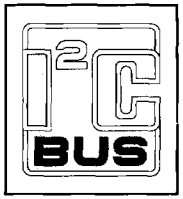
Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



GENERAL DESCRIPTION

The 80CE558/83CE558/89CE558 (hereafter generically referred to as 8XCE558) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XCE558 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83CE558 — 32k bytes mask programmable ROM
- 80CE558 — ROMless version of the 83CE558
- 89CE558 — 32k bytes FEEPROM

The 8XCE558 contains a non-volatile 32k × 8 read-only program memory (83CE558) or FEEPROM (89CE558), a volatile 1024 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority- level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I²C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XCE558 can be expanded using standard TTL compatible memories and logic.

In addition, the 8XCE558 has two software selectable modes of power reduction — idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16 MHz system clock, 58% of the instructions are executed in 0.75µs and 40% in 1.5µs. Multiply and divide instructions require 3µs.

FEATURES

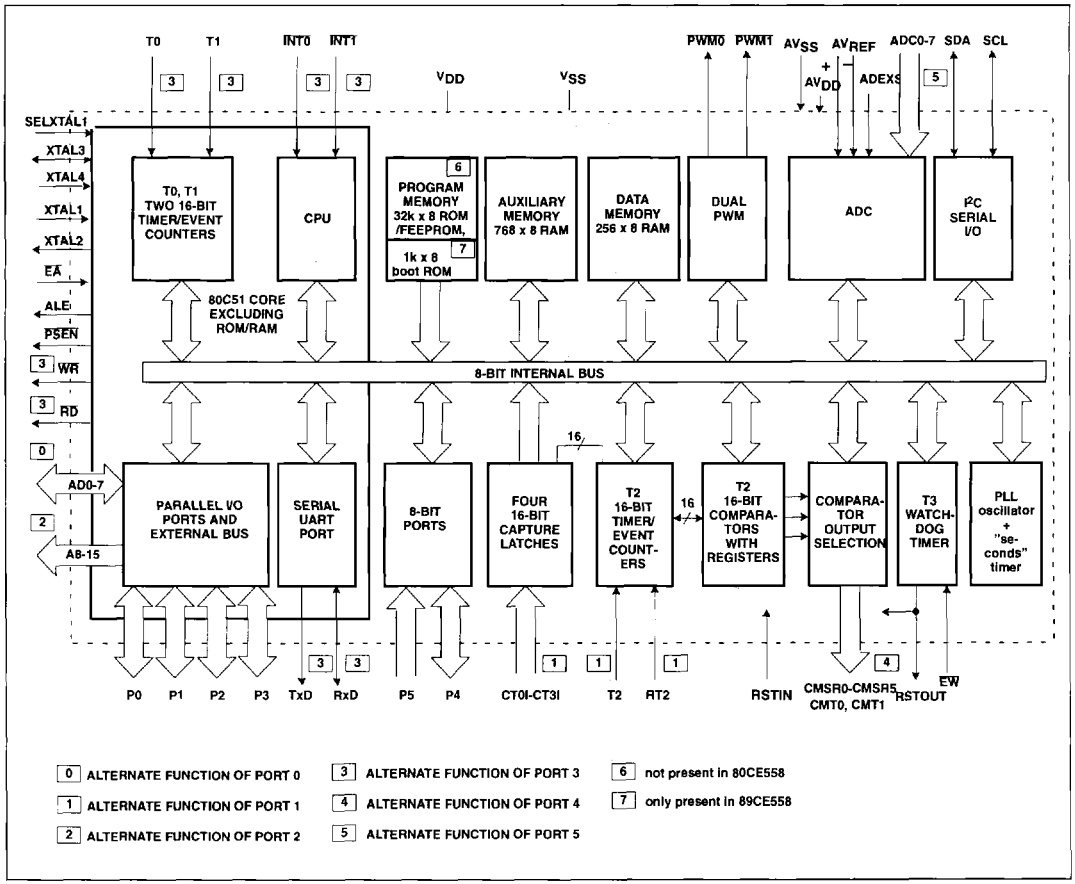
- 80C51 central processing unit
- 32k × 8 ROM resp. FEEPROM expandable externally to 64k bytes
- ROM/FEEPROM Code protection
- 1024 × 8 RAM, expandable externally to 64k bytes
- Seconds Timer
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- A 10-bit ADC with eight multiplexed analog inputs and programmable autoscan
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with anatog inputs
- I²C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- 15 interrupt sources with 2 priority levels (2 to 6 external sources possible)
- Extended temperature range (–40 to +85°C)
- 4.5 to 5.5V supply voltage range
- Frequency range for 80C51 standard oscillator: 3.5 MHz to 16 MHz
- PLL oscillator with 32 kHz reference and software-selectable system clock frequency[1]
- Software enable/disable of ALE output pulse
- Electromagnetic compatibility improvements
- Wake-up from Power-down by external or seconds interrupt

