pointer low	83H									00H
DPL		82H									00H
EBTCON	Enable T3	EBH							EB		xxxxxx1xB
FMCON	Flash control	E4H	PPARAM	PALE	PCEB	POEB	PWEB	-	-	PREADYB	00000001B
FMDATA	Flash data	E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	00000000B
			AF	AE	AD	AC	AB	AA	A9	A8	
IE	Interrupt Enable	A8H	EA	ET2	ES1	ES	ET1	EX1	ET0	EX0	00000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP	Interrupt priority	B8H	-	PT2	PS1	PS	PT1	PX1	PT0	PX0	x0000000B
			B7	B6	B5	B4	B3	B2	B1	B0	
IPH	Interrupt priority high	B7H	-	PT2H	PS1H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000B
			87	86	85	84	83	82	81	80	
P0	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1	Port 1	90H	P17	P16	P15	P14	P13	P12	P11	P10	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2	Port 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
							C3	C2	C1	C0	
P4	Port4	C0H	-	-	-	-	PWM3	PWM2	PWM1	PWM0	FH
PCON	Power Control	87H	SMOD1	SMOD0	-	WLE	GF1	GF2	PD	IDL	00xx0000B
PDCON	ROM enable code	F8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	00000000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	-	P	000000x0B
PWMC	PWM control	F1H	PWMD	DSCB	PWM3E	PWM2E		DSCA	PWM1E	PWM0E	1000x000B
PWMP0	Prescaler vector 0	FEH	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	00000000B
			0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0	
PWMP1	Prescaler vector 1	FBH	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	00000000B
			1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0	
PWMP2	Prescaler vector 2	F6H	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	00000000B
			2.7	2.6	2.5	2.4	2.3	2.2	2.1	2.0	
PWMP3	Prescaler vector 3	F2H	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	PWMP	00000000B

			3.7	3.6	3.5	3.4	3.3	3.2	3.1	3.0	
PWM0	PWM0 ratio	FCH	PWM	PWM	PWM	PWM	PWM	PWM	PWM	PWM	00000000B
			0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0	
PWM1	PWM1 ratio	FDH	PWM	PWM	PWM	PWM	PWM	PWM	PWM	PWM	00000000B
			1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0	
PWM2	PWM2 ratio	F4H	PWM	PWM	PWM	PWM	PWM	PWM	PWM	PWM	00000000B
			2.7	2.6	2.5	2.4	2.3	2.2	2.1	2.0	
PWM3	PWM3 ratio	F5H	PWM	PWM	PWM	PWM	PWM	PWM	PWM	PWM	00000000B
			3.7	3.6	3.5	3.4	3.3	3.2	3.1	3.0	
RACAP2H	Timer 2 Capture High	CBH									00H
RACAP2L	Timer 2 Capture Low	CAH									00H
SADDR	Slave Address	A9H									00H
SADEN	Slave address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H									07H
			DF	DE	DD	DC	DB	DA	D9	D8	
S1CON	I^2C Control	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00H
S1STA	I^2C Status	D9H	S1STA.7	S1STA.6	S1STA.5	S1STA.4	S1STA.3				00H
S1DAT	I^2C data	DAH	S1DAT.7	S1DAT.6	S1DAT.5	S1DAT.4	S1DAT.3	S1DAT.2	S1DAT.1	S1DAT.0	00H
S1ADR	I^2C address	DBH	S1ADR.7	S1ADR.6	S1ADR.5	S1ADR.4	S1ADR.3	S1ADR.2	S1ADR.1	GC	00H
TCON	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL	00H
T2MOD	Timer 2 Mode Control	C9H	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00B
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TH2	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
T3	Timer 3	FFH									FFH

AUXR (8EH)

						EXTRAM	A0

- EXTRAM : External RAM Select Switch. Set 1 to select (MOVX) the external RAM when the address is larger than 256.
  Default is 0 to switch (MOVX) to external RAM only when the address is larger than 1k.
- AO : Turn off ALE output in internal execution mode.

Watchdog Timer/WDT/T3 (FFH)

- WDT consists of an 11-bit prescaler and an 8-bit timer formed by SFR T3.

EBTCON (EBH)

						/EW	

- /EW: After reset, /EW bit is set, and WDT is disable.

POWER CONTROL Register/PCON (87H)

SMOD1	SMOD0	X	WLE	GF1	GF0	PD	IDL

- SMOD1: Double baud rate bit for UART.
- SMOD0: Frame error detection bit.
- WLE: Watchdog load enable. This flag must be set prior to loading WDT and is cleared when WDT is loaded.
- GF1/GF0: general-purpose flag bit.
- PD: Power-down bit. Setting it activates power-down mode.
- IDL: Idle mode bit. Setting it activates idle mode.
- The CPU & Peripheral status during 2 power saving mode:

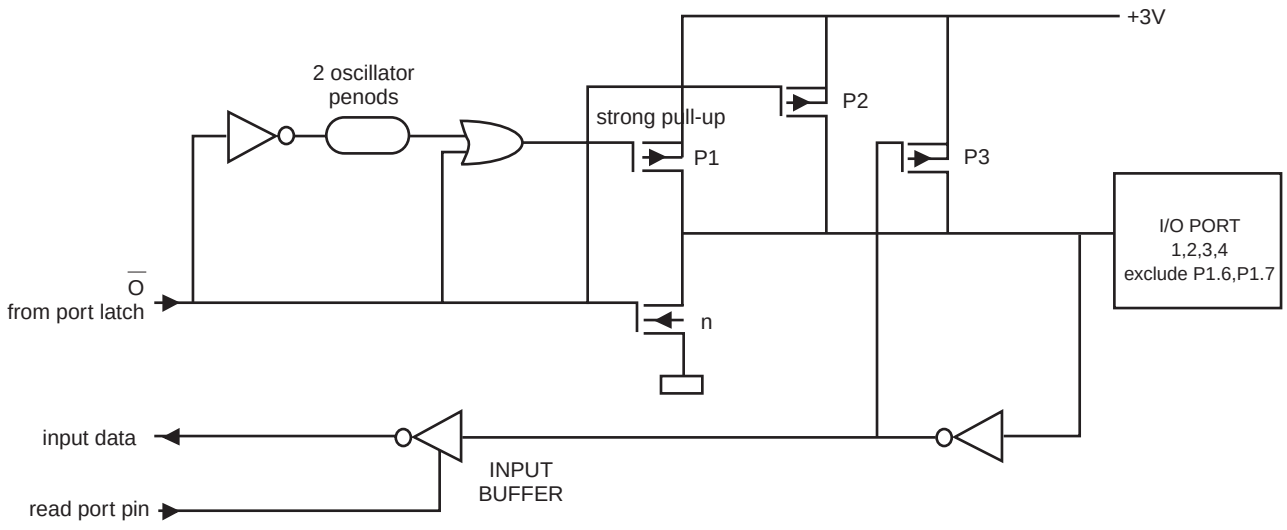
	Idle mode	Power-down mode
CPU	OFF	OFF
Int,Timer.	ON	OFF
Oscillator ckt	ON	OFF

I/O facilities

MX10E8050I has one 8 bits port, port 0, which is open drain, three 8 bits ports, port1/2/3 and a four-bits port port 4 . They are quasi bi-directional ports except P1.6 and P1.7. These five ports are fully compatible to standard 80C51's port 0/1/2/3/4.

- Port3: pins can be configured individually to provide: external interrupt inputs (external interrupt 0/1); external inputs for Timer/ counter 0 and Timer /counter1, and UART receive / transmit.
- Port 1.6, Port 1.7 : pins are used to be I^2C clock and data I/O, which are open drain

Port pins which are not used for alternate functions may be used as normal bidirectional I/O pins. The generation or use of a Port 1 or Port 3 pin as an alternate function is carried out automatically by writing the associated SFR bit with proper value.



I/O buffers in the MX10E8050I (Ports 1,2,3,4)

Timer/Counter

MX10E8050I Serial Timer/Counter 0 and 1 are fully compatible to standard 80C51's.

The MX10E8050I Serial contains two 16-bit Timer/counters, Timer 0 and Timer 1. Timer 0 and Timer 1 may be programmed to carry out the following functions:

- measure time intervals and pulse durations
- count events
- generate interrupt requests.

Timer 0 and Timer 1
Timers 0 and 1 each have a control bit in TMOD SFR that selects the Timer or counter function of the corresponding Timer. In the Timer function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the counter function, the register is incremented in response to a HIGH-to-LOW transition at the corresponding samples, when the transition shows a HIGH in one cycle and a LOW in the next cycle, the counter is incremented. Thus, it takes two machine cycles (24 oscillator periods) to recognize a HIGH-to-LOW transition. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

Timer 0 and Timer 1 can be programmed independently to operate in one of four modes (refer to table 5) :
- Mode 0 : 8-bit Timer/counter with devided-by-32 prescaler
- Mode 1 : 16-bit Timer/counter
- Mode 2 : 8-bit Timer/counter with automatic reload
- Mode 3 : Timer 0 :one 8-bit Timer/counter and one 8-bits Timer. Timer 1 :stopped.

When Timer 0 is in Mode 3, Timer 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag and generate an interrupt. However, the overflow from Timer 1 can be used to pulse the Serial Port transmission-rgate generator. With a 16 MHz crystal, the counting frequency of these Timer/counters is as follows:
- in the Timer function, the Timer is incremented at a frequency of 1.33 MHz (oscillator frequency divided by 12).
- in the counter function, the frequency handling range for external inputs is 0 Hz to 0.66 MHz (oscillator frequency divided by 24).

Both internal and external inputs can be gated to the Timer by a second external source for directly measuring pulse duration.
The Timers are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all logic 1's to all logic 0's (respectively, the automatic reload value), with the exception of Mode 3 as previously described.

TMOD : TIMER/COUNTER MODE CONTROL REGISTER
This register is located at address 89H.

Table. 4 TMOD SFR (89H)

7	6	5	4	3	2	1	0
GATE	C/ T	M1	M0	GATE	C/ T	M1	M0
(MSB)							(LSB)
TIMER 1				TIMER 0			

keep the above table with the following table

Table. 5 Description of TMOD bits

MNEMONIC	POSITION	FUNCTION
TIMER 1		
GATE	TMOD.7	Timer 1 gating control : when set, Timer/counter '1' is enabled only while 'Int1' pin is high and 'tr1' control bit is set. when cleared, Timer/counter '1' is enabled whenever 'tr1' control bit is set.
C/T	TMOD.6	Timer or counter selector: cleared for Timer operation (input from internal system clock). set for counter operation (input from 'T1' input pin).
M1	TMOD.5	Operation mode: see table 6.
M0	TMOD.4	Operation mode: see table 6.
TIMER 0		
GATE	TMOD.3	Timer 0 gating control: when set, Timer/Counter '0' is enabled only while 'Int0' pin is high and 'tr0' control bit is set. when cleared, Timer/counter '0' is enabled whenever 'tr0' control bit is set.
C/T	TMOD.2	Timer or counter selector: cleared for Timer operation (input from internal system clock). set for counter operation (input from 'T0' input pin).
M1	TMOD.1	Operation mode: see table 6.
M0	TMOD.0	Operation mode: see table 6.

Table. 6 TMOD M1 and M0 operating modes

M1	M0	FUNCTION
0	0	8-bit Timer/counter : 'THx' with 5-bit prescaler.
0	1	16-bit Timer/counter : 'THx' and 'TLx' are cascaded, there is no prescaler.
1	0	8-bit autoload Timer/counter : 'THx' holds a value which is to be reloaded into 'TLx' each time it overflows.
1	1	Timer 0: TL0 is an 8-bit Timer/counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer controlled by Timer 1 control bits.
1	1	Timer 1 : Timer/counter 1 stopped.

TCON : TIMER/COUNTER CONTROL REGISTER
This register is located at address 88H.
Notes :

Symbol	Description	Direct Address	Bit Address, Symbol, or Alternative Port Function MSB							LSB	Reset Function
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

Table. 7 TCON SFR (88H)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
(MSB)							(LSB)

keep the above table with the following table

Table. 8 Description of TCON bits

MNEMONIC	POSITION	FUNCTION
TF1	TCON.7	Timer 1 overflow flag : set by hardware on Timer/Counter overflow. Cleared when interrupt is processed.
TR1	TCON.6	Timer 1 control bit : set/cleared by software to turn Timer/counter ON/OFF.
TF0	TCON.5	Timer 0 overflow flag: set by hardware on Timer/Counter overflow. Cleared when interrupt is processed.
TR0	TCON.4	Timer 0 control bit : set/cleared by software to turn Timer/counter ON/OFF.
IE1	TCON.3	Interrupt 1 edge flag: set by hardware when external interrupt is detected. Cleared when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit : set/cleared by software to specify falling edge/LOW level triggered external interrupt.
IE0	TOCN.1	Interrupt 0 edge flag: set by hardware when external interrupt is detected. Cleared when interrupt is processed.
IT0	TOCN.0	Interrupt 0 type control bit: set/cleared by software tospecify falling edge/LOW level triggered external interrupt.

TIMER 2 OPERATION

Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate as either an event timer or an event counter, as selected by C/T2* in the special function register T2CON (see Figure 2). Timer 2 has three operating modes: Capture, Auto-reload (up or down counting), and Baud Rate Generator, which are selected by bits in the T2CON as shown in Table 9.

Capture Mode

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2=0, then timer 2 is a 16-bit timer or counter (as selected by C/T2* in T2CON) which, upon overflowing sets bit TF2, the timer 2 overflow bit. This bit can be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IE register). If EXEN2= 1, Timer 2 operates as described above, but with the added feature that a 1- to -0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2 like TF2 can generate an interrupt (which vectors to the same location as Timer 2 overflow interrupt. The Timer 2 interrupt service routine can interrogate TF2 and EXF2 to determine which event caused the interrupt). The capture mode is illustrated in Figure B (There is no reload value for TL2 and TH2 in this mode. Even when a capture event occurs from T2EX, the counter keeps on counting T2EX pin transitions or osc/6 pulses (osc/12 in 12 clock mode).).

Auto-Reload Mode (Up or Down Counter)

In the 16-bit auto-reload mode, Timer 2 can be configured (as either a timer or counter [C/T2* in T2CON]) then programmed to count up or down. The counting direction is determined by bit DCEN (Down Counter Enable) which is located in the T2MOD register (see Figure 4). When reset is applied the DCEN=0 which means Timer 2 will default to counting up. If DCEN bit is set, Timer 2 can count up or down depending on the value of the T2EX pin.

Figure 5 shows Timer 2 which will count up automatically since DCEN=0. In this mode there are two options selected by bit EXEN2 in T2CON register. If EXEN2=0, then Timer 2 counts up to 0FFFFH and sets the TF2 (Overflow Flag) bit upon overflow. This causes the Timer 2 registers to be reloaded with the 16-bit value in RCAP2L and RCAP2H. The values in RCAP2L and RCAP2H are preset by software means.

If EXEN2=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at input T2EX. This transition also sets the EXF2 bit. The Timer 2 interrupt, if enabled, can be generated when either TF2 or EXF2 are 1.

In Figure 6 DCEN=1 which enables Timer 2 to count up or down. This mode allows pin T2EX to control the direction of count. When a logic 1 is applied at pin T2EX Timer 2 will count up. Timer 2 will overflow at 0FFFFH and set the TF2 flag, which can then generate an interrupt, if the interrupt is enabled. This timer overflow also causes the 16-bit value in RCAP2L and RCAP2H to be reloaded into the timer registers TL2 and TH2.

When a logic 0 is applied at pin T2EX this causes Timer 2 to count down. The timer will underflow when TL2 and TH2 become equal to the value stored in RCAP2L and RCAP2H. Timer 2 underflow sets the TF2 flag and causes 0FFFFH to be reloaded into the timer registers TL2 and TH2.

The external flag EXF2 toggles when Timer 2 underflows or overflows. This EXF2 bit can be used as a 17th bit of resolution if needed. The EXF2 flag does not generate an interrupt in this mode of operation.

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/6 in 6 clock mode or OSC/12 in 12 clock mode) 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Figure 2. Timer/Counter 2 (T2CON) Control Register

Table 9 : Timer 2 Operation Modes

RCLK + TCLK	CP / $\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud rate generator
X	X	0	(off)

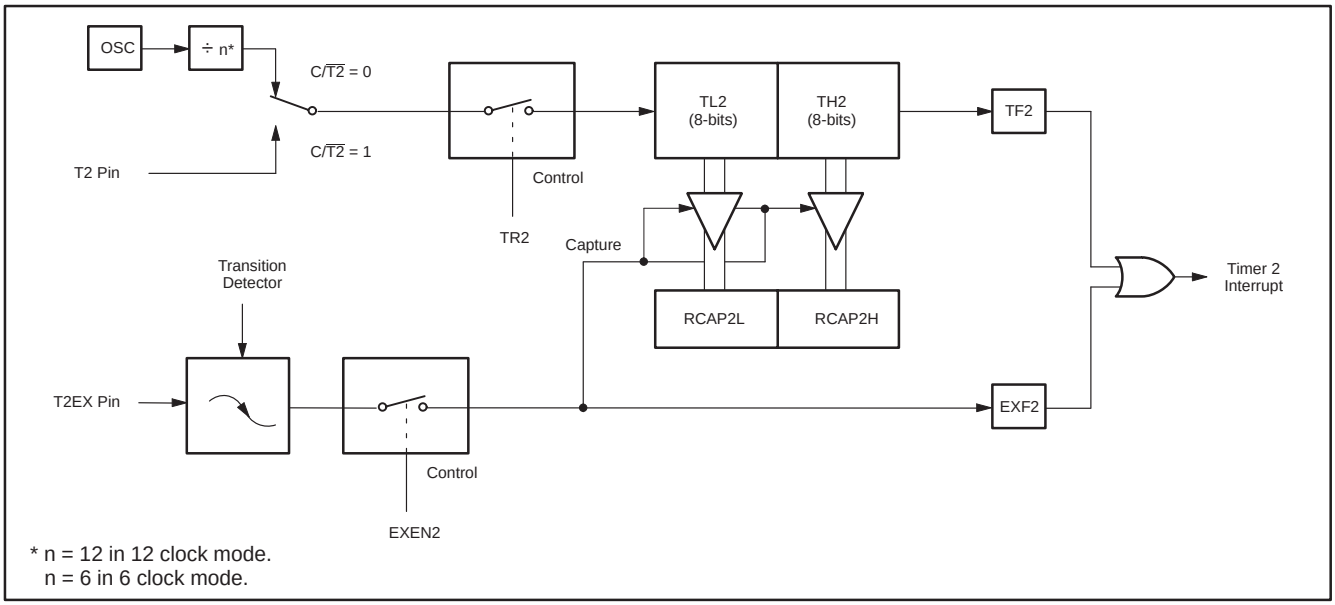


Figure 3 : Timer 2 in Capture Mode

T2MOD Address = 0C9H Reset Value = XXXX XX00B

Not Bit Addressable

–	–	–	–	–	–	T2OE	DCEN
7	6	5	4	3	2	1	0

Symbol	Function
-	Not implemented, reserved for future use.*
T2OE	Timer 2 Output Enable bit.
DCEN	Down Count Enable bit. When set, this allows Timer 2 to be configured as an up/down counter.

* User software should not write 1s to reserved bits. These bits may be used in future 8051 family 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. The value read from a reserved bit is indeterminate.

Figure 4 : Timer 2 Mode (T2MOD) Control Register

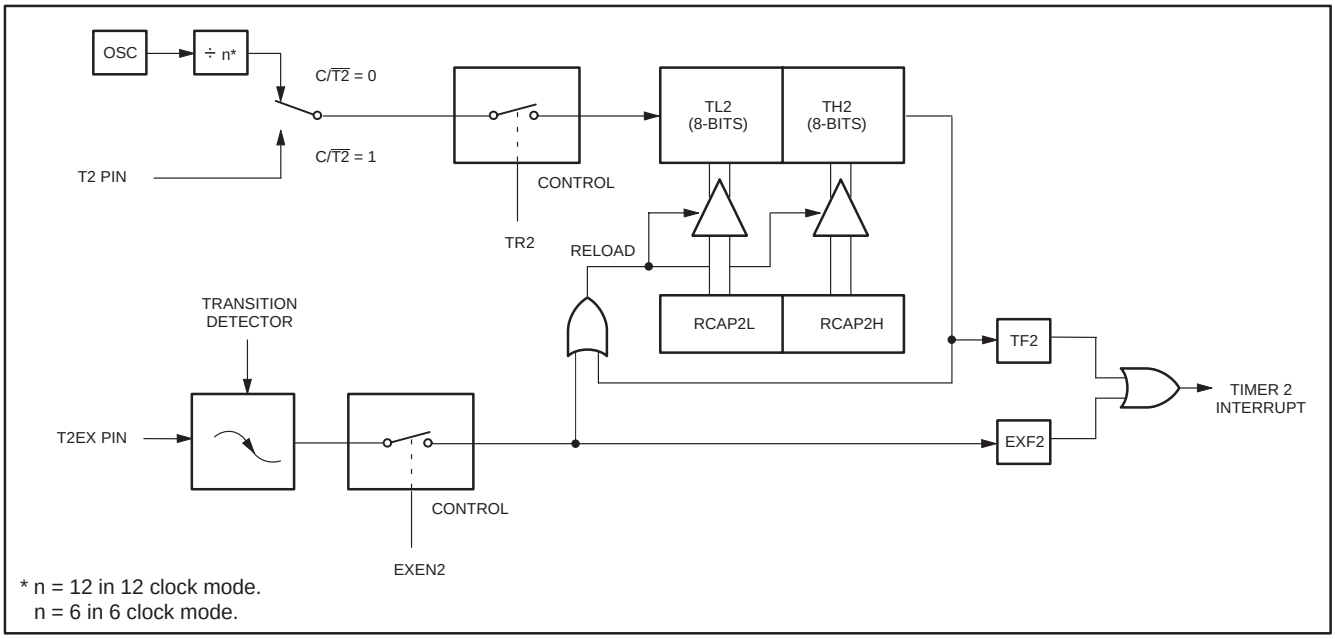


Figure 5 : Timer 2 in Auto-Reload Mode (DCEN = 0)

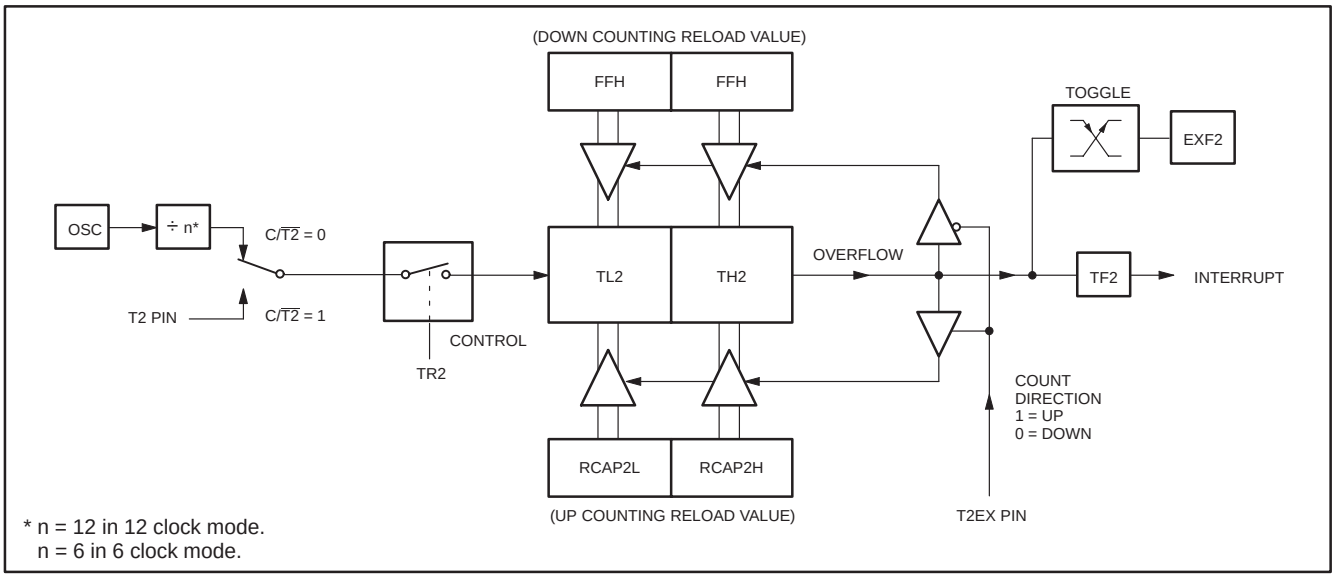


Figure 6 : Timer 2 Auto-Reload Mode (DCEN = 1)

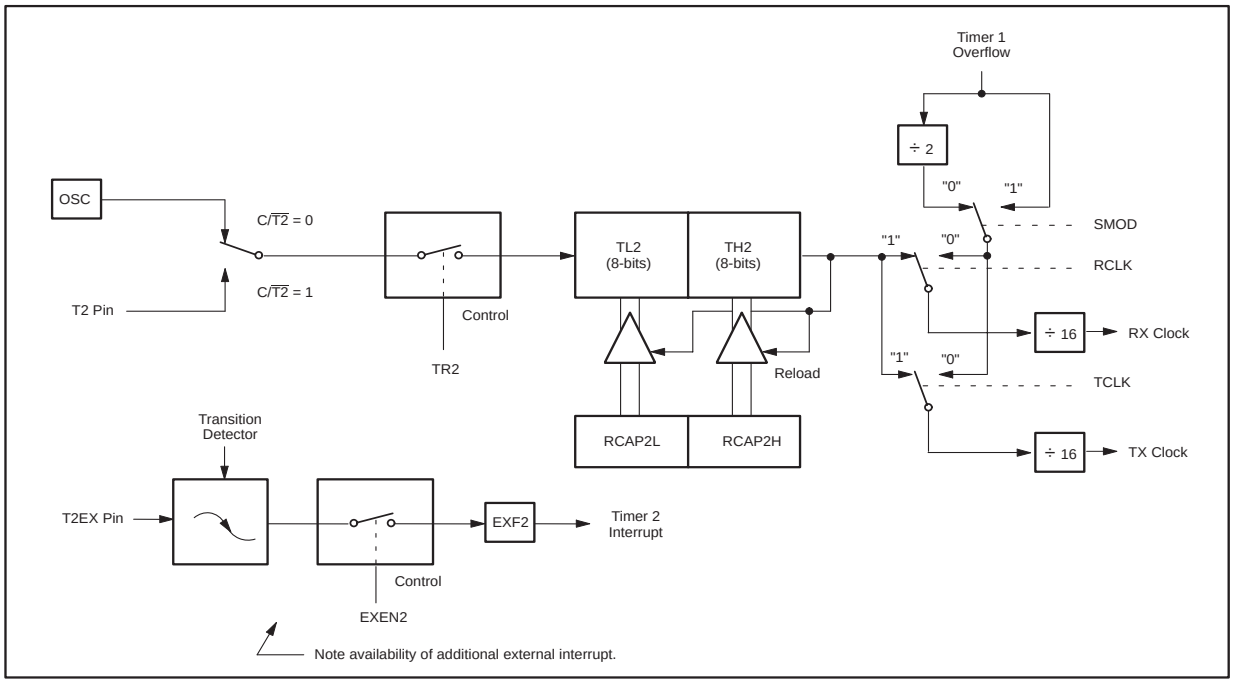


Figure 7. Timer 2 in Baud Rate Generator Mode

Table 10 : Timer 2 Generated Commonly Used Baud Rates

Baud Rate	Osc Freq	Timer 2	
12 clock mode		RCAP2H	RCAP2L
375 k	12 MHz	FF	FF
9.6 k	12 MHz	FF	D9
2.8 k	12 MHz	FF	B2
2.4 k	12 MHz	FF	64
1.2 k	12 MHz	FE	C8
300	12 MHz	FB	1E
110	12 MHz	F2	AF
300	6 MHz	FD	8F
110	6 MHz	F9	57

Baud Rate Generator Mode

Bits TCLK and / or RCLK in T2CON (Table 10) allow the serial port transmit and receive baud rates to be derived from either Timer 1 or Timer 2. When TCLK = 0, Timer 1 is used as the serial port transmit baud rate generator. When TCLK = 1, Timer 2 is used as the serial port transmit baud rate generator. RCLK has the same effect for the serial port receive baud rate. With there two bits, the serial port can have different receive and transmit baud rates - one generated by Timer1, the other by Timer2.

Figure 7 shows the Timer2 in baud rate generation mode. The baud rate generation mode is like the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in modes 1 and 3 are determined by Timer 2's overflow rate given below :

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either "timer" or "counter" operation. In many applications, it is configured for "timer" operation (C/T 2* = 0). Timer operation is different for Timer 2 when it is being used as a baud rate generator.
Usually, as a timer it would increment every machine cycle (i.e., 1/6 the oscillator frequency in 6 clock mode, 1/12 the oscillator frequency in 12 clock mode). As a baud rate generator, it increments at the oscillator frequency in 6 clock mode (OSC/2 in 12 clock mode).
Thus the baud rate formula is as follows :

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Oscillator Frequency}}{[\,n^* \times [65536 - (RCAP2H, RCAP2L)]]}$$

*n = 32 in 12 clock mode or 16 in 6 clock mode

Where : (RCAP2h, RCAP2L) = The content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

The Timer 2 as a baud rate generator mode shown in Figure7, is valid only if RCLK and / or TCLK = 1in T2CON register. Note that a rollover in TH2 does not set TF2, and Will not generate an interrupt. Thus, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baudrate generator mode. Also if the EXEN2 (T2 external enable flag) is set, a 1-to-0 transition in T2EX (Timer / counter 2 trigger input) will set EXF2 (T2 external flag) but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Therefore when Timer 2 is in use as a baud rate generator, T2EX can be used as an additional external interrupt, if needed.

When Timer 2 is in the baud rate generator mode, one should not try to read or write TH2 and TL2. As a baud rate generator, Timer 2 is accurate. The RCAP2 registers may be read, but should not be written to, because a write might overlap a reload and cause write and / or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Table 10 shows commonly used baud rates and how they can be obtained from Timer 2.

Summary Of Baud Rate Equations

Timer 2 is in baud rate generating mode. If Timer 2 is being clocked through pin T2(P1.0) the baud rate is :

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

If Timer 2 is being clocked internally, The baud rate is :

$$\text{Baud Rate} = \frac{f_{osc}}{[\,n^* \times [65536 - (RCAP2H, RCAP2L)]]}$$

*n = 32 in 12 clock mode or 16 in 6 clock mode

Where f_{osc} = Oscillator Frequency

To obtain the reload value for RCAP2H and RCAP2L, the above equation can be rewritten as :

$$RCAP2H, RCAP2L = 65536 - \left(\frac{f_{osc}}{n^* \times \text{Baud Rate}}\right)$$

Timer / Counter 2 Set-up
Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the timer on. see Table 11 for set-up of Timer 2 as a timer. Also see Table 12 for set-up of Timer 2 as a counter.

Table 11 : Timer 2 as a Timer

MODE	T2CON	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
Baud rate generator receive and transmit same baud rate	34H	36H
Receive only	24H	26H
Transmit only	14H	16H

Table 12 : Timer 2 as a Counter

MODE	TMOD	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit	02H	0AH
Auto-Reload	03H	0BH

NOTES :
1. Capture / reload occurs only on timer / counter overflow.
2. Capture / reload occurs on timer / counter overflow and a 1-to-0 transition on T2EX (P1.1) pin except when Timer 2 is used in the baud rate generatior mode.

Interrupt system

The MX10E8050I Serial contains a 7-source (2 external interrupts, Timer 0, Timer1, Timer2, I^2C and UART) with four priority levels interrupt structure.

Each External interrupts INT0 and INT1, can be either level-activated or transition-activated depending on bits IT0 and IT1 in TCON SFR. The flags that actually generate these interrupts are bits IE0, IE1 in TCON. When an external interrupt is generated, the corresponding request flag is cleared by the hardware where the service routine is vectored to, if the interrupt is transition-activated. If the interrupt is level-activated the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactive the request before the interrupt service routine is completed, otherwise another interrupt will be generated.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/counter register (except for Timer 0 in Mode 3 of the serial interface). When a Timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

IE : INTERRUPT ENABLE REGISTER
This register is located at address A8H.

Table. 13 IE SFR (A8H)

7	6	5	4	3	2	1	0
EA	ET2	ES1	ES	ET1	EX1	ET0	EX0
(MSB)							(LSB)

keep the above table with the following table

Table. 14 Description of IE bits

MNEMONIC	POSITION	FUNCTION
EA	IE.7	Disable all interrupt - Low, all disabled. - High, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
ET2	IE.6	Enable / Disable Timer2 interrupt. - Low, disabled - High, enabled
ES1	IE.5	Enable / Disable I^2C Interrupt. - Low, disabled - High, enabled
ES	IE.4	Enable / Disable UART interrupt. - Low, disabled - High, enabled
ET1	IE.3	Enable / Disable Timer1 overflow interrupt.
EX1	IE.2	Enable / Disable External interrupt 1. - Low, disabled - High, enabled
ET0	IE.1	Enable / disable Timer0 overflow interrupt.
EX0	IE.0	Enable / Disable External interrupt 0. - Low, disabled - High, enabled

IP : INTERRUPT PRIORITY REGISTER
This register is located at address B8H.

Table. 15 IP SFR (B8H)

7	6	5	4	3	2	1	0
-	PT2H	PS1H	PSH	PT1H	PX1H	PT0H	PX0H (LSB)

keep the above table with the following table

Table. 16 Description of IP bits

MNEMONIC	POSITION	FUNCTION
-	IP.7	RESERVED
PT2H	IP.6	Define Timer2 interrupt priority level. - High, assign a high priority level.
PS1H	IP.5	Define I^2C interrupt priority level. - High, assign a high priority level.
PSH	IP.4	Define interrupt priority level of UART.
PT1H	IP.3	Define Timer1 overflow interrupt priority level.
PX1H	IP.2	Define External interrupt 1 interrupt priority level. - High, assign a high priority level.
PT0H	IP.1	Define Timer0 overflow interrupt priority level.
PX0H	IP.0	Define External interrupt 0 interrupt priority level. - High, assign a high priority level.

IPH : INTERRUPT HIGH PRIORITY REGISTER
This register is located at address B7H.

Table. 17 IPH SFR (B7H)

7	6	5	4	3	2	1	0
-	PT2H	PS1H	PSH	PT1H	PX1H	PT0H	PX0H (LSB)

keep the above table with the following table

Table. 18 Description of IPH bits

MNEMONIC	POSITION	FUNCTION
-	IPH.7	RESERVED
PT2H	IPH.6	Define Timer2 interrupt priority level. - High, assign a high priority level.
PS1H	IPH.5	Define I^2C interrupt priority level. - High, assign a high priority level.
PSH	IPH.4	Define interrupt priority level of UART.
PT1H	IPH.3	Define Timer1 overflow interrupt priority level.
PX1H	IPH.2	Define External interrupt 1 interrupt priority level. - High, assign a high priority level.
PT0H	IPH.1	Define Timer0 overflow interrupt priority level.
PX0H	IPH.0	Define External interrupt 0 interrupt priority level. - High, assign a high priority level.

NAME	PRIORITY WITHIN LEVEL		VECTOR ADDRESS
IE0	(HIGHEST)	1	0003H
I^2C		2	002BH
TF0		3	000BH
IE1		4	0013H
TF1		5	001BH
RI + TI		6	0023H
TF2 + EXF2	(LOWEST)	7	0033H

Watchdog Timer

The Watchdog Timer (WDT) see Fig.8 , consists of an 11-bit prescaler and an 8-bit Timer formed by SFR T3. The Timer is incremented every 1.5 ms, derived from the system clock frequency of 16 MHz by the following formula : $f_{Timer} = f_{clk} / (12 \times (2048))$. The 8-bit Timer increments every 12 x 2048 cycles of the on-chip oscillator. When a Timer overflow occurs, the microcontroller is reset. The internal RESET signal is not inhibited when the external RST pin is kept 0 into high impedance, no matter if the XTAL-clock is running or not.

To prevent a system reset the Timer must be reloaded in time by the application software. If the processor suffers a hardware / software malfunction, the software will fail to reload the Timer. This failure will result in an overflow thus prevent the processor from running out of control. This time interval is determined by the 8-bit reload value that is written into register T3.

Watchdog time interval = [100 - T3] x 12 x 2048 / oscillator frequency (12x mode)
[100 - T3] x 6 x 2048 / oscillator frequency (6x mode)

The watch-dog Timer can only be reloaded if the condition flag WLE (SFR PCON bit 4) has been previously set high by software. At the moment the counter is loaded WLE is automatically cleared.

In the idle state the watchdog Timer and reset circuitry remain active.

The watchdog Timer is controlled by the watchdog enable signal EW (SFR EBTCON bit 1). A LOW level enables the watchdog Timer. A HIGH level disable the watchdog Timer.

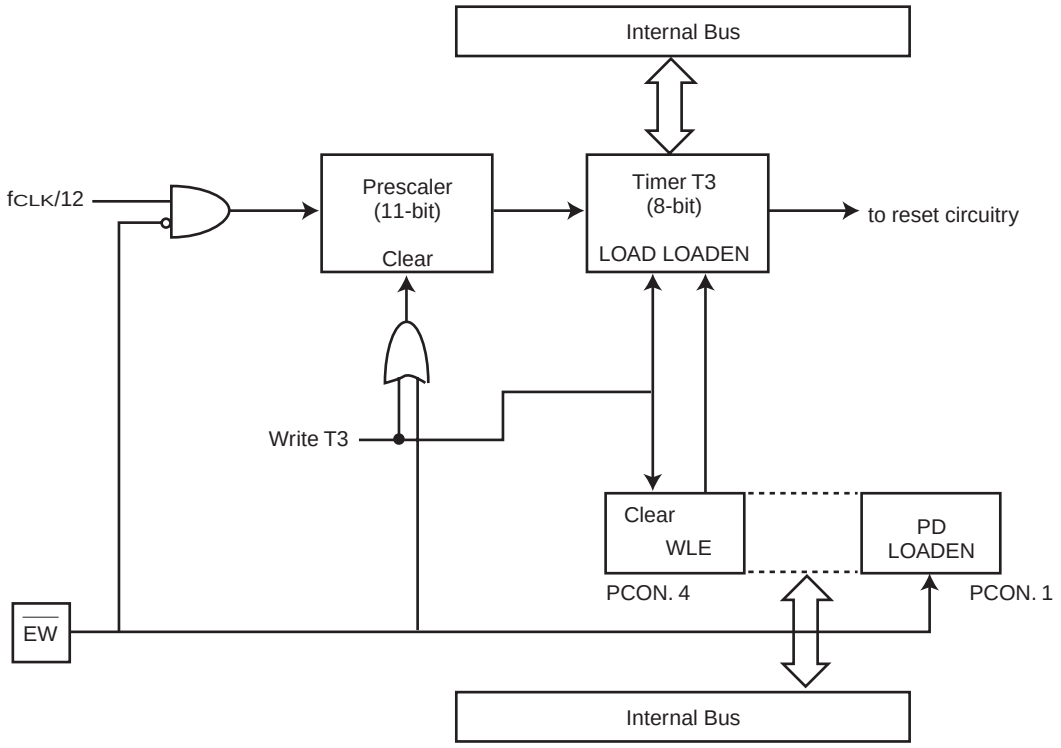


Fig. 8 Watchdog Timer T3

Pulse Width Modulated Outputs

The MX10E8050I contains four pulse width modulated output channels. These channels generate pulses of programmable length and interval. Two kinds of user modes are available. One is to use two channels as a pair of PWM output with one prescaler and four channels as two pairs of PWM outputs with each own single prescaler. The operation thus is like two set of independently PWM modules. The repetition frequency is defined by an 8-bit prescaler, which supplies the clock for the counter. The prescaler and counter are common to the both PWM channels in each set. The 8-bit counter counts modulo 255, i.e., from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of two registers: PWM0 and PWM1 or PWM2 and PWM3. Provided the contents of either of these registers is greater than the counter value, the corresponding PWM0 or PWM1 or PWM2 or PWM3 output is set LOW. If the contents of these registers are equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the registers PWM0 and PWM1 or PWM2 and PWM3. The pulse-width-ratio is in the range of 0 to 1 and may be programmed in increments of 1/255. The other one operation is that to use four channels as four independently PWM outputs with each own prescaler.

$$f_{PWM} = \frac{f_{OSC}}{2 \times (1 + PWMP) \times 255}$$

This gives a repetition frequency range of 123Hz to 31.4KHz (f_{OSC} = 16MHz). At f_{OSC} = 24MHz, the frequency range is 184Hz to 47.1KHz. By loading the PWM registers with either 00H or FFH, the PWM channels will output a constant HIGH or LOW level, respectively. Since the 8-bit counter counts modulo 255, it can never actually reach the value of the PWM registers when they are loaded with FFH.

When a compare register (PWM0 or PWM1 or PWM2 or PWM3) is loaded with a new value, the associated output is updated immediately. It does not have to wait until the end of the current counter period. Every PWMn output pins are driven by push-pull drivers. These pins are not used for any other purpose.

The PWM function is enabled by setting SFR PWMC. SFR PWMC also controls operational mode and enable out. After reset, P4.0 to P4.3 are used to as the PWM output.

PWM Module Control Register/PWMC (F1H)

PWMD	DSCB	PWM3E	PWM2E	-	DSCA	PWM1E	PWM0E

. PWMD: Enable/Disable PWM function (if 1, then disable PWM)
. DSCB: Dual or single output select for B module (if 0, then Dual output)
. PWM3E: PWM3 output enable
. PWM2E: PWM2 output enable
. DSCA: Dual or single output select for A module (if 0, then Dual output)
. PWM1E: PWM1 output enable
. PWM0E: PWM0 output enable

PWM Module Prescaler frequency Control Register/PWMPX

PWMPX.7	PWMPX.6	PWMPX.5	PWMPX.4	PWMPX.3	PWMPX.2	PWMPX.1	PWMPX.0

PWMPX	X = 0, 1, 2, or 3	
	PWMP0	0FEH
	PWMP1	0FBH
	PWMP2	0F6H
	PWMP3	0F2H

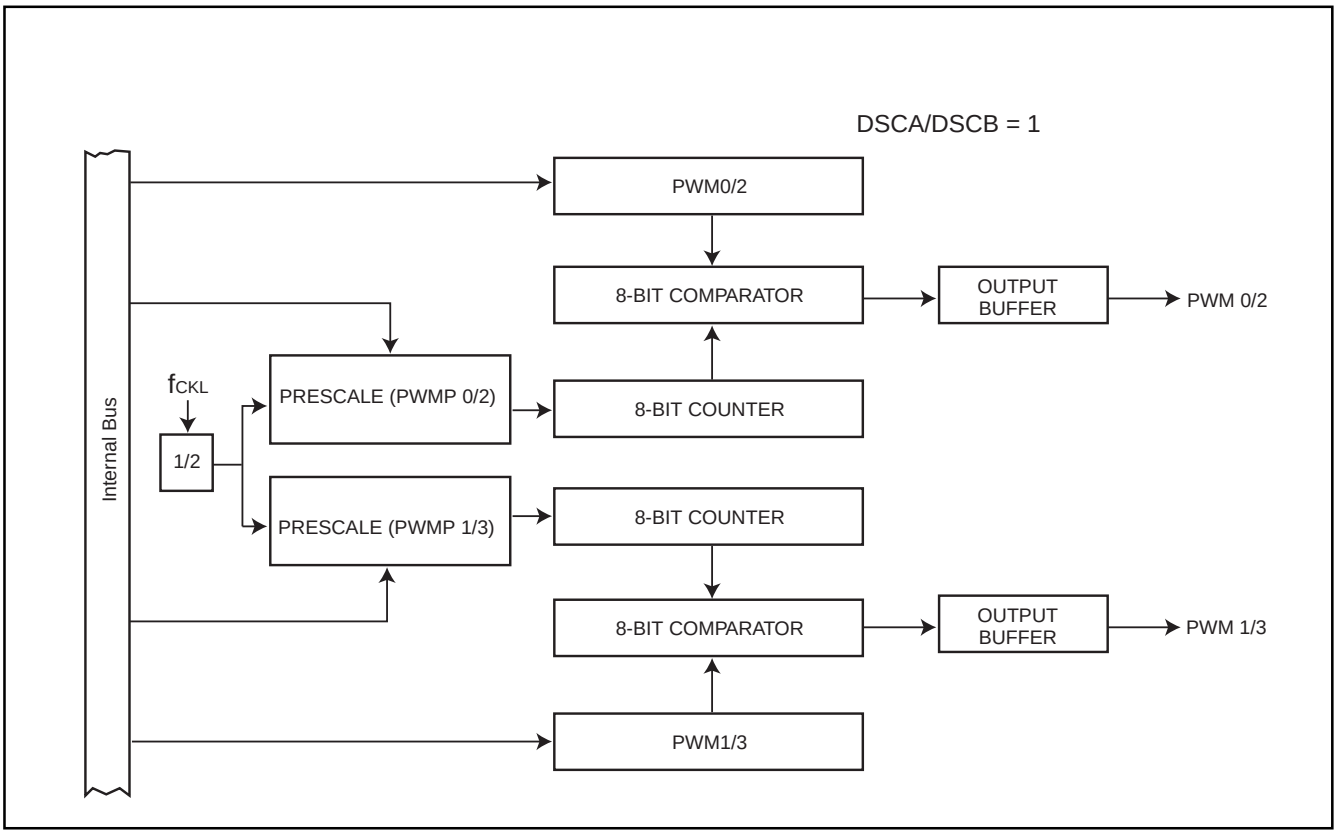
BIT	SYMBOL	FUNCTION
PWMPX.7~0	PWMPX.7~0	Prescaler division factor = (PWMPX) + 1

PWM Module Pulse width Register/PWMX

PWMX.7	PWMX.6	PWMX.5	PWMX.4	PWMX.3	PWMX.2	PWMX.1	PWMX.0

PWMX	X = 0, 1, 2, or 3	
	PWM0	0FCH
	PWM1	0FDH
	PWM2	0F4H
	PWM3	0F5H

BIT	SYMBOL	FUNCTION
PWMX.7~0	PWMX.7~0	LOW/HIGH ration of PWMX signal = (PWMX)/[255-(PWMX)]



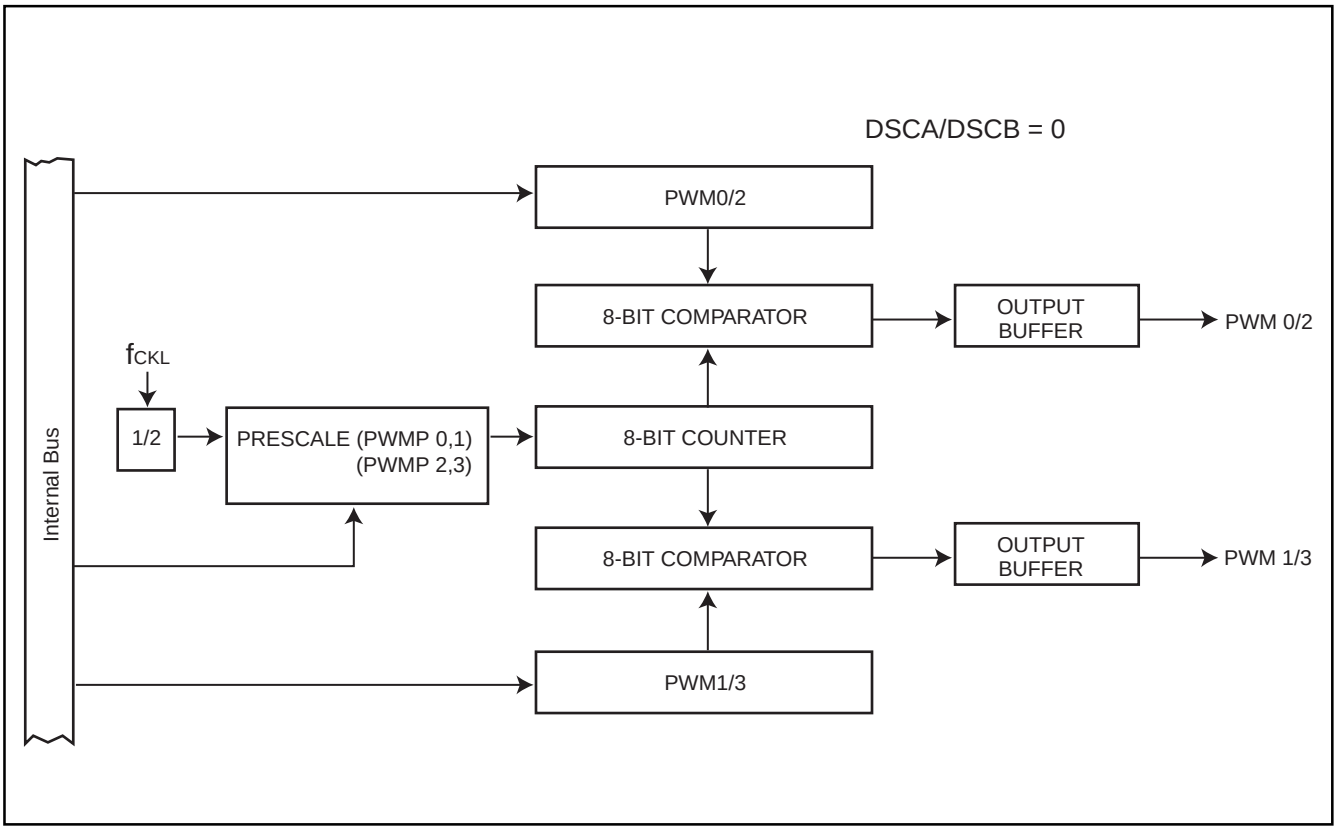


Fig. 9 Functional Diagram of Pulse Width Modulated Outputs

UART

Enhanced UART

In addition to the standard operation the UART can perform framing error detect by looking for missing stop bits, and automatic address recognition. The UART also fully supports multiprocessor communication as does the standard 80C51 UART.
When used for framing error detect the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0) (see Figure 10). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE SCON.7 can only be cleared by software.

Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the “Given” address or the “Broadcast” address. The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recogintion is shown in figure 12.

The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave’s address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to b used and which bits are “don’t care”. The SADEN mask can be logically ANDed with the SADDR to create the “Given” address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0	SADDR =	1100 0000
	SADEN =	1111 1101
	Given =	1100 00X0
Slave 1	SADDR =	1100 0000
	SADEN =	1111 1110
	Given =	1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0	SADDR =	1100 0000
	SADEN =	1111 1001
	Given =	1100 0XX0
Slave 1	SADDR =	1110 0000
	SADEN =	1111 1010
	Given =	1110 0X0X
Slave 2	SADDR =	1110 0000
	SADEN =	1111 1100
	Given =	1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are trended as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are leaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

SCON Address = 98H Reset Value = 0000 0000B

Bit Addressable

	SM0/FE	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
Bit:	7	6	5	4	3	2	1	0
	(SMOD0 = 0/1)*							

Symbol	**Function**
FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
SM1	Serial Port Mode Bit 1

SM0	**SM1**	**Mode**	**Description**	**Baud Rate****
0	0	0	shift register	f_{OSC}/6 (6-clock mode) or f_{OSC}/12 (12-clock mode)
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	f_{OSC}/32 or f_{OSC}/16 (6-clock mode) or f_{OSC}/64 or f_{OSC}/32 (12-clock mode)
1	1	3	9-bit UART	variable

SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
RB8	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

NOTE:
*SMOD0 is located at PCON6.
**f_{OSC} = oscillator frequency

Figure 10. SCON : Serial Port Control Register

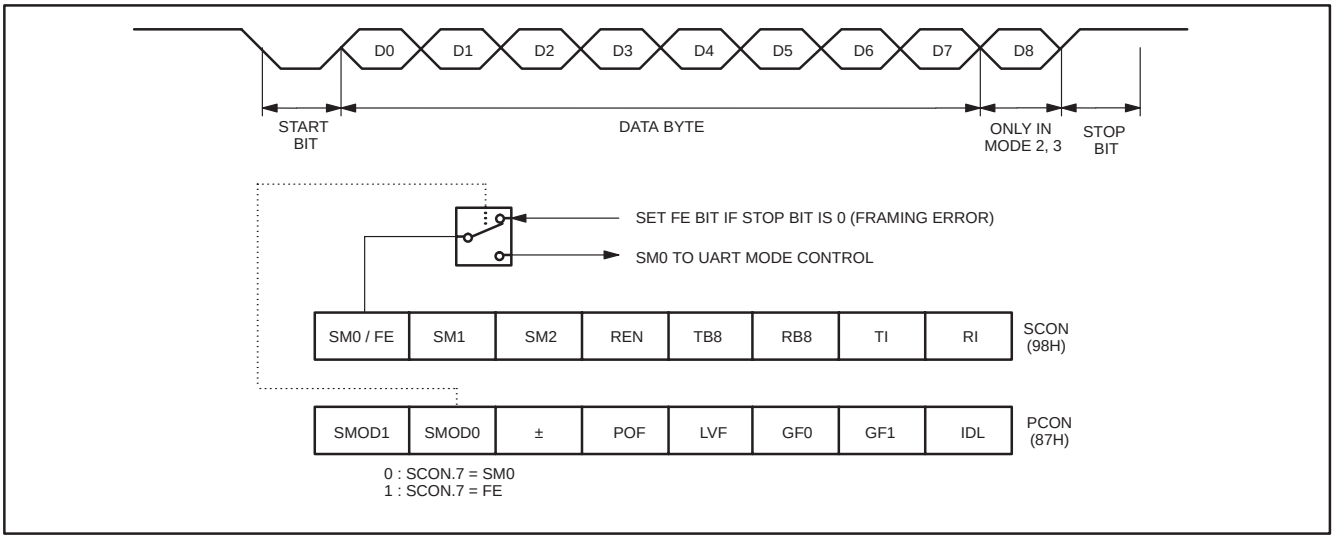


Figure 11. UART Framing Error Detection

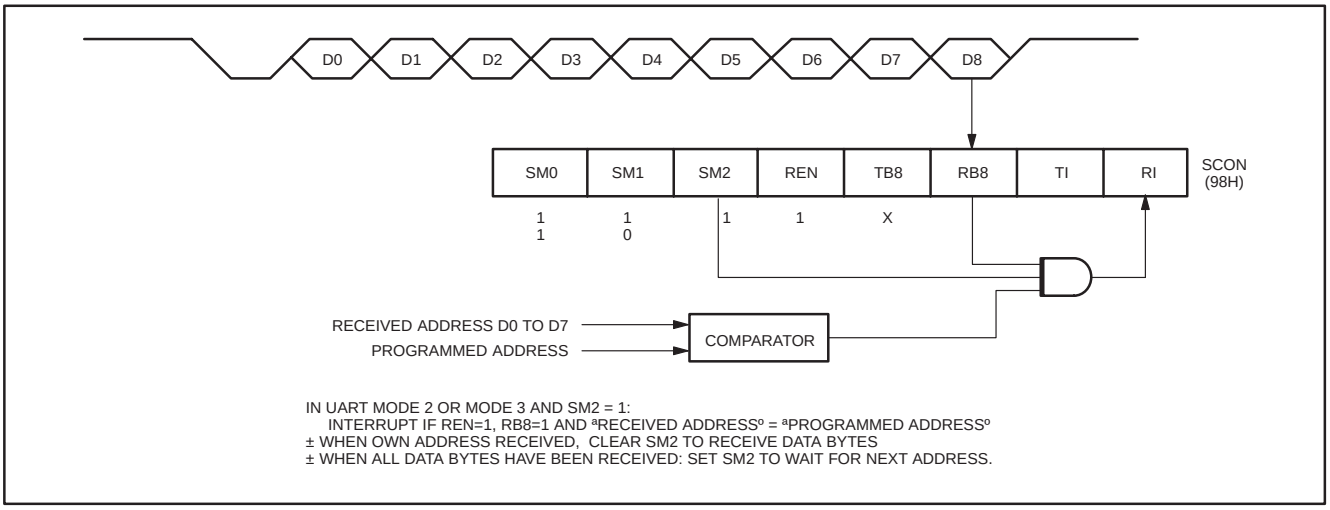


Figure 12. UART Multiprocessor Communication, Automatic Address Recognition

Serial I/O

The MX10E8050I Serial is equipped with two independent serial ports : SIO0 and SIO1. SIO0 is a full duplex UART port and is identical to the 80C51 serial port.

SIO0 : SIO0 is a full duplex serial I/O port identical to that on the 80C51. It's operation is the same, including the use of timer 1 as a baud rate generator.

SIO1, I^2C Serial I/O : The I^2C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are :

- Bidirectional data transfer between masters and slaves
- Multimaster bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- The I^2C bus may be used for test and diagnostic purposes

The output latches of P1.6 and P1.7 must be set to logic 1 in order to enable SIO1.

The MX10E8050I Serial on-chip I^2C logic provides a serial interface that meets the I^2C bus specification and supports all transfer modes (other than the low-speed mode) from and to the I^2C bus. The SIO1 logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (S1STA) reflects the status of SIO1 and the I^2C bus.

The CPU interfaces to the I^2C logic via the following four special function register : S1CON (SIO1 control register), S1STA (SIO1 status register), S1DAT (SIO1 data register), and S1ADR (SIO1 slave address register). The SIO1 logic interfaces to the external I^2C bus via two port 1 pins : P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I^2C bus configuration is shown in Figure 13, and Figure 14 shows how a data transfer is accomplished on the bus. Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I^2C bus:

1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.

2. Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a “not acknowledge” is returned.
The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I^2C bus will not be released.

Modes of Operation: The on-chip SIO1 logic may operate in the following four modes:

1. Master Transmitter Mode:
Serial data output through P1.7/SDA while P1.6/SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 0, and we say that a "W" is transmitted. Thus the first byte transmitted is SLA+W. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

2. Master Receiver Mode:
The first byte transmitted contains the slave address of the transmitting device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 1, and we say that an "R" is transmitted. Thus the first byte transmitted is SLA+R. Serial data is received via P1.7/SDA while P1.6/SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

3. Slave Receiver Mode:
Serial data and the serial clock are received through P1.7/SDA and P1.6/SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

4. Slave Transmitter Mode:
The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via P1.7/SDA while the serial clock is input through P1.6/SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

In a given application, SIO1 may operate as a master and as a slave. In the slave mode, the SIO1 hardware looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, SIO1 switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

SIO1 Implementation and Operation: Figure 15 shows how the on-chip I^2C bus interface is implemented, and the following text describes the individual blocks.

INPUT FILTERS AND OUTPUT STAGES

The input filters have I^2C compatible input levels. If the input voltage is less than 1.5V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock ($f_{OSC}/4$), and spikes shorter than three oscillator periods are filtered out.

The output stages consist of open drain transistors that can sink 3mA at $V_{OUT} < 0.4V$. These open drain outputs do not have clamping diodes to V_{DD}. Thus, if the device is connected to the I^2C bus and V_{DD} is switched off, the I^2C bus is not affected.

ADDRESS REGISTER, S1ADR

This 8-bit special function register may be loaded with the 7-bit slave address (7 most significant bits) to which SIO1 will respond when programmed as a slave transmitter or receiver. The LSB (GC) is used to enable general call address (00H) recognition.

COMPARATOR

The comparator compares the received 7-bit slave address with its own slave address (7 most significant bits in S1ADR). It also compares the first received 8-bit byte with the general call address (00H). If an equality is found, the appropriate status bits are set and an interrupt is requested.

SHIFT REGISTER, S1DAT

This 8-bit special function register contains a byte of serial data to be transmitted or a byte which has just been received. Data in S1DAT is always shifted from right to left; the first bit to be transmitted is the MSB (bit 7) and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.

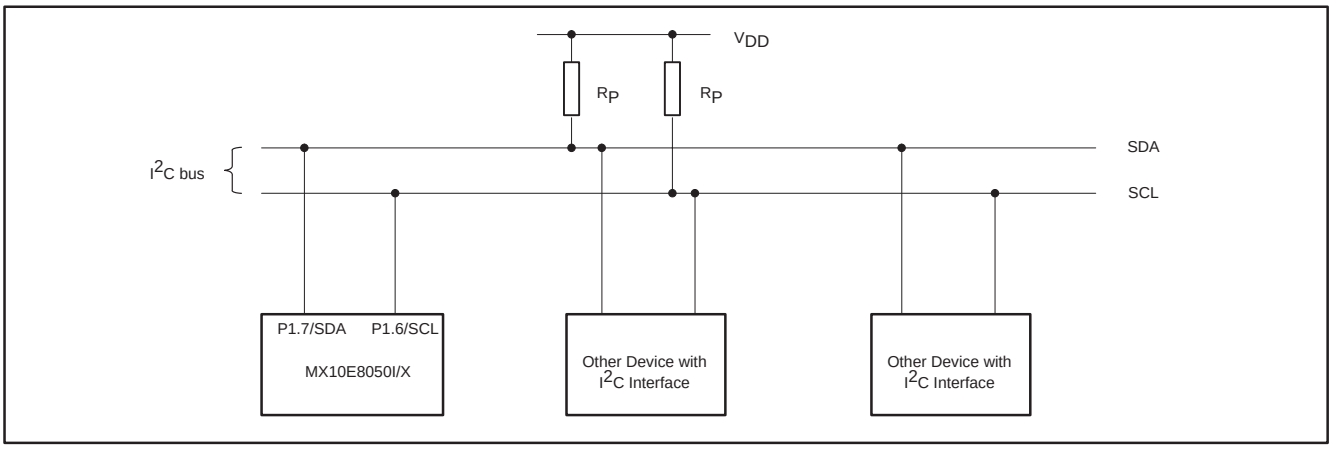


Figure 13. Typical I²C Bus Configuration

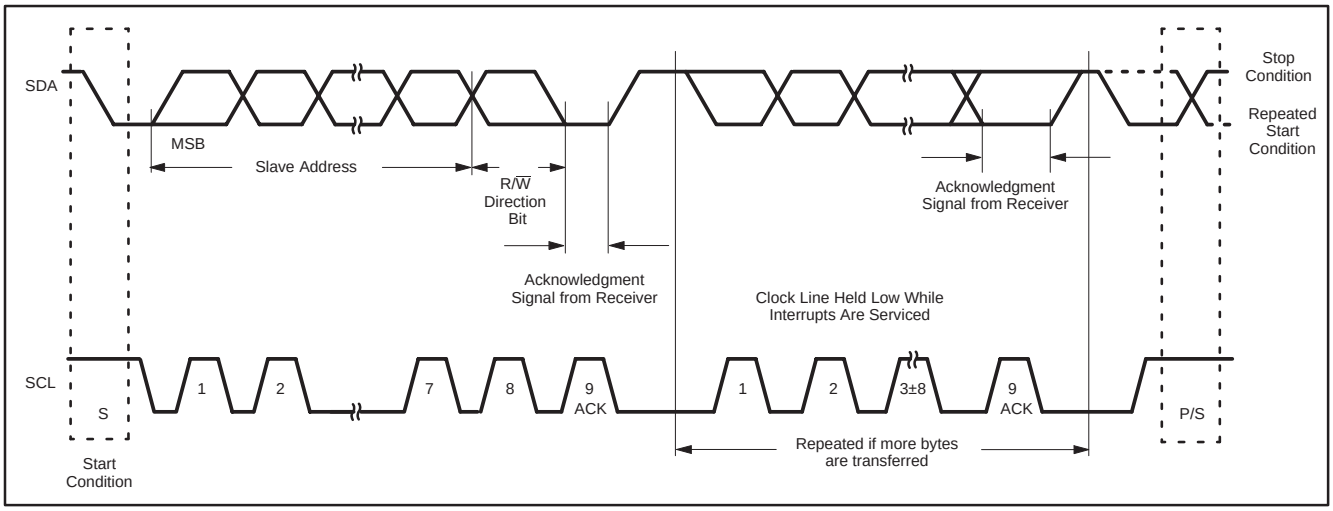


Figure 14. Data Transfer on the I²C Bus

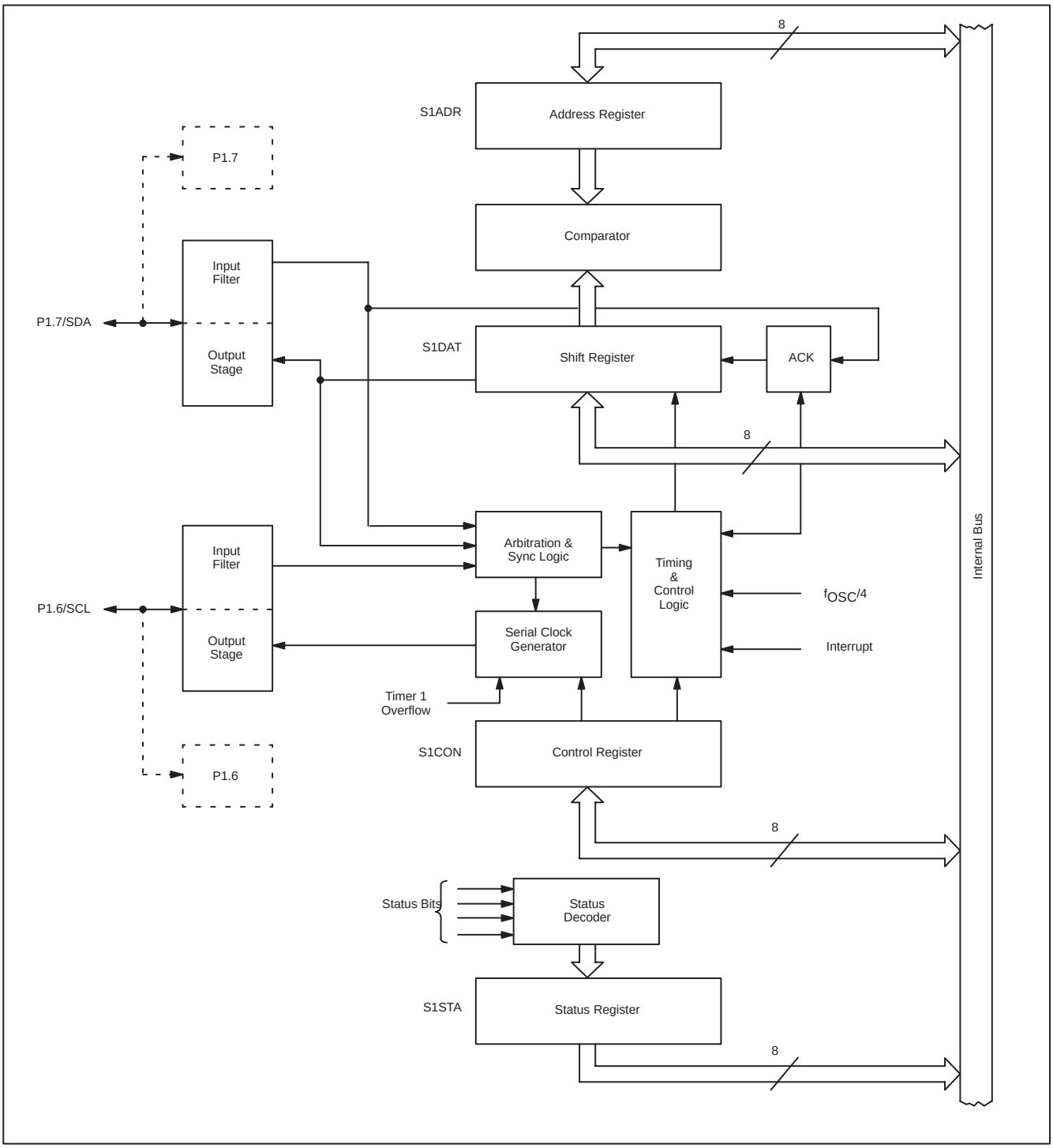
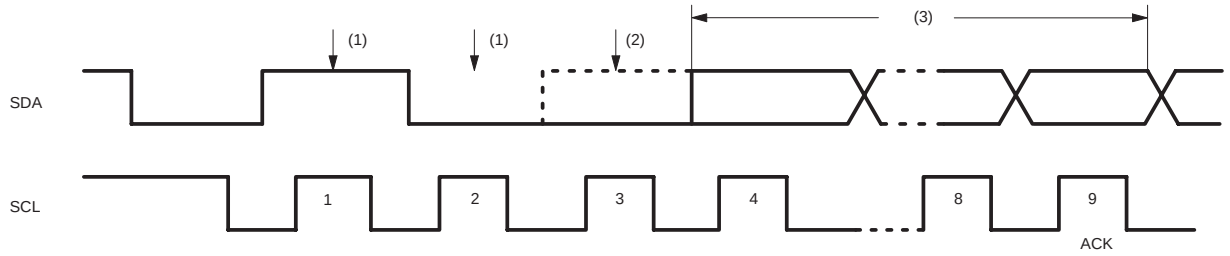
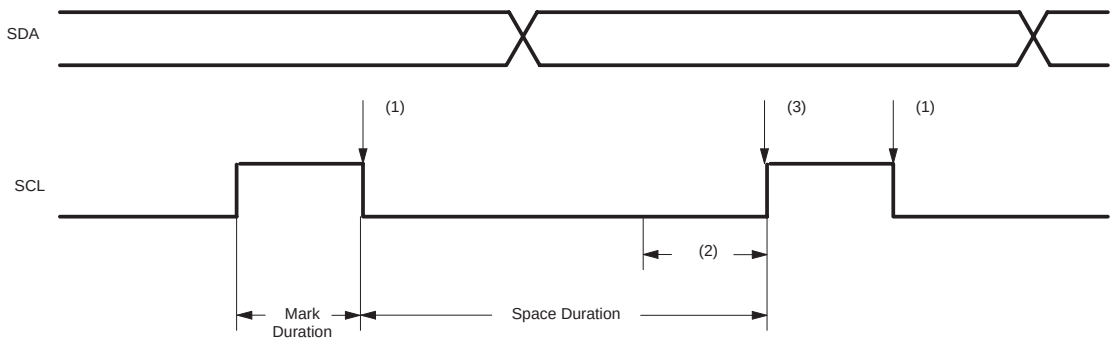


Figure 15. I²C Bus Serial Interface Block Diagram



1. Another device transmits identical serial data.
2. Another device overrules a logic 1 (dotted line) transmitted by SIO1 (master) by pulling the SDA line low. Arbitration is lost, and SIO1 enters the slave receiver mode.
3. SIO1 is in the slave receiver mode but still generates clock pulses until the current byte has been transmitted. SIO1 will not generate clock pulses for the next byte. Data on SDA originates from the new master once it has won arbitration.

Figure 16. Arbitration Procedure



1. Another service pulls the SCL line low before the SIO1"mark" duration is complete. The serial clock generator is immediately reset and commences with the "space" duration by pulling SCL low.
2. Another device still pulls the SCL line low after SIO1 releases SCL. The serial clock generator is forced into the wait state until the SCL line is released.
3. The SCL line is released, and the serial clock generator commences with the mark duration.

Figure 17. Serial Clock Synchronization

ARBITRATION AND SYNCHRONIZATION LOGIC

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I^2C bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost, and SIO1 immediately changes from master transmitter to slave receiver. SIO1 will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while SIO1 is returning a "not acknowledge: (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal LOW. Since this can occur only at the end of a serial byte, SIO1 generates no further clock pulses.
Figure 16 shows the arbitration procedure.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the "mark" duration is determined by the device that generates the shortest "marks," and the "space" duration is determined by the device that generates the longest "spaces." Figure 17 shows the synchronization procedure.

A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. SIO1 will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.

SERIAL CLOCK GENERATOR

This programmable clock pulse generator provides the SCL clock pulses when SIO1 is in the master transmitter or master receiver mode. It is switched off when SIO1 is in a slave mode. The programmable output clock frequencies are: $f_{osc}/120$, $f_{osc}/9600$, and the Timer 1 overflow rate divided by eight. The output clock pulses have a 50% duty cycle unless the clock generator is synchronized with other SCL clock sources as described above.

TIMING AND CONTROL

The timing and control logic generates the timing and control signals for serial byte handling. This logic block provides the shift pulses for S1DAT, enables the comparator, generates and detects start and stop conditions, receives and transmits acknowledge bits, controls the master and slave modes, contains interrupt request logic, and monitors the I^2C bus status.

CONTROL REGISTER, S1CON

This 7-bit special function register is used by the microcontroller to control the following SIO1 functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

STATUS DECODER AND STATUS REGISTER

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I^2C bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of SIO1 are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code is sufficient for most of the service routines (see the software example in this section).

The Four SIO1 Special Function Registers: The microcontroller interfaces to SIO1 via four special function registers. These four SFRs (S1ADR, S1DAT, S1CON, and S1STA) are described individually in the following sections.

The Address Register, S1ADR: The CPU can read from and write to this 8-bit, directly addressable SFR. S1ADR is not affected by the SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a master mode. In the slave modes, the seven most significant bits must be loaded with the microcontroller' s own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored.

	7	6	5	4	3	2	1	0
S1ADR (DBH)	X	X	X	X	X	X	X	GC
	own slave address							

The most significant bit corresponds to the first bit received from the I^2C bus after a start condition. A logic 1 in S1ADR corresponds to a high level on the I^2C bus, and a logic 0 corresponds to a low level on the bus.

The Data Register, S1DAT: S1DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in S1DAT remains stable as long as SI is set. Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.

	7	6	5	4	3	2	1	0
S1ADR (DAH)	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
	shift direction							

SD7 - SD0:

Eight bits to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I^2C bus, and a logic 0 corresponds to a low level on the bus. Serial data shifts through S1DAT from right to left. Figure 18 shows how data in S1DAT is serially transferred to and from the SDA line.

S1DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled by the SIO1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into S1DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into S1DAT, the serial data is available in S1DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from S1DAT via a buffer (BSD7) on the falling edges of clock pulses on the SCL line.

When the CPU writes to S1DAT, BSD7 is loaded with the content of S1DAT.7, which is the first bit to be transmitted to the SDA line (see Figure 19). After nine serial clock pulses, the eight bits in S1DAT will have been transmitted to the SDA line, and the acknowledge bit will be present in ACK. Note that the eight transmitted bits are shifted back into S1DAT.

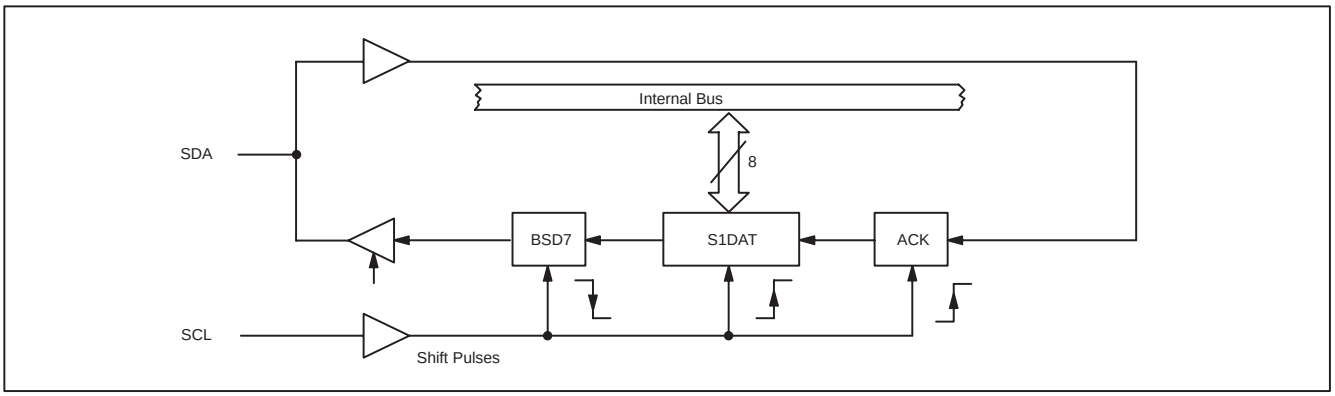


Figure 18. Serial Input/Output Configuration

The Control Register, S1CON: The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I^2C bus. The STO bit is also cleared when ENS1 = "0".

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

ENS1, THE SIO1 ENABLE BIT
ENS1 = "0": When ENS1 is "0", the SDA and SCL outputs are in a high impedance state. SDA and SCL input signals are ignored, SIO1 is in the "not addressed" slave state, and the STO bit in S1CON is forced to "0". No other bits are affected. P1.6 and P1.7 may be used as open drain I/O ports.

ENS1 = "1": When ENS1 is "1", SIO1 is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

ENS1 should not be used to temporarily release SIO1 from the I^2C bus since, when ENS1 is reset, the I^2C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

In the following text, it is assumed that ENS1 = "1".

STA, THE START FLAG
STA = "1": When the STA bit is set to enter a master mode, the SIO1 hardware checks the status of the I^2C bus and generates a START condition if the bus is free. If the bus is not free, then SIO1 waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.

If STA is set while SIO1 is already in a master mode and one or more bytes are transmitted or received, SIO1 transmits a repeated START condition. STA may be set at any time. STA may also be set when SIO1 is an addressed slave.
STA = "0": When the STA bit is reset, no START condition or repeated START condition will be generated.

STO, THE STOP FLAG
STO = "1": When the STO bit is set while SIO1 is in a master mode, a STOP condition is transmitted to the I^2C bus. When the STOP condition is detected on the bus, the SIO1 hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I^2C bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined "not addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

If the STA and STO bits are both set, the a STOP condition is transmitted to the I^2C bus if SIO1 is in a master mode (in a slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.
STO = “0”: When the STO bit is reset, no STOP condition will be generated.

SI, THE SERIAL INTERRUPT FLAG
SI = “1”: When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of 25 of the 26 possible SIO1 states is entered. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = “0”: When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

AA, THE ASSERT ACKNOWLEDGE FLAG
AA = “1”: If the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- The “own slave address” has been received
- The general call address has been received while the general call bit (GC) in S1ADR is set
- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

AA = “0”: if the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

When SIO1 is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 23). When SI is cleared, SIO1 leaves state C8H, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I^2C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the part’s own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

CR0, CR1, AND CR2, THE CLOCK RATE BITS
These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 19.

A 12.5kHz bit rate may be used by devices that interface to the I^2C bus via standard I/O port lines which are software driven and slow. 100kHz is usually the maximum bit rate and can be derived from a 16MHz, 12MHz, or a 6MHz oscillator. A variable bit rate (0.5kHz to 62.5kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a master mode.

The frequencies shown in Table 19 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100kHz.

The Status Register, S1STA:
S1STA is an 8-bit read-only special function register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 26 possible status codes. When S1STA contains F8H, no relevant state information is available and no serial interrupt is requested. All other S1STA values correspond to defined SIO1 states. When each of these states is entered, a serial interrupt is requested (SI = "1"). A valid status code is present in S1STA one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

More Information on SIO1 Operating Modes: The four operating modes are:
- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in Figures 20~28. These figures contain the following abbreviations:

Abbreviation	Explanation
S	Start condition
SLA	7-bit slave address
R	Read bit (high level at SDA)
W	Write bit (low level at SDA)
$\underline{A}$	Acknowledge bit (low level at SDA)
$\overline{A}$	Not acknowledge bit (high level at SDA)
Data	8-bit data byte
P	Stop condition

In Figures 20~28, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the S1STA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in S1STA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Table 20~24.

Master Transmitter Mode:
In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 20). Before the master transmitter mode can be entered, S1CON must be initialized as follows:

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	bit rate	1	0	0	0	x	bit rate	

CR0, CR1, and CR2 define the serial bit rate. ENS1 must be set to logic 1 to enable SIO1. If the AA bit is reset, SIO1 will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO0 cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit using the SETB instruction. The SIO1 logic will now test the I^2C bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (S1STA) will be 08H. This status code must be used to vector to an interrupt service routine that loads S1DAT with the slave address and the data direction bit (SLA+W). The SI bit in S1CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. There are 18H, 20H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 20. After a repeated start condition (state 10H). SIO1 may switch to the master receiver mode by loading S1DAT with SLA+R).

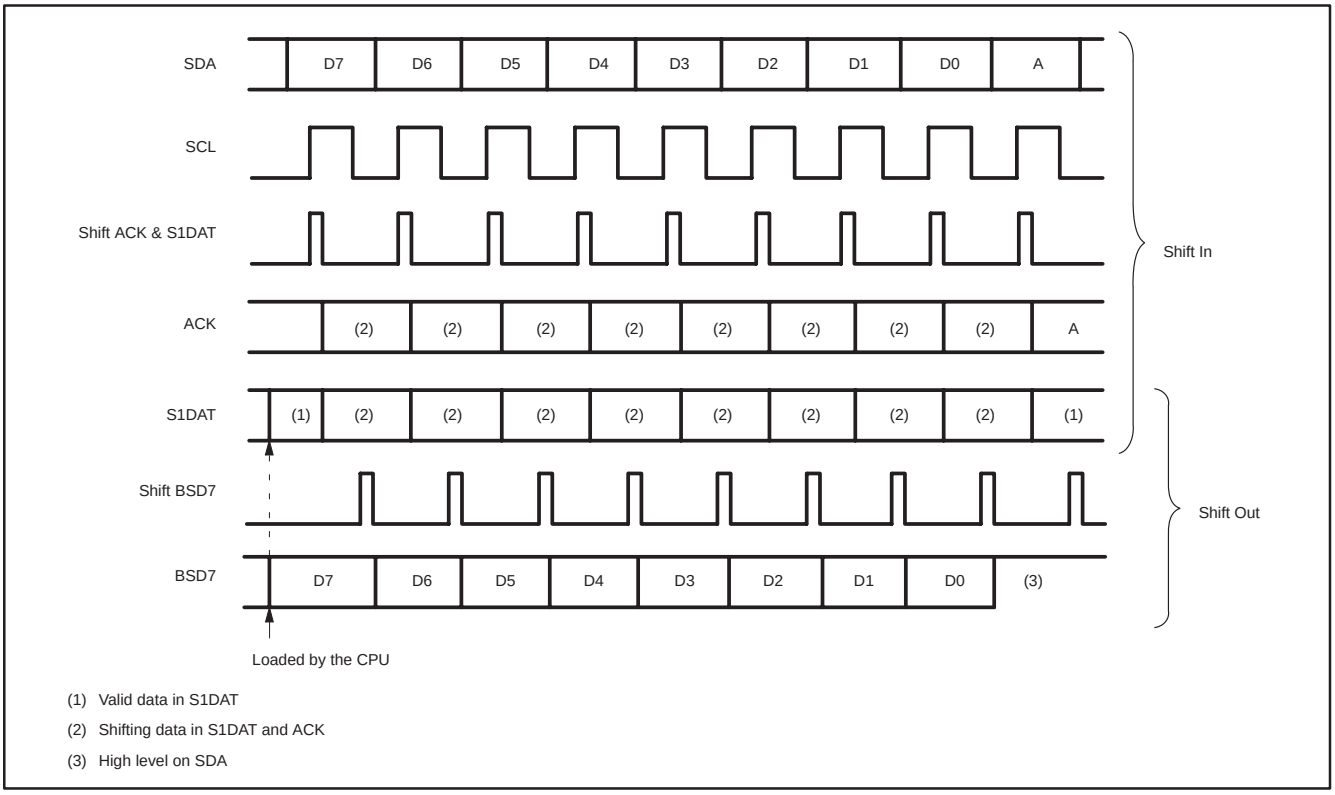


Figure 19. Shift-in and Shift-out Timing

Table 19 : Serial Clock Rates

			BIT FREQUENCY (kHz) AT f_{OSC}			
CR2	CR1	CR0	6MHz	12MHz	16MHz	f_{OSC} DIVIDED BY
0	0	0	23	47	63	256
0	0	1	27	54	71	224
0	1	0	31	63	83	192
0	1	1	37	75	100	160
1	0	0	6.25	12.5	17	960
1	0	1	50	100	-	120
1	1	0	100	-	-	60
1	1	1	0.25<62.5	0.5<62.5	0.67<56	96x(256-reload value Timer1) (Reload value range:0-254 in mode2)

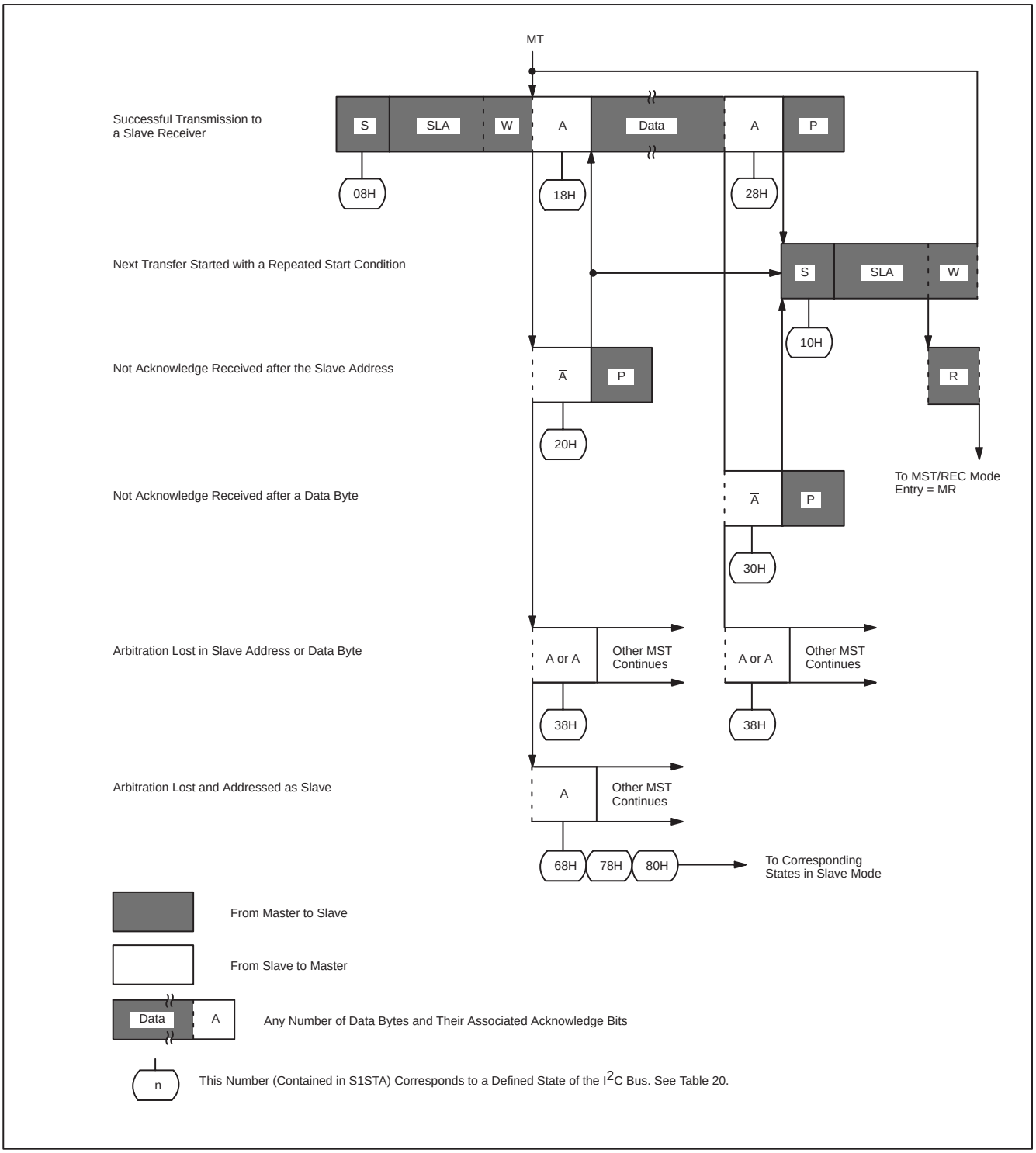


Figure 20. Format and States in the Master Transmitter Mode

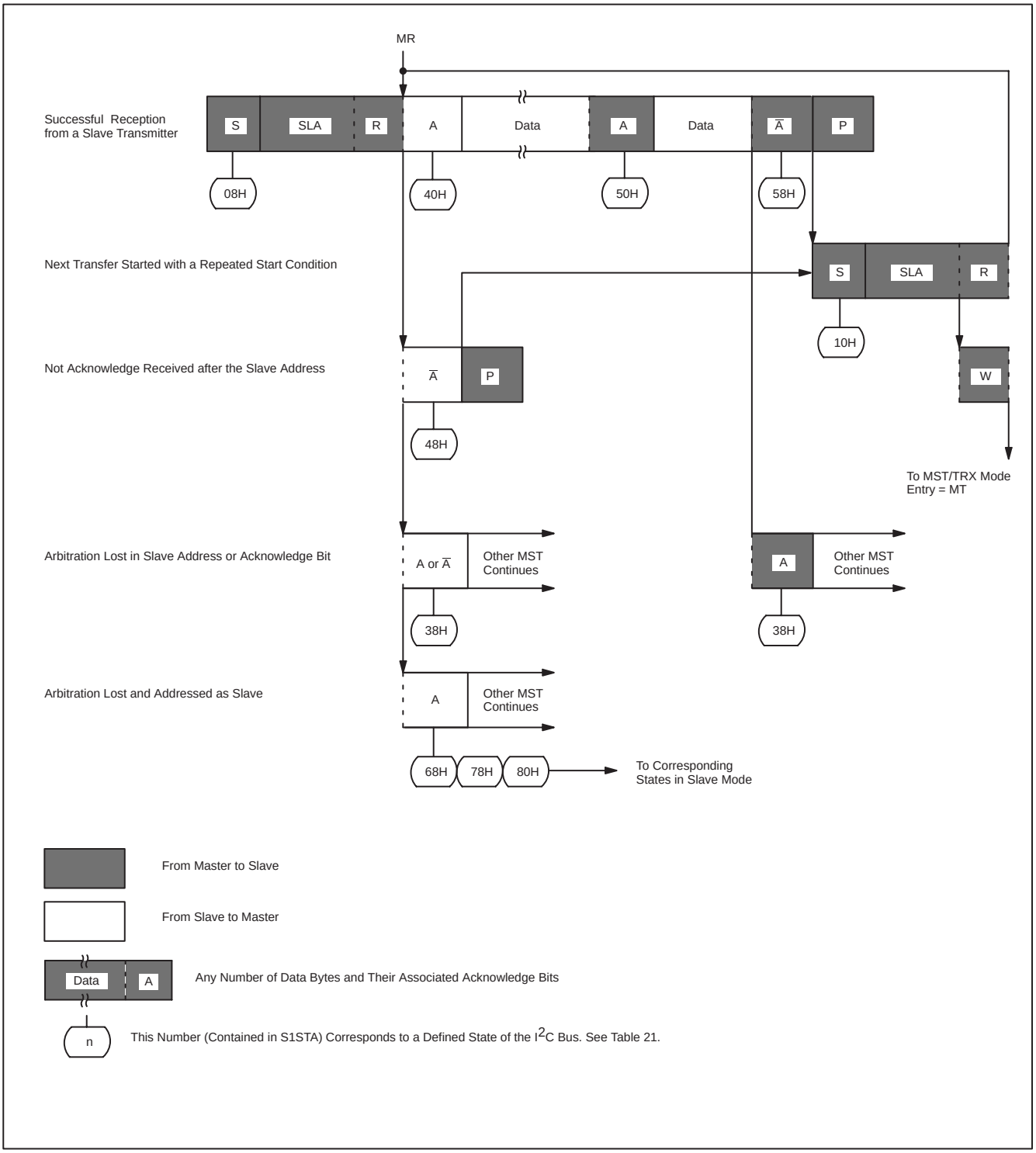


Figure 21. Format and States in the Master Receiver Mode

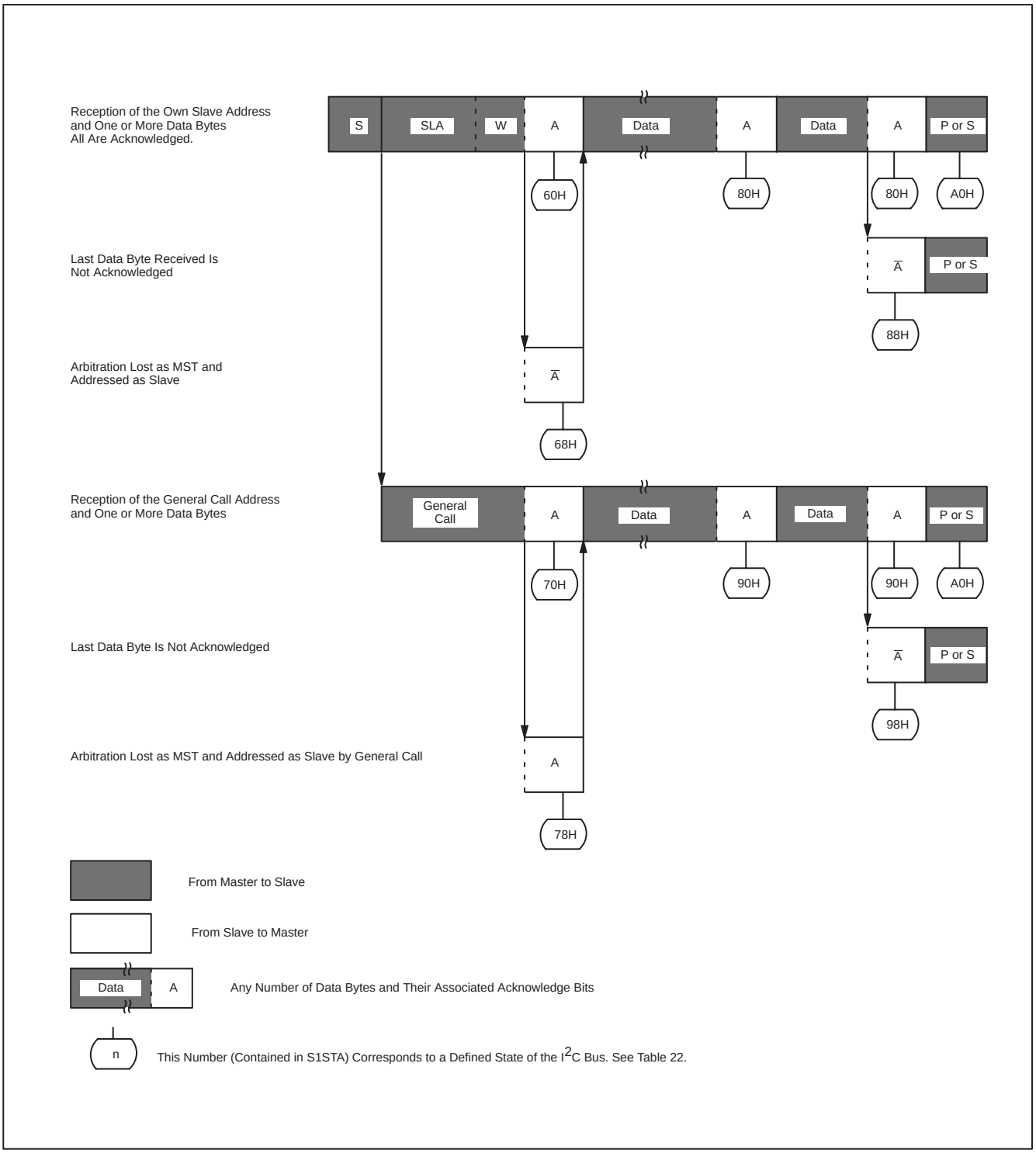


Figure 22. Format and States in the Slave Receiver Mode

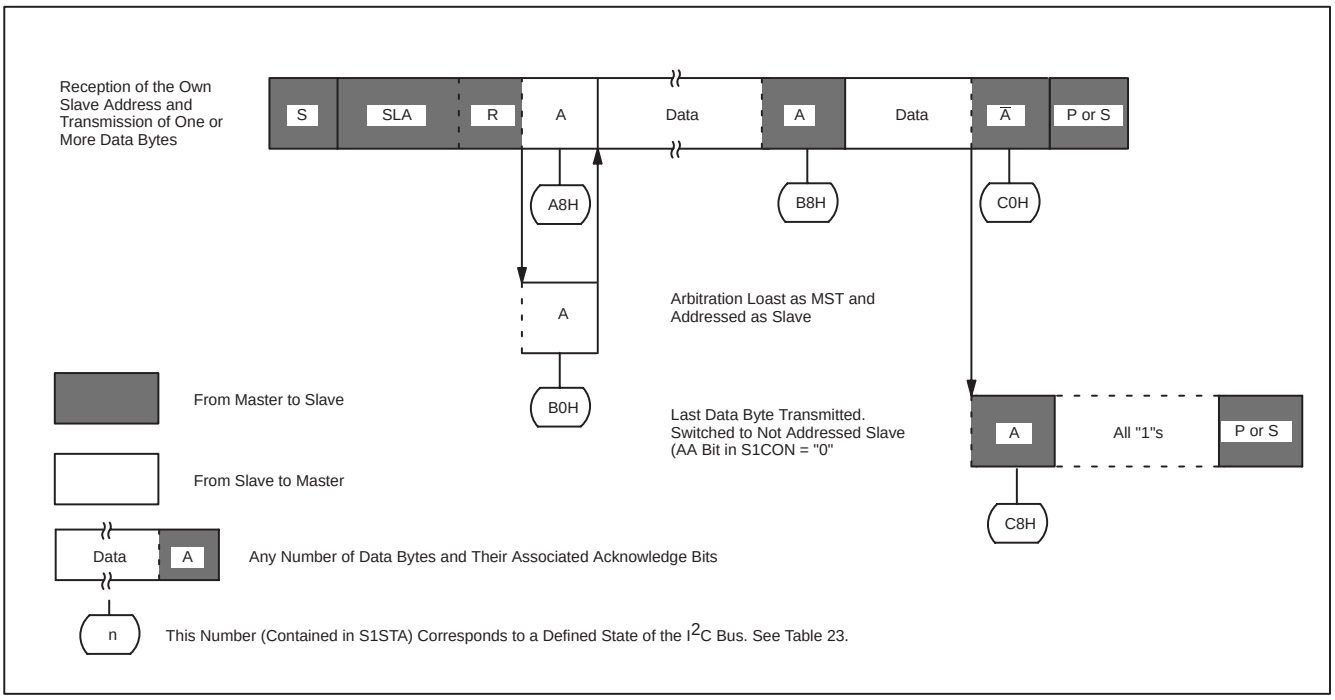


Figure 23. Format and States of the Slave Transmitter Mode

Master Receiver Mode:
In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 21). The transfer is initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load S1DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in S1CON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. These are 40H, 48H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 21. ENS1, CR1, and CR0 are not affected by the serial transfer and are not referred to in Table 21. After a repeated start condition (state 10H), SIO1 may switch to the master transmitter mode by loading S1DAT with SLA+W.

Slave Receiver Mode:
In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 22). To initiate the slave receiver mode, S1ADR and S1CON must be loaded as follows:

	7	6	5	4	3	2	1	0
S1ADR (DBH)	X	X	X	X	X	X	X	GC
	own slave address							

The upper 7 bits are the address to which SIO1 will respond when addressed by a master. If the LSB (GC) is set, SIO1 will respond to the general call address (00H); otherwise it ignores the general call address.

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	X	1	0	0	0	1	X	X

CR0, CR1, and CR2 do not affect SIO1 in the slave mode. ENS1 must be set to logic 1 to enable SIO1. The AA bit must be set to enable SIO1 to acknowledge its own slave address or the general call address. STA, STO, and SI must be reset.

When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for SIO1 to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (I) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 22. The slave receiver mode may also be entered if arbitration is lost while SIO1 is in the master mode (see status 68H and 78H).

If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^2C bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I^2C bus.

Table 20 : Master Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+W or Load SLA+R	X X	0 0	0 0	X X	As above SLA+W will be transmitted; SIO1 will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or no S1DAT action or no S1DAT action or no S1DAT action	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or no S1DAT action or no S1DAT action or no S1DAT action	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset
28H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or no S1DAT action or no S1DAT action or no S1DAT action	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in S1DAT has been transmitted; NOT ACK has been received	Load data byte or no S1DAT action or no S1DAT action or no S1DAT action	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	Data byte will be transmitted; ACK bit will be received Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R/$\overline{W}$ or Data bytes	No S1DAT action or No S1DAT action	0 1	0 0	0 0	X X	I²C bus will be released; not addressed slave will be entered A START condition will be transmitted when the bus becomes free

Table 21 : Master Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or Load SLA+W	X X	0 0	0 0	X X	As above SLA+W will be transmitted; SIO1 will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	No S1DAT action or No S1DAT action	0 1	0 0	0 0	X X	I²C bus will be released; SIO1 will enter a slave mode A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No S1DAT action or no S1DAT action	0 0	0 0	0 0	0 1	Data byte will be received; NOT ACK bit will be returned Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No S1DAT action or no S1DAT action or no S1DAT action	1 0 1	0 1 1	0 0 0	X X X	Repeated START condition will be transmitted STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte or read data byte	0 0	0 0	0 0	0 1	Data byte will be received; NOT ACK bit will be returned Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or read data byte or read data byte	1 0 1	0 1 1	0 0 0	X X X	Repeated START condition will be transmitted STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset

Table 22 : Slave Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
60H	Own SLA+W has been received; ACK has been returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/$\overline{W}$ as master; Own SLA+W has been received, ACK returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
70H	General call address (00H) has been received; ACK has been returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
78H	Arbitration lost in SLA+R/$\overline{W}$ as master; General call address has been received, ACK has been returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK has been returned	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		read data byte	X	0	0	1	Data byte will be received and ACK will be returned
88H	Previously addressed with own SLA; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		read data byte or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		read data byte	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
90H	Previously addressed with General Call; DATA byte has been received; ACK has been returned	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		read data byte	X	0	0	1	Data byte will be received and ACK will be returned
98H	Previously addressed with General Call; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		read data byte or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		read data byte	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 22 : Slave Receiver Mode (Continued)

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No STDAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		No STDAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		No STDAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		No STDAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 23 : Slave Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/$\overline{W}$ as master; Own SLA+R has been received, ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received
B8H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received
C0H	Data byte in S1DAT has been transmitted; NOT ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		no S1DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		no S1DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		no S1DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
C8H	Last data byte in S1DAT has been transmitted (AA = 0); ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		no S1DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		no S1DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		no S1DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Slave Transmitter Mode:
In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 23). Data transfer is initialized as in the slave receiver mode. When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be "1" (R) for SIO1 to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 23. The slave transmitter mode may also be entered if arbitration is lost while SIO1 is in the master mode (see state B0H).

If the AA bit is reset during a transfer, SIO1 will transmit the last byte of the transfer and enter state C0H or C8H. SIO1 is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1s as serial data. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^2C bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I^2C bus.

Miscellaneous States:
There are two S1STA codes that do not correspond to a defined SIO1 hardware state (see Table 24). These are discussed below.

S1STA = F8H:
This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when SIO1 is not involved in a serial transfer.

S1STA = 00H:
This status code indicates that a bus error has occurred during an SIO1 serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal SIO1 signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes SIO1 to enter the "not addressed" slave mode (a defined state) and to clear the STO flag (no other bits in S1CON are affected). The SDA and SCL lines are released (a STOP condition is not transmitted).

Some Special Cases:
The SIO1 hardware has facilities to handle the following special cases that may occur during a serial transfer:

Simultaneous Repeated START Conditions from Two Masters A repeated START condition may be generated in the master transmitter or master receiver modes. A special case occurs if another master simultaneously generates a repeated START condition (see Figure 24). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the SIO1 hardware detects a repeated START condition on the I^2C bus before generating a repeated START condition itself, it will release the bus, and no interrupt request is generated. If another master frees the bus by generating a STOP condition, SIO1 will transmit a normal START condition (state 08H), and a retry of the total serial data transfer can commence.

DATA TRANSFER AFTER LOSS OF ARBITRATION

Arbitration may be lost in the master transmitter and master receiver modes (see Figure 16). Loss of arbitration is indicated by the following states in S1STA; 38H, 68H, 78H, and B0H (see Figures 20 and 21).

If the STA flag in S1CON is set by the routines which service these states, then, if the bus is free again, a START condition (state 08H) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

FORCED ACCESS TO THE I²C BUS

In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I²C bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I²C bus is possible. This is achieved by setting the STO flag while the STA flag is still set. No STOP condition is transmitted. The SIO1 hardware behaves as if a STOP condition was received and is able to transmit a START condition. The STO flag is cleared by hardware.

I²C BUS OBSTRUCTED BY A LOW LEVEL ON SCL OR SDA

An I²C bus hang-up occurs if SDA or SCL is pulled LOW by an uncontrolled source. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the SIO1 hardware cannot resolve this type of problem. When this occurs, the problem must be resolved by the device that is pulling the SCL bus line LOW.

If the SDA line is obstructed by another device on the bus (e.g., a slave device out of bit synchronization), the problem can be solved by transmitting additional clock pulses on the SCL line(see figure 26). The SIO1 hardware transmits additional clock pulses when the STA flag is set, but no START condition can be generated because the SDA line is pulled LOW while the I²C bus is considered free. The SIO1 hardware attempts to generate a START condition after every two additional clock pulses on the SCL line. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transfer continues.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the SIO1 hardware performs the same action as described above. In each case, state 08H is entered after a successful START condition is transmitted and normal serial transfer continues. Note that the CPU is not involved in solving these bus hang-up problems.

BUS ERROR

A bus error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data or an acknowledge bit.

The SIO1 hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, SIO1 immediately switches to the not addressed slave mode, releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 00H. This status code may be used to vector to a service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in Table 24.

Table 24 : Miscellaneous States

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
F8H	No relevant state information available; SI = 0	No S1DAT action	No S1CON action				Wait or proceed current transfer
00H	Bus error during MST or selected slave modes, due to an illegal START or STOP condition. State 00H can also occur when interference causes SIO1 to enter an undefined state.	No S1DAT action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and SIO1 is switched to the not addressed SLV mode. STO is reset.

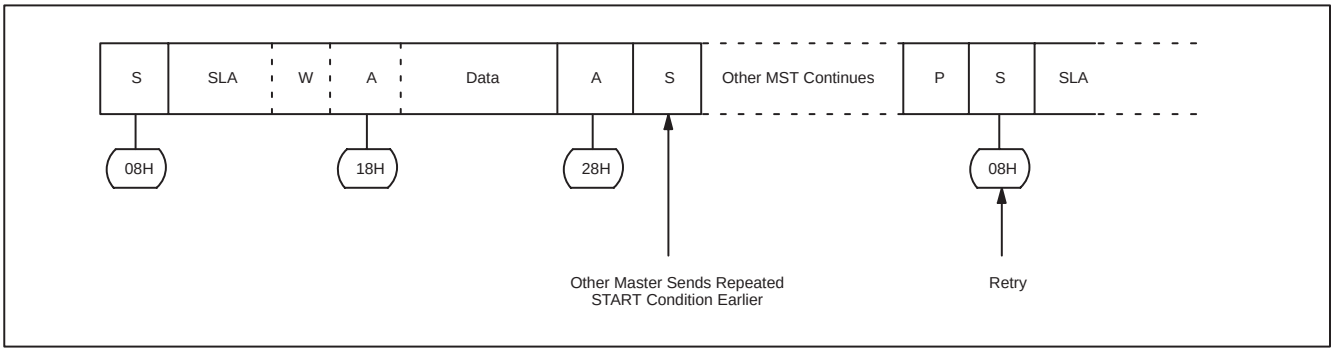


Figure 24. Simultaneous Repeated START Conditions from 2 Masters

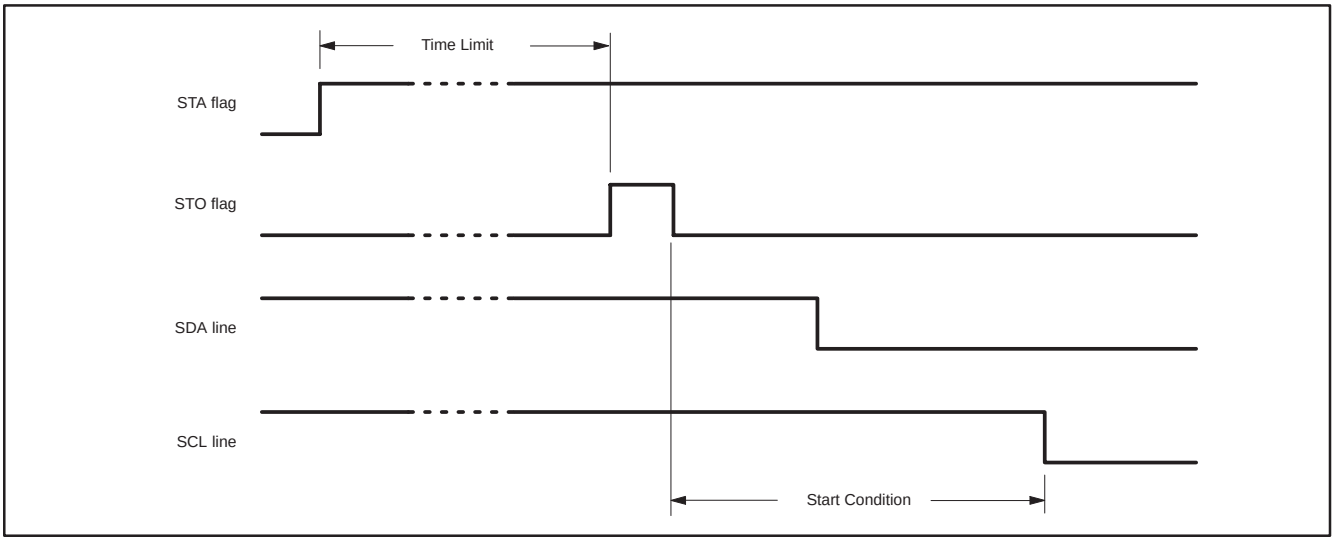


Figure 25. Forced Access to a Busy I²C Bus

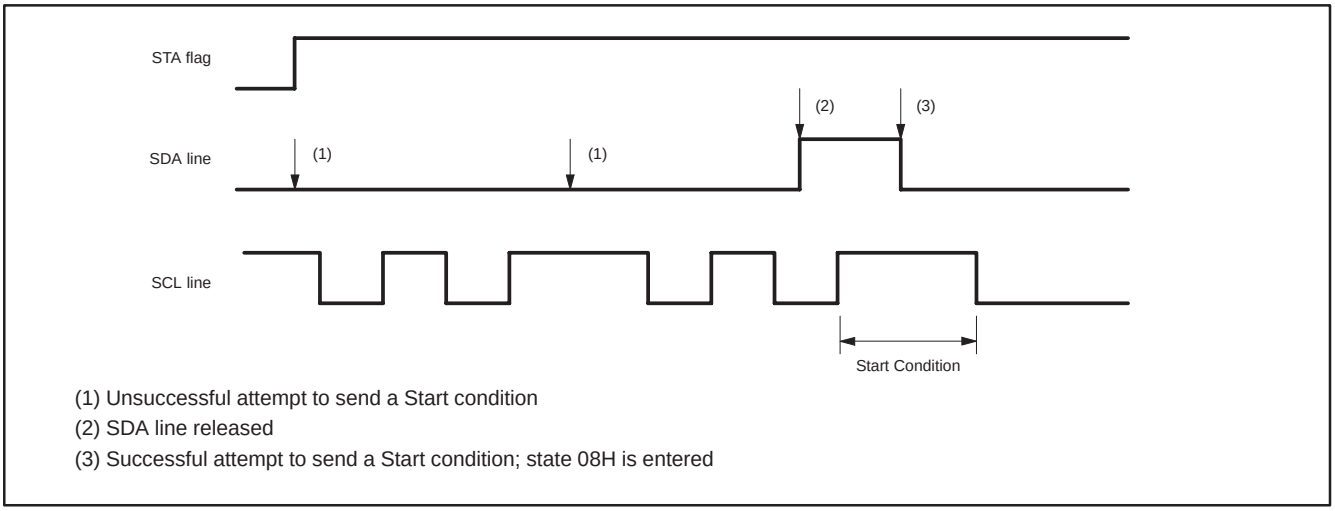


Figure 26. Recovering from a Bus Obstruction Caused by a Low Level on SDA

FLASH EPROM MEMORY

GENERAL DESCRIPTION

The MX10E8050I Serial Flash memory augments EPROM functionality with in-circuit electrical erasure and programming. (MX10E8050I) The Flash can be read and written as bytes. The Chip Erase operation will erase the entire program memory. The Block Erase function can erase any Flash byte block. In-system programming and standard parallel programming are both available for MX10E8050I. Standard parallel programming is available for MX10E8050X. On-chip erase and write timing generation contribute to a user-friendly programming interface.
The MX10E8050I Flash reliably stores memory contents even after 1,000 erase and program cycles. The cell is designed to optimize the erase and programming mechanisms. In addition, the combination of advanced tunnel oxide processing and low internal electric fields for erase and programming operations produces reliable cycling. The MX10E8050I uses a +3.3 V VPP supply to perform the Program/Erase algorithms.

FEATURES

- Flash EPROM internal program memory with Block Erase.
- Internal 2 k byte fixed boot ROM, containing low-level in-system programming routines and a default serial loader. User program can call these routines to perform In-Application Programming (IAP). The Boot ROM can be turned off to provide access to the full 64 k byte Flash memory. (MX10E8050I only)
- Boot vector allows user provided Flash loader code to reside anywhere in the Flash memory space. This configuration provides flexibility to the user. (MX10E8050I only)
- Default loader in Boot ROM allows programming via the serial port without the need for a user provided loader. (MX10E8050I only)
- Up to 64 kB external program memory if the internal program memory is disabled (EA = 0).
- Programming and erase voltage +3.3 V
    - Read/Programming/Erase:
    - Byte-width read (90 ns access time).
    - Byte Programming (50 us typically).
    - Typical erase times:
      Block Erase (16 k bytes) in 1 second.
      Full Erase (64 k bytes) in 4 seconds.
- Parallel programming with JEDEC compatible hardware interface to programmer.
- In-system programming.
- Programmable security for the code in the Flash.
- 1,000 minimum erase/program cycles for each byte.
- 10-year minimum data retention.

CAPABILITIES OF THE MX10E8050I FLASH-BASED MICROCONTROLLERS

Flash organization

The MX10E8050I Serial contains 64Kbytes of Flash program memory. This memory is organized as 4 separate blocks. Each of the blocks is 16k bytes.

Figure 27 depicts the Flash memory configurations.

Flash Programming and Erasure

MX10E8050I has three methods of erasing or programming of the Flash memory that may be used. First, the Flash may be programmed or erased in the end-user application by calling low-level routines through a common entry point in the Boot ROM. The end-user application, though, must be executing code from a different block than the block that is being erased or programmed. Second, the on-chip ISP boot loader may be invoked. This ISP boot loader will, in turn, call low-level routines through the same common entry point in the Boot ROM that can be used by the end-user application. Third, the Flash may be programmed or erased using the parallel method by using a commercially available EPROM programmer. The parallel programming method used by these devices is similar to that used by EPROM 87C51, but it is not identical, and the commercially available programmer will need to have support for these devices. MX10E8050X has parallel programming method of erasing or programming of the Flash memory.

Boot ROM (MX10E8050I)

When the microcontroller programs its own Flash memory, all of the low level details are handled by code that is permanently contained in a 2k byte Boot ROM that is separate from the Flash memory. A user program simply calls the common entry point with appropriate parameters in the Boot ROM to accomplish the desired operation. Boot ROM operations include things like: erase block, program byte, verify byte, program security lock bit, etc. The Boot ROM overlays the program memory space at the top of the address space from F800 to FFFF hex, when it is enabled. The Boot ROM may be turned off so that the upper 2k bytes of Flash program memory are accessible for execution.

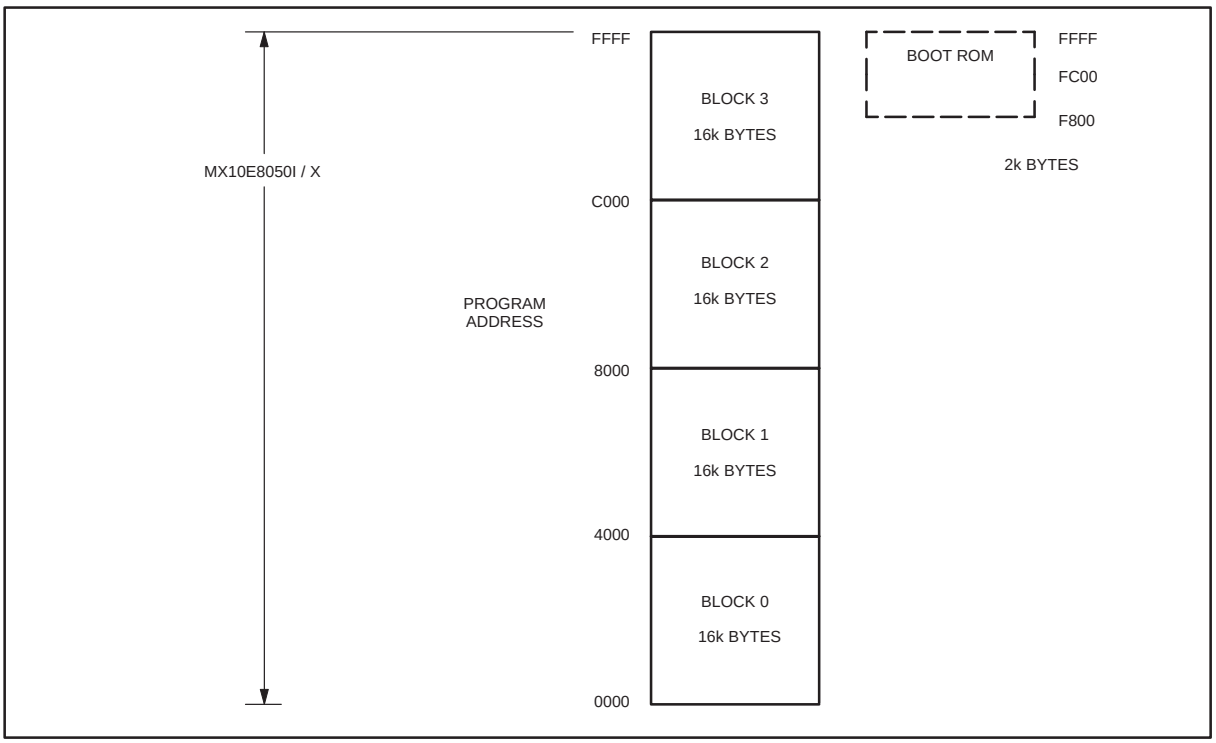


Fig 28. Flash Memory Configurations

Power-On Reset Code Execution

The MX10E8050I contains two special Flash registers: the BOOT VECTOR and the STATUS BYTE. At the falling edge of reset, the MX10E8050I examines the contents of the Status Byte. If the Status Byte is set to zero, power-up execution starts at location 0000H, which is the normal start address of the user's application code. When the Status Byte is set to a value other than zero, the contents of the Boot Vector is used as the high byte of the execution address and the low byte is set to 00H. The factory default setting is 0FCH, corresponds to the address 0FC00H for the factory masked-ROM ISP boot loader. A custom boot loader can be written with the Boot Vector set to the custom boot loader.

NOTE: When erasing the Status Byte or Boot Vector, both bytes are erased at the same time. It is necessary to reprogram the Boot Vector after erasing and updating the Status Byte.

Hardware Activation of the Boot Loader (MX10E8050I)

The boot loader can also be executed by holding PSEN LOW, $\overline{EA}$ greater than V_{IH} (such as +3.3 V), and ALE HIGH (or not connected) at the falling edge of RESET. This is the same effect as having a non-zero status byte. This allows an application to be built that will normally execute the end user' s code but can be manually forced into ISP operation.

If the factory default setting for the Boot Vector (0FCH) is changed, it will no longer point to the ISP masked-ROM boot loader code. If this happens, the only way it is possible to change the contents of the Boot Vector is through the parallel programming method, provided that the end user application does not contain a customized loader that provides for erasing and reprogramming of the Boot Vector and Status Byte.

After programming the Flash, the status byte should be programmed to zero in order to allow execution of the user' s application code beginning at address 0000H.

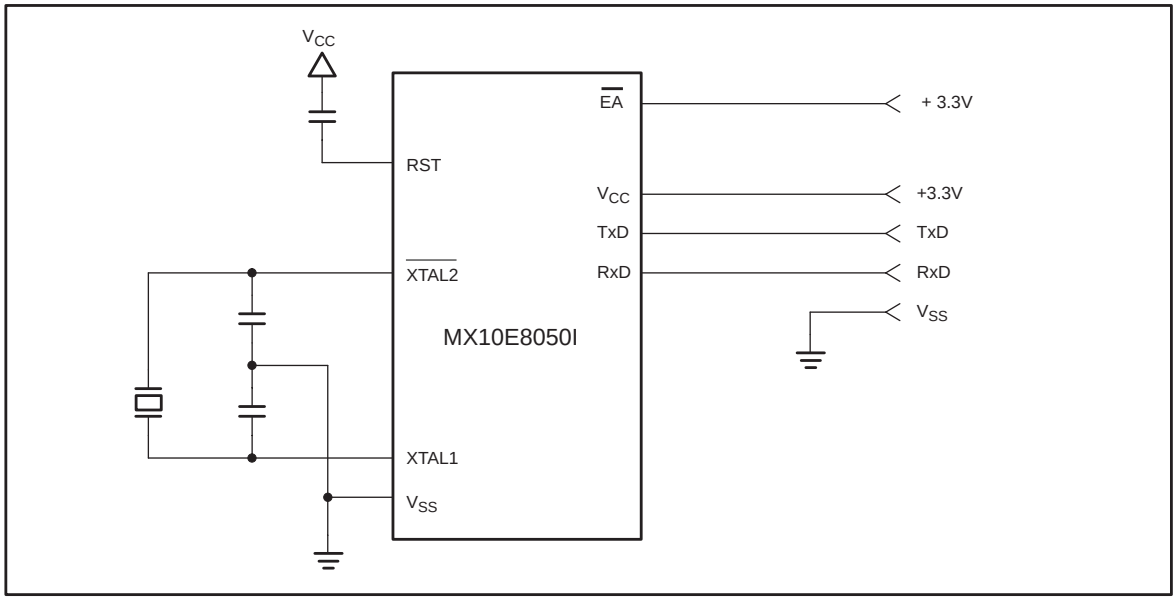


Fig 29. In-System Programming with a Minimum of Pins

In-System Programming (ISP)

* MX10E8050I : UART

The In-System Programming (ISP) is performed without removing the microcontroller from the system. The In-System Programming (ISP) facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the MX10E8050I Serial through the serial port. This firmware is provided by MXIC and embedded within each MX10E8050I Serial device.

The MXIC In-System Programming (ISP) facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area.

The ISP function uses five pins: TxD, RxD, V_{SS}, V_{CC}, and V_{PP}. Only a small connector needs to be available to interface your application to an external circuit in order to use this feature. The V_{PP} supply should be adequately decoupled and V_{PP} not allowed to exceed datasheet limits.

Using the In-System Programming (ISP) (MX10E8050I)

The ISP feature allows for a wide range of baud rates to be used in your application, independent of the oscillator frequency. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts based on the oscillator frequency. The ISP feature requires that an initial character (an uppercase U) be sent to the MX10E8050I to establish the baud rate. The ISP firmware provides auto-echo of received characters.

Once baud rate initialization has been performed, the ISP firmware will only accept Intel Hex-type records. Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below:

:NNAAAARRDD..DDCC<crlf>

In the Intel Hex record, the “NN” represents the number of data bytes in the record. The MX10E8050I will accept up to 16 (10H) data bytes. The “AAAA” string represents the address of the first byte in the record. If there are zero bytes in the record, this field is often set to 0000. The “RR” string indicates the record type. A record type of “00” is a data record. A record type of “01” indicates the end-of-file mark. In this application, additional record types will be added to indicate either commands or data for the ISP facility. The maximum number of data bytes in a record is limited to 16 (decimal). ISP commands are summarized in Table 18.

As a record is received by the MX10E8050I, the information in the record is stored internally and a checksum calculation is performed. The operation indicated by the record type is not performed until the entire record has been received. Should an error occur in the checksum, the MX10E8050I will send an “X” out the serial port indicating a checksum error. If the checksum calculation is found to match the checksum in the record, then the command will be executed. In most cases, successful reception of the record will be indicated by transmitting a “.” character out the serial port (displaying the contents of the internal program memory is an exception).

In the case of a Data Record (record type 00), an additional check is made. A “.” character will NOT be sent unless the record checksum matched the calculated checksum and all of the bytes in the record were successfully programmed. For a data record, an “X” indicates that the checksum failed to match, and an “R” character indicates that one of the bytes did not properly program.

WinISP, a software utility to implement ISP programming with a PC, is available from MXIC. Commercial serial ISP programmers are available from third parties.

Table 25 : Intel-Hex Records Used by In-System Programming

RECORD TYPE	COMMAND/DATA FUNCTION
00	Program Data :nnaaaa00dd....ddcc Where: Nn = number of bytes (hex) in record Aaaa = memory address of first byte in record dd....dd = data bytes cc = checksum Example: :10008000AF5F67F0602703E0322CFA92007780C3FD
01	End of File (EOF), no operation :xxxxxx01cc Where: xxxxxx = required field, but value is a “don’t care” cc = checksum Example: :00000001FF
02	Specify Oscillator Frequency :01xxxx02ddcc Where: xxxx = required field, but value is a “don’t care” dd = integer oscillator frequency rounded down to nearest MHz cc = checksum Example: :0100000210ED (dd = 10h = 16, used for 16.0–16.9 MHz)

RECORD TYPE	COMMAND/DATA FUNCTION
03	Miscellaneous Write Functions :nnxxxx03ffssddcc Where: nn = number of bytes (hex) in record xxxx = required field, but value is a “don’t care” 03 = Write Function ff = subfunction code ss = selection code dd = data input (as needed) cc = checksum Subfunction Code = 01 (Erase Blocks) ff = 01 ss = block code as shown below: block 0, 0k to 16k, 00H block 1, 16k to 32k, 40H block 2, 32k to 48k, 80H block 3, 48k to 64k, C0H Example: :0200000301C03A erase block 3 Subfunction Code = 04 (Erase Boot Vector and Status Byte) ff = 04 ss = don’t care Example: :020000030400F7 erase boot vector and status byte Subfunction Code = 05 (Program security bits or config bit) ff = 05 ss = 00 program security bit 1 (inhibit writing to Flash) 01 program security bit 2 (inhibit Flash verify) 02 program security bit 3 (disable external memory) 03 program config bit (6x / 12x clock mode) Example: :020000030501F5 program security bit 2 Subfunction Code = 06 (Program Status Byte or Boot Vector) ff = 06 ss = 00 program status byte 01 program boot vector Example: :030000030601FCF7 program boot vector with 0FCH Subfunction Code = 07 (Full Chip Erase) Erases all blocks, security bits, and sets status and boot vector to default values ff = 07 ss = don’t care dd = don’t care Example: :0100000307F5 full chip erase

RECORD TYPE	COMMAND/DATA FUNCTION
04	Display Device Data or Blank Check – Record type 04 causes the contents of the entire Flash array to be sent out the serial port in a formatted display. This display consists of an address and the contents of 16 bytes starting with that address. No display of the device contents will occur if security bit 2 has been programmed. Data to the serial port is initiated by the reception of any character and terminated by the reception of any character. General Format of Function 04 :05xxxx04ssssseeeeffcc Where: 05 = number of bytes (hex) in record xxxx = required field, but value is a “don’t care” 04 = “Display Device Data or Blank Check” function code ssss = starting address eeee = ending address ff = subfunction 00 = display data 01 = blank check cc = checksum Example: :0500000440004FFF0069 display 4000–4FFF
05	Miscellaneous Read Functions General Format of Function 05 :02xxxx05ffsscc Where: 02 = number of bytes (hex) in record xxxx = required field, but value is a “don’t care” 05 = “Miscellaneous Read” function code ffss = subfunction and selection code 0700 = read security bits and config bit 0701 = read status byte 0702 = read boot vector cc = checksum Example: :020000050701F1 read status byte
06	Direct Load of Baud Rate General Format of Function 06 :02xxxx06hhllcc Where: 02 = number of bytes (hex) in record xxxx = required field, but value is a “don’t care” 06 = ”Direct Load of Baud Rate” function code hh = high byte of Timer 2 ll = low byte of Timer 2 cc = checksum Example: :02000006F500F3

* MX10E8050IA : I²C

The In-System Programming (ISP) is performed without removing the microcontroller from the system. The In-System Programming (ISP) facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the MX10E8050IA Serial through the serial port. This firmware is provided by MXIC and embedded within each MX10E8050IA Serial device.

The MXIC In-System Programming (ISP) facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area. The ISP function uses five pins: SDA, SCL, V_{SS}, V_{CC}, and V_{PP}. The V_{PP} supply should be adequately decoupled and V_{PP} not allowed to exceed datasheet limits.

WinISP, a software utility to implement ISP programming with a PC, is available from MXIC. Commercial serial ISP programmers are available from third parties. WinISP is the master and the MX10E8050IA is the slave in ISP through IIC. The default device address word of MX10E8050IA is 0x26 and the slave address performs on initialization. The slave address can be changed using programmer or calling IAP.

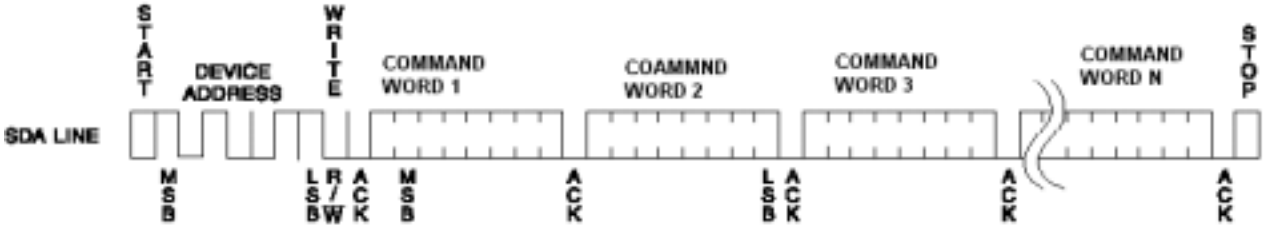
A write sequence requires some command words, summarized in Table 26, following the device address word and acknowledgment. Upon receipt of this address, the MX10E8050IA will respond with a zero and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the MX10E8050IA will output a zero. The MX10E8050IA, such as WinISP, then must terminate the write sequence with a stop condition. At this time the MX10E8050IA interprets the received command words. The MX10E8050IA will not respond acknowledgment until programming or erasing FlashROM is complete.

Once the programming or erasing has started and the MX10E8050IA inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the programming or erasing has completed will the MX10E8050IA respond with a zero, allowing the read or write sequence to continue.

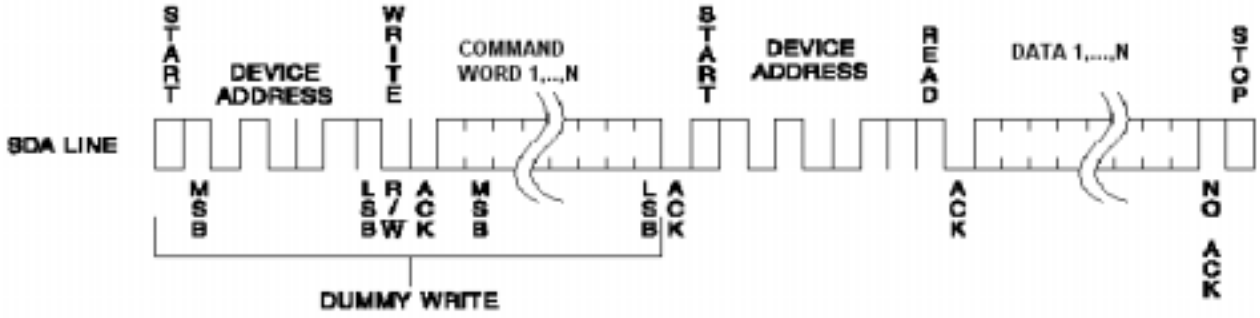
A read sequence are initiated the same way as write sequence with the exception that the read/write select bit in the device address word is set to one. A command read requires a “dummy” byte write sequence to load in the command words. Once the device address word and command words are clocked in and acknowledged by MX10E8050IA. WinISP now initiates a command read by sending device address with the read/write select bit high. The MX10E8050IA acknowledges the device address and serially clocks out the data word. The WinISP does not respond with a zero but does generate a following stop condition.

Sequential reads are initiated a read sequence. After WinISP receives a data word, it responds with an acknowledge. As long as MX10E8050IA receives an acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will “roll over” and the sequential read will continue. The sequential read operation is terminated when the WinISP does not respond with a zero but does generate a following stop condition.

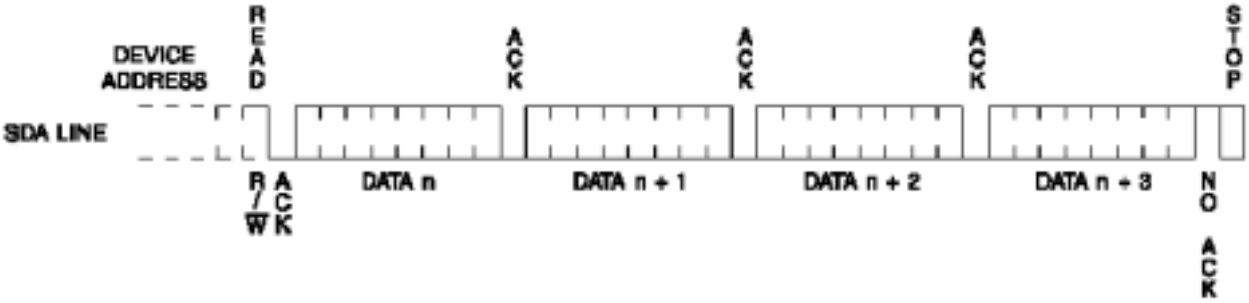
Write sequence



Read sequence



Sequential Read



Device Address : **0x26**
CWn : Command Word

Table 26 : I²C Command Table

Command Type

Commnad 1 ;Read device parameters (status)

CW1=0x01
Data1=I/F status
Data2=SB
Data3=BV
Data 4=security bits and config bit
Data 5=iic slave address
Data 6=FAILH (Hi-Byte Fail address)
Data 7=FAILL (Lo-Byte Fail address)

Commnad 2 ;Read device data

CW1=0x02
CW2=Hi-Byte of address
CW3=Lo-Byte of address
Data n=DATAn
Notes: If lock2 was programmed, users are inhibited reading data and then 0xFF was sent.

Commnad 3 ;Blank check

CW1=0x03
CW2=Hi-Byte of start address
CW3=Lo-Byte of start address
CW4=Hi-Byte of end address
CW5=Lo-Byte of end address
The result is in I/F STATUS.2 (BLANK), I/F STATUS.3 (LOCKED) and the first non-blank address is recorded in FAILH (Hi-Byte address) and FAILL (Lo-Byte address). If the range of start address and end address is non-blank, the BLANK will be set. And if the Lock bits is programmed, the LOCKED will be set.

Commnad 4 ;Program security bits and config

CW1=0x04
CW2=0x01 for lock1
0x02 for lock2
0x04 for lock3
0x80 for config bit
The result is in I/F STATUS.0 (PGFAIL). If the programming is failed, the PGFAIL will be set.

Commnad 5 ;Program status byte, boot vectors and iic slave address

CW1=0x05
CW2=0x00 for status byte
0x01 for boot vector
0x02 for iic salve address
CW3= data for program
The result is in I/F STATUS.0 (PGFAIL) and I/F STATUS.3 (LOCKED). If the programming is failed, the PGFAIL will be set. And if the Lock bits is programmed, the LOCKED will be set.

Commnad 6 **;Erase blocks**

CW1=0x06
CW2=0x00 for block 0
0x40 for block 1
0x80 for block 2
0xC0 for block 3
The result is in I/F STATUS.1 (ERFAIL) and I/F STATUS.3 (LOCKED). If erasing is failed, the ERFAIL will be set. And if the Lock bits is programmed, the LOCKED will be set.

Commnad 7 **;Program device data**

CW1=0x07
CW2=Hi-Byte of start address
CW3=Lo-Byte of start address
CW4~CW19=data (the max of data is 16 bytes)
The result is in I/F STATUS.0 (PGFAIL), I/F STATUS.3(LOCKED) and the fail address is recorded FAILH (Hi-Byte of address) and FAILL (Lo-Byte of address). If the programming is failed, the PGFAIL will be set. And if the Lock bits is programmed, the LOCKED will be set.

Commnad 8 **;Erase status byte, boot vectors**

CW1=0x08
The result is in I/F STATUS.1 (ERFAIL) and I/F STATUS.3 (LOCKED). If erasing is failed, the ERFAIL will be set. And if the Lock bits is programmed, the LOCKED will be set.

Commnad 9 **;Full chip erase**

CW1=0x09
The result is in I/F STATUS.1 (ERFAIL). If erasing is failed, the ERFAIL will be set.

I/F status (interface status)

```
PGFAIL    EQU IFSTA.0    ;programming fail if set = 1
ERFAIL    EQU IFSTA.1    ;erase fail if set = 1
BLANK     EQU IFSTA.2    ;blank check fail if set = 1
LOCKED    EQU IFSTA.3    ;security bit violation if set = 1
RCVR      EQU IFSTA.4    ;rcvr/xmtr flag, rcvr if set = 1
```

*I/F status will update every erase or program command. After erasing or programming complete, WinISP should read the I/F status to know the result. RCVR flag will be set only execute command 1.

In Application Programming Method (MX10E8050I)

Several In Application Programming (IAP) calls are available for use by an application program to permit selective erasing and programming of Flash sectors. All calls are made throguh a common interface, PGM_MTP. the programming functions are selected by setting up the microcontroller's registers before making a call to PGM_MTP at FFF0H. The IAP calls are shown in Table 27.

Notes : Interrupts and the Watchdog Timer must be disabled while IAP subroutines are executing.

ROM enable security code Register/PDCON(F8H)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

To execute IAP or to enter ISP by software setting, this ROM enable security code register must be written to 5Ah. Only when the value of this register is 5Ah, user can set ENBOOT bit in AUXR1 register. In conclusion, to software enable ROM user must write 5Ah to PDCON and then set ENBOOT bit in AUXR1.

```
Example : MOV  PDCON, #0x5AH
          ORL  AUXR1, #0x20H
```

Remember to turn off ROM after executing IAP commands.

```
Example : ANL  AUXR1, #0xDFH
          ANL  PDCON, #0x00H
```

AUXR1 (A2H)

		ENBOOT			0		DPS

ENBOOT : This bit determines the BOOTROM is enabled or disabled.
This bit will automatically be set if the status bytes is not zero during reset or entering ISP pin setting mode. Note this bit is cleared by s/w only.
Bit2 : Bit2 is not writable and alleyways read as a zero.
DPS : Switch between DPTR0 and DPRT1.

Table 27 : IAP Calls

IAP CALL	PARAMETER
ISP MODE CHECK	Input Parameters: R1 = 00h Return Parameter: ACC = 5Ah if pass, but ACC != 5Ah if fail
PROGRAM DATA BYTE	Input Parameters: R1 = 02h DPTR = address of byte to program ACC = byte data to program Return Parameter: ACC = 00 if pass, but ACC != 00 if fail
ERASE BLOCK	Input Parameters: R1 = 01h DPH = block code as shown below: Block 0, 0k to 16K, 00h Block 1, 16k to 32k, 40h Block 2, 32k to 48k, 80h Block 3, 48k to 64k, C0h DPL = 00h Return Parameter: ACC = 00 if pass, but ACC != 00 if fail
ERASE BOOT VECTOR and STATUS BYTE	Input Parameters: R1 = 04h DPH = 00h DPL = don't care Return Parameter: ACC = 00 if pass, but ACC != 00 if fail
PROGRAM SECURITY BIT and CONFIG BIT	Input Parameters: R1 = 05h DPH = 00h DPL = 00h – security bit # 1 (inhibit writing to Flash) 01h – security bit # 2 (inhibit Flash verify) 02h – security bit # 3 (inhibit external memory) 03h – config bit (6/12 clock mode) Return Parameter: ACC = 00 if pass, but ACC != 00 if fail
PROGRAM STATUS BYTE	Input Parameters: R1 = 06h DPH = 00h DPL = 00h – program status byte ACC = data of status byte Return Parameter: ACC = 00 if pass, but ACC != 00 if fail

PROGRAM BOOT VECTOR	Input Parameters: R1 = 06h DPH = 00h DPL = 01h – program boot vector ACC = data of boot vector Return Parameter: ACC = 00 if pass, but ACC != 00 if fail
PROGRAM I^2C SLAVE ADDRESS	Input Parameters: R1 = 06h DPH = 00h DPL = 02h – program i^2c slave address ACC = data of boot vector Return Parameter: ACC = 00 if pass, but ACC != 00 if fail
READ DEVICE DATA	Input Parameters: R1 = 03h DPTR = address of byte to read Return Parameter: ACC = value of byte read
READ SECURITY BITS and CONFIG BIT	Input Parameters: R1 = 07h DPH= 00h DPL = 00h (security bits) Return Parameter: ACC = value of byte read
READ STATUS BYTE	Input Parameters: R1 = 07h DPH= 00h DPL = 01h (status byte) Return Parameter: ACC = value of byte read
READ BOOT VECTOR	Input Parameters: R1 = 07h DPH= 00h DPL = 02h (boot vector) Return Parameter: ACC = value of byte read
READ I2C SLAVE ADDRESS	Input Parameters: R1 = 07h DPH= 00h DPL = 03h (i^2c slave address) Return Parameter: ACC = value of byte read

Security

The security feature protects against software piracy and prevents the contents of the Flash from being read. The Security Lock bits are located in Flash. The MX10E8050I Serial has 3 programmable security lock bits that will provide different levels of protection for the on-chip code and data (see Table 28).

Table 28

	SECURITY LOCK BITS1			
Level	LB1	LB2	LB3	PROTECTION DESCRIPTION
1	0	0	0	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory.
2	1	0	0	Program and block erase is disabled. Erase or programming of the status byte or boot vector is disabled.
3	1	1	0	Verify of code memory is disabled.
4	1	1	1	External execution is disabled.

NOTE :

1. Security bits are independent of each other. Full-chip erase may be performed regardless of the state of the security bits.
2. Any other combination of lock bits is undefined.

CONFIG :

12-clock or 6-clock mode configuration bit is saved in flash special cell CONFIG. The address of CONFIG bit is the same as security bits, so do their program or erase algorithm. CONFIG bit can be normally programmed but can be erased by chip erased only. Defaultly CONFIG bit is clear for MX10E8050I, that means 12 clock mode. If CONFIG bit is programmed to HIGH, 6-clock mode is enabled when RESET goes low. Note that when programming CONFIG to 6-clock mode by ISP, the chip would not immediately change to 6-clock mode until another RESET going low. MX10E8050X is defaultly 6-clock mode.

LOCK function table

The security bits LOCK 1~3 could prevent form illegal writing or reading at ISP mode or external programming mode. The detail security function table is shown as below:

		Parallel Programming		
		LOCK1	LOCK2	LOCK 3
Read Array			X	
Read Special Cell				
Read ID				
Program Array		X		
Program Special Cell	LOCK			
	I^2C Address			
	SBYTE	X		
	BVEC	X		
Erase Array		X		
Erase Special Cell	LOCK	X	X	X
	I^2C Address			
	SBYTE	X		
	BVEC	X		
Chip Erase				

ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Operation temperature	0 to +70	^{o}C
Storage temperature range	- 65 to +150	^{o}C
Voltage on Xtal1, $\overline{\text{Xtal2}}$ pin to V_{SS}	V_{DD}+0.5	V
Maximum I_{OL} per I/O pin	15	mA
Power dissipation (based on package heat transfer, not device power consumption)	1.5	W

Notes:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification are not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maximum.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

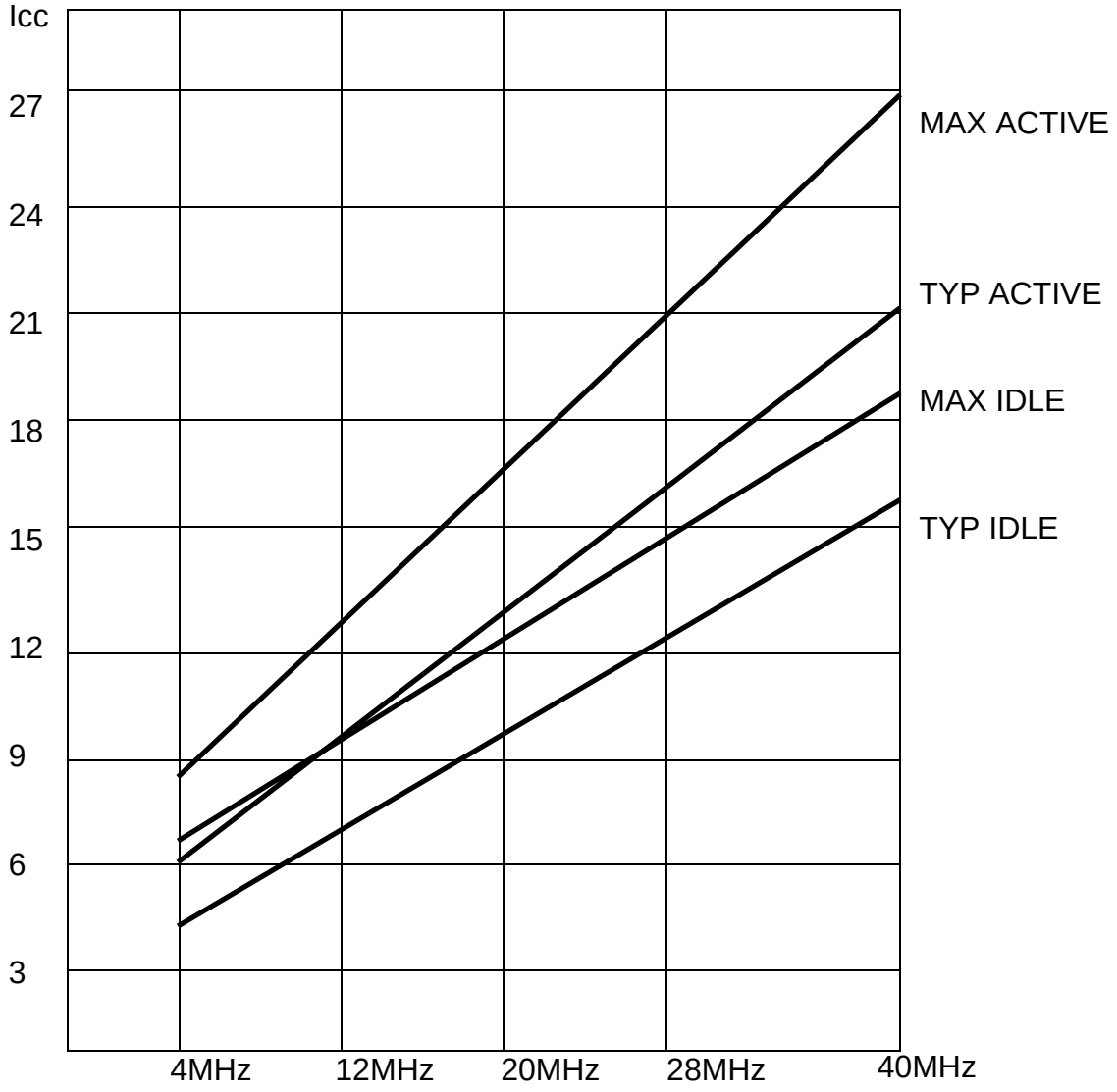
DC ELECTRICAL CHARACTERISTICS

V_{DD} = 3.0 V to 3.6 V unless otherwise specified;
T_{amb} = 0°C to +70°C for commercial for industrial unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
I_{DD}	Power supply current, operating	3.6 V, 40 MHz[11]	-	15	30	mA
I_{ID}	Power supply current, Idle mode	3.6 V, 40 MHz[11]	-	10	20	mA
I_{PD1}	Power supply current, Total Power-down mode	3.0 V [11]	-	10		uA
V_{DDR}	Vdd rise time		-	-	2	mV/us
V_{DDF}	Vdd fall time		-	-	50	mV/us
V_{RAM}	RAM keep-alive voltage		1.5	-	-	V
V_{IL}	Input low voltage (TTL input)	2.4 V < V_{DD} < 3.6 V	-0.5	-	$0.22V_{DD}$-0.1	V
V_{IH}	Input high voltage (TTL input)		$0.7V_{DD}$+0.1	-	5.5	V
V_{OL}	Output low voltage all ports [5, 9]	I_{OL} = 20mA; V_{DD} = 2.4 V	-	-	1.0	V
		I_{OL} = 3.2mA; V_{DD} = 2.4 V	-	-	0.3	V
V_{OH}	Output high voltage, all ports [3]	I_{OH} =- 20uA; V_{DD} = 2.4 V	V_{DD}-0.2	-	-	V
C_{IO}	Input/Output pin capacitance [10]		-	-	15	pF
I_{IH}	Logical 1 input current, all ports [8]	V_{IN} = 3.3 V	-	-	-50	uA
I_{LI}	Input leakage current, all ports [7]	V_{IN} = V_{IL} or V_{IH}	-	-	±30	uA
I_{TL}	Logical 1-to-0 transition current, all ports [3, 6]	V_{IN} = 1.5 V at V_{DD} = 3.6 V	-30	-	-250	uA
R_{RST}	Internal reset pull-up resistor		40	-	225	k ohm

Notes:
1. Typical ratings are not guaranteed. The values listed are at room temperature, 3.3 V.
2. Active mode: $I_{CC(MAX)}$ = 30mA
   Idle mode: $I_{CC(MAX)}$ = 20mA
3. Ports in quasi-bidirectional mode with weak pull-up (applies to all port pins with pull-ups). Does not apply to open drain pins.
4. Ports in PUSH-PULL mode. Does not apply to open drain pins.
5. In all output modes except high impedance mode.
6. Port pins source a transition current when used in quasi-bidirectional mode and externally driven from 1 to 0. This current is highest when V_{IN} is approximately 2 V.
7. Measured with port in high impedance mode.
8. Measured with port in quasi-bidirectional mode.
9. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 15 mA
   Maximum total I_{OL} for all outputs: 26 mA
   Maximum total I_{OH} for all outputs: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
10. Pin capacitance is characterized but not tested.
11. The I_{DD} and I_{ID} specifications are measured using an external clock. This is 12 clock mode.

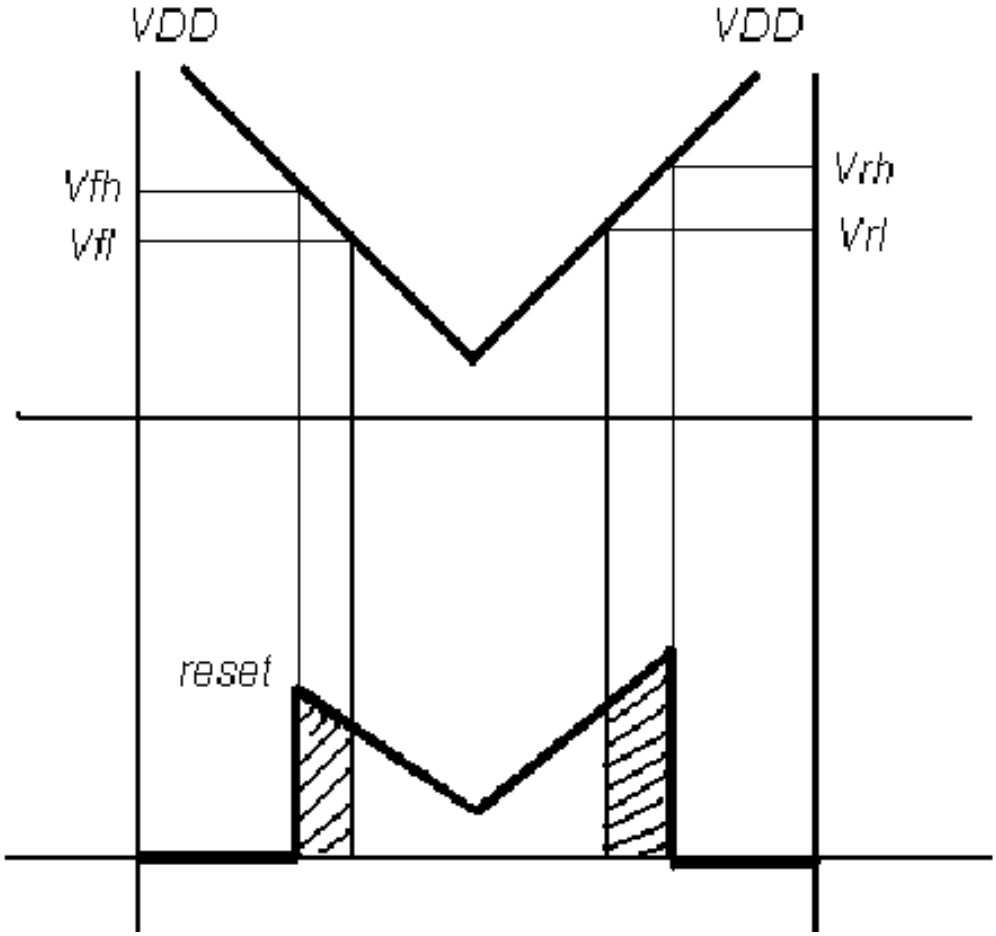
ICC_VS freq


Icc
27
24
21
18
15
12
9
6
3
4MHz
12MHz
20MHz
28MHz
40MHz
MAX ACTIVE
TYP ACTIVE
MAX IDLE
TYP IDLE

Low Voltage Detector (MX10E8050I)

This low voltage detector will reset the chip when detecting a V_{DD} lower than a designed level. The reset status remains till V_{DD} rise to normal operating voltage. Since the reset shall keeps at lease two machine cycle, the V_{DD} low to V_{DD} high shall not transit sooner than two machine cycle.

Detecting level	Min	Max
V_{DD} falling	2.40 V	2.60 V
V_{DD} rising	2.43 V	2.65 V



AC CHARACTERISTICS

(Over Operating Conditions, Load Capacitance for Port 0, ALE/PROG and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

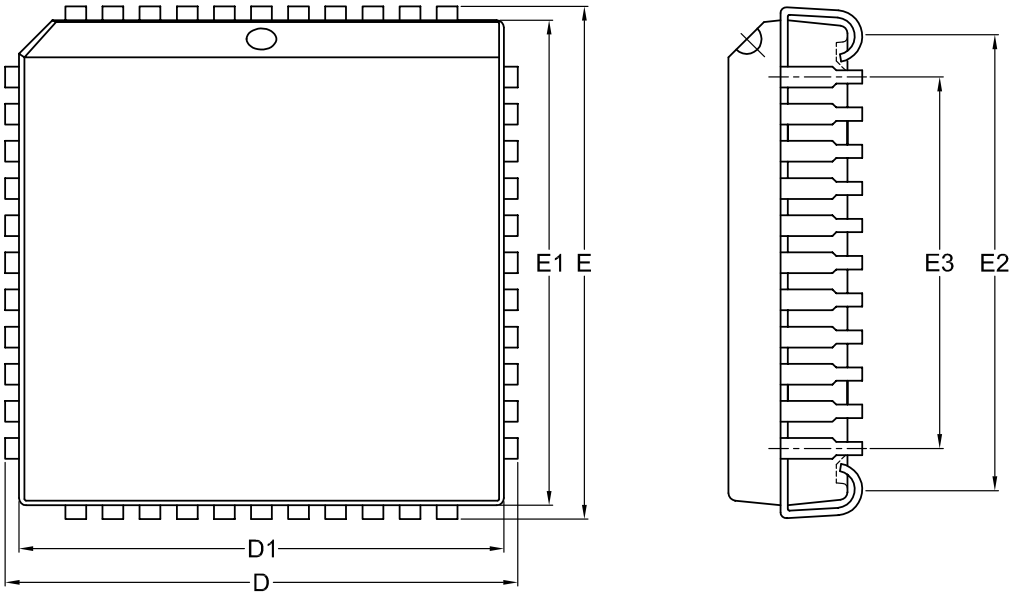
tCK min. = 1/f max. (maximum operating frequency); tCK=clock period

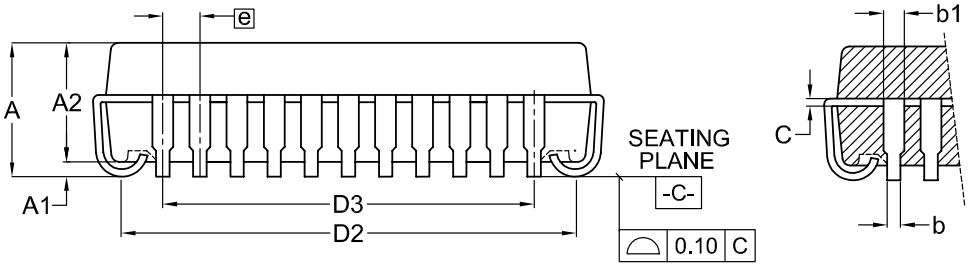
SYMBOL	PARAMETER	40 MHz, x12 mode		UNIT
		MIN	MAX	
EXTERNAL PROGRAM MEMORY				
TLHLL	ALE PULSE DURATION	20	-	NS
TAVLL	ADDRESS SET-UP TIME TO ALE	17	-	NS
TLLAX	ADDRESS HOLD TIME AFTER ALE	10	-	NS
TLLIV	TIME FROM ALE TO VALID INSTRUCTION INPUT	-	55	NS
TLLPL	TIME FROM ALE TO CONTROL PULSE $\overline{PSEN}$	17	-	NS
TPLPH	CONTROL PULSE DURATION $\overline{PSEN}$	70	-	NS
TPLIV	TIME FROM PSEN TO VALID INSTRUCTION INPUT	-	12	NS
TPXIX	INPUT INSTRUCTION HOLD TIME AFTER $\overline{PSEN}$	0	-	NS
TPXIZ	INPUT INSTRUCTION FLOAT DELAY AFTER $\overline{PSEN}$	-	20	NS
TAVIV	ADDRESS TO VALID INSTRUCTION INPUT	-	95	NS
TPLAZ	TO $\overline{PSEN}$ ADDRESS FLOAT TIME	-	10	NS
EXTERNAL DATA MEMORY				
TLHLL	ALE PULSE DURATION	20	-	NS
TAVLL	ADDRESS SET-UP TIME TO ALE	17	-	NS
TLLAX	ADDRESS HOLD TIME AFTER ALE	10	-	NS
TRLRH	RD PULSE DURATION	80	-	NS
TWLWH	WR PULSE DURATION	80	-	NS
TRLDV	RD TO VALID DATA INPUT	-	60	NS
TRHDX	DATA HOLD TIME AFTER RD	0	-	NS
TRHDZ	DATA FLOAT DELAY AFTER RD	32	-	NS
TLLDV	TIME FROM ALE TO VALID DATA INPUT	-	90	NS
TAVDV	ADDRESS TO VALID INPUT	-	105	NS
TLLWL	TIME FROM ALE TO RD OR WR	40	140	NS
TAVWL	TIME FROM ADDRESS TO RD OR WR	45	-	NS
TWHLH	TIME FROM RD OR WR HIGH TO ALE HIGH	10	55	NS
TQVWX	DATA VALID TO WR TRANSITION	10	-	NS
TQVWH	DATA SET-UP TIME BEFORE WR	125	-	NS
TWHQX	DATA HOLD TIME AFTER WR	10	-	NS
TRLAZ	ADDRESS FLOAT DELAY AFTER RD	-	0	NS

NOTE:
1. The maximun operating frequency is limited to 40 MHz and the minimum to 3.5 MHz.

PLCC44

Title: **Package Outline for 44L PLCC**





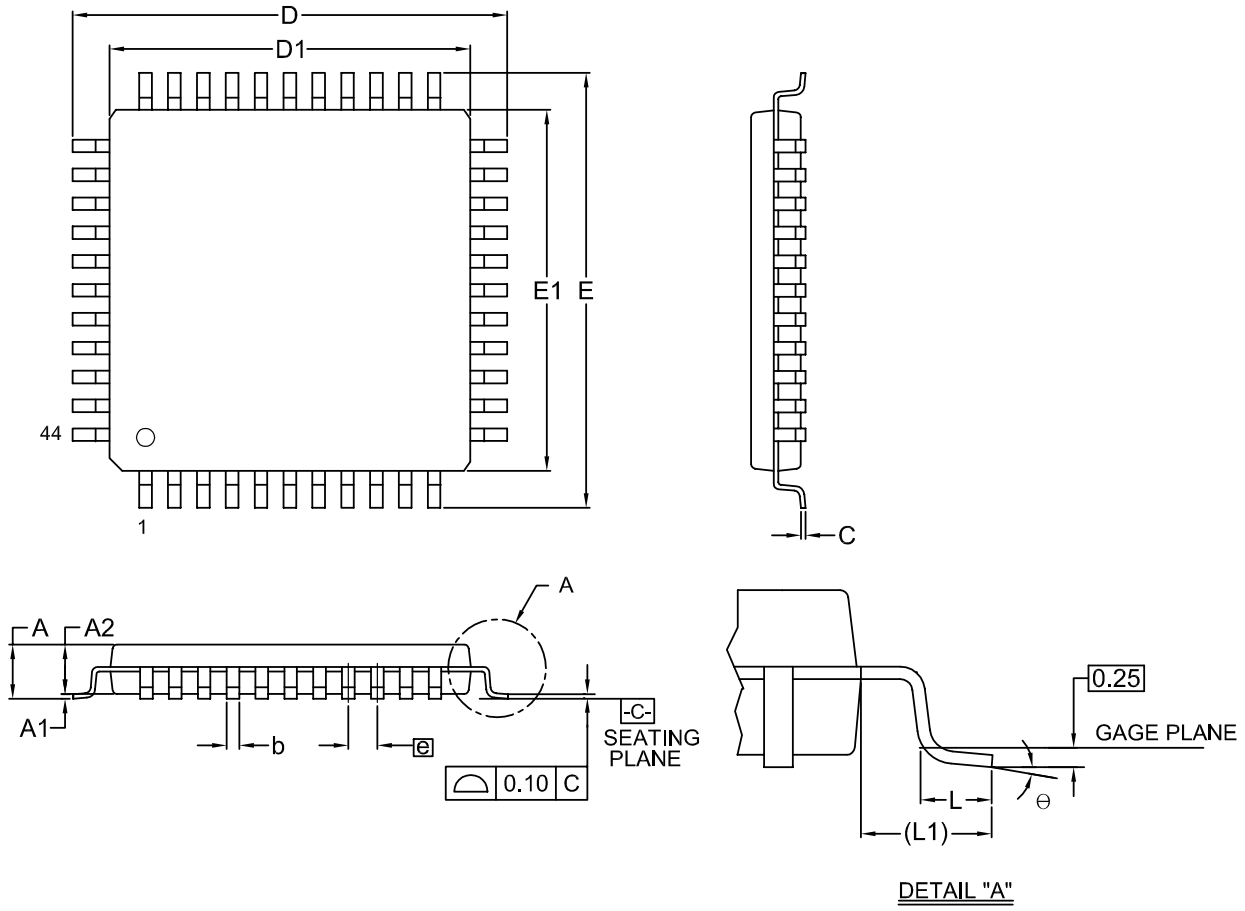
Dimensions (inch dimensions are derived from the original mm dimensions)

UNIT \ SYMBOL		A	A1	A2	b	b1	C	D	D1	D2	D3	E	E1	E2	E3	e
mm	Min.	---	0.38	3.71	0.38	0.61	0.20	17.40	16.52	15.19		17.40	16.52	15.19		
	Nom.	---	0.50	3.81	0.46	0.71	0.25	17.53	16.59	15.49	12.70	17.53	16.59	15.49	12.70	1.27
	Max.	4.57	0.66	3.91	0.54	0.81	0.30	17.66	16.66	15.79		17.66	16.66	15.79		
Inch	Min.	---	0.015	0.146	0.015	0.024	0.008	0.685	0.650	0.598		0.685	0.650	0.598		
	Nom.	---	0.020	0.150	0.018	0.028	0.010	0.690	0.653	0.610	0.500	0.690	0.653	0.610	0.500	0.050
	Max.	0.180	0.026	0.154	0.021	0.032	0.012	0.695	0.656	0.622		0.695	0.656	0.622		

DWG.NO.	REVISION	REFERENCE			ISSUE DATE
		JEDEC	EIAJ		
6110-2003	4	MS-016			09-24-'02

LQFP44

Title: **Package Outline for LQFP 44L (10X10X1.4mm)**



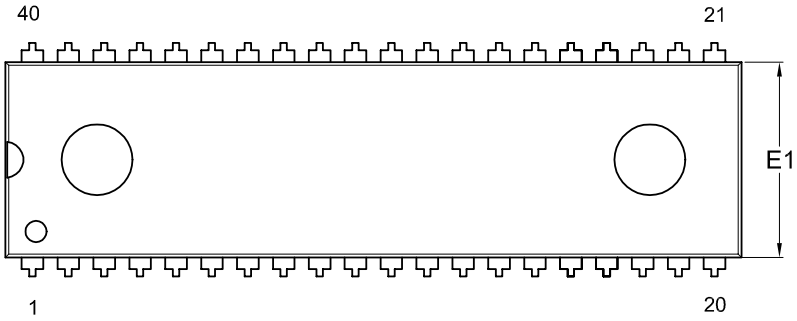
Dimensions (inch dimensions are derived from the original mm dimensions)

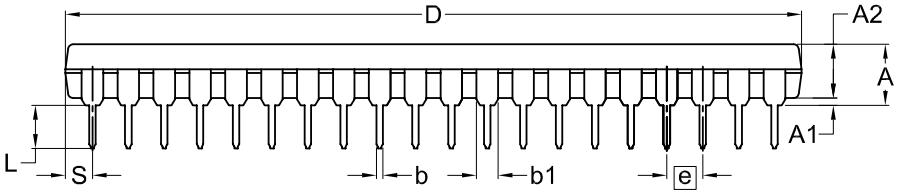
UNIT \ SYMBOL		A	A1	A2	b	C	D	D1	E	E1	e	L	L1	Θ
mm	Min.	---	0.05	1.35	0.22	0.09	11.80	9.90	11.80	9.90		0.45	0.85	0
	Nom.	---	0.10	1.40	0.30	0.13	12.00	10.00	12.00	10.00	0.80	0.60	1.00	3.5
	Max.	1.60	0.15	1.45	0.38	0.20	12.20	10.10	12.20	10.10		0.75	1.15	7
Inch	Min.	---	0.002	0.053	0.009	0.004	0.465	0.390	0.465	0.390		0.018	0.033	0
	Nom.	---	0.004	0.055	0.012	0.005	0.472	0.394	0.472	0.394	0.031	0.024	0.039	3.5
	Max.	0.063	0.006	0.057	0.015	0.008	0.480	0.398	0.480	0.398		0.030	0.045	7

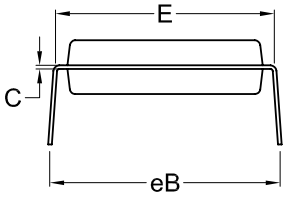
DWG.NO.	REVISION	REFERENCE			ISSUE DATE
		JEDEC	EIAJ		
6110-1202	5	MS-026			07-15-'02

PDIP40

Title: **Package Outline for PDIP 40L(600MIL)**







Dimensions (inch dimensions are derived from the original mm dimensions)

UNIT \ SYMBOL		A	A1	A2	b	b1	C	D	E	E1	e	eB	L	S
mm	Min.	---	0.25	3.73	0.38	1.14	0.20	51.31	15.11	13.84		15.75	2.92	1.65
	Nom.	---	---	3.94	0.46	1.27	0.25	51.94	15.24	13.97	2.54	16.51	3.30	1.90
	Max.	4.90	0.76	4.14	0.53	1.40	0.30	52.57	15.37	14.10		17.27	3.68	2.16
Inch	Min.	---	0.010	0.147	0.015	0.045	0.008	2.020	0.595	0.545		0.620	0.115	0.065
	Nom.	---	---	0.155	0.018	0.050	0.010	2.045	0.600	0.550	0.100	0.650	0.130	0.075
	Max.	0.193	0.030	0.163	0.021	0.055	0.012	2.070	0.605	0.555		0.680	0.145	0.085

DWG.NO.	REVISION	REFERENCE			ISSUE DATE
		JEDEC	EIAJ		
6110-0202.4	7				01-20-'03

REVISION HISTORY

REVISION	DESCRIPTION	DATE
1.1	- Addition I^2C pin release	Jan / 29 / 2003
	- Pin Description release	
	- Addition internal Data memory Sample code release	
	- Explain Special function registers	
	- Boot ROM release	
	- ISP release	
1.2	- Addition Programming spec.	Jun / 24 / 2003
	- Addition I^2C / UART description	
1.3	- Update I^2C, UART function	Jul / 21 / 2003
	- Update page 10 symbol IE / IP / IPH volume	
	- Addition MX10E8050IA (ISP + I^2C) function	

MACRONIX INTERNATIONAL CO., LTD.

HEADQUARTERS:
TEL:+886-3-578-6688
FAX:+886-3-563-2888

EUROPE OFFICE:
TEL:+32-2-456-8020
FAX:+32-2-456-8021

JAPAN OFFICE:
TEL:+81-44-246-9100
FAX:+81-44-246-9105

SINGAPORE OFFICE:
TEL:+65-348-8385
FAX:+65-348-8096

TAIPEI OFFICE:
TEL:+886-2-2509-3300
FAX:+886-2-2509-2200

MACRONIX AMERICA, INC.

TEL:+1-408-453-8088
FAX:+1-408-453-8488

CHICAGO OFFICE:
TEL:+1-847-963-1900
FAX:+1-847-963-1909

http : //www.macronix.com

MACRONIX INTERNATIONAL CO., LTD. reserves the right to change product and specifications without notice.