Note 1 Not available yet in the 89CE558

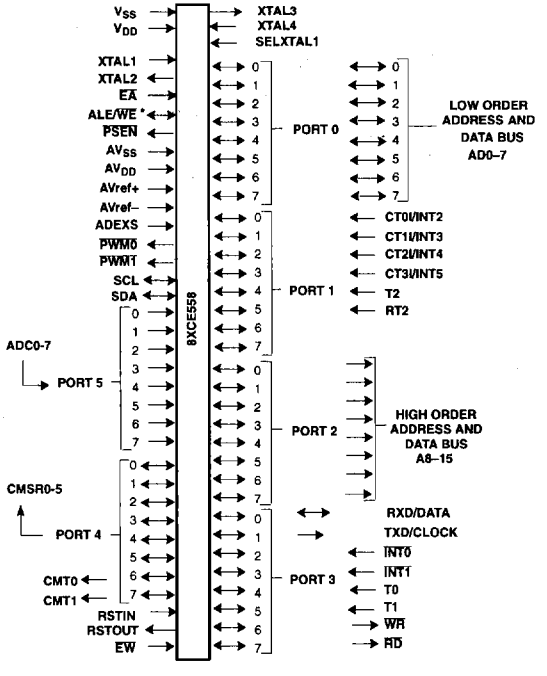
ORDERING INFORMATION

ROMless	ROM	EPROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHz	DRAWING NUMBER
80CE558EBB	83CE558EBB	89CE558EBB	0 to +70, 80-Pin Plastic Quad Flat Pack	3.5 to 16	SOT318
80CE558EFB	83CE558EFB	89CE558EFB	–40 to +85, 80-Pin Plastic Quad Flat Pack	3.5 to 16	SOT318

BLOCK DIAGRAM



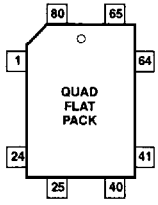
FUNCTIONAL DIAGRAM



*) only 89CE558 with alternate function

PINNING DIAGRAM

Pin	Function	Pin	Function
1	AVref–	42	P3.1/TXD/CLOCK
2	AVref+	43	P3.2/$\overline{INT0}$
3	AV_{SS1}	44	P3.3/$\overline{INT1}$
4	AV_{DD1}	45	P3.4/T0
5	P5.7/ADC7	46	P3.5/T1
6	P5.6/ADC6	47	P3.6/$\overline{WR}$
7	P5.5/ADC5	48	P3.7/$\overline{RD}$
8	P5.4/ADC4	49	NC
9	P5.3/ADC3	50	NC
10	P5.2/ADC2	51	XTAL2
11	P5.1/ADC1	52	XTAL1
12	P5.0/ADC0	53	V_{DD3}
13	V_{SS1}	54	V_{SS3}
14	V_{DD1}	55	P2.0/A8
15	ADEXS	56	P2.1/A9
16	$\overline{PWM0}$	57	P2.2/A10
17	$\overline{PWM1}$	58	P2.3/A11
18	$\overline{EW}$	59	P2.4/A12
19	P4.0/CMSR0	60	P2.5/A13
20	P4.1/CMSR1	61	P2.6/A14
21	P4.2/CMSR2	62	P2.7/A15
22	P4.3/CMSR3	63	$\overline{PSEN}$
23	RSTOUT	64	ALE/$\overline{WE}$ *
24	P4.4/CMSR4	65	$\overline{EA}$
25	P4.5/CMSR5	66	V_{DD4}
26	P4.6/CMT0	67	V_{SS4}
27	P4.7/CMT1	68	P0.7/AD7
28	V_{DD2}	69	P0.6/AD6
29	V_{SS2}	70	P0.5/AD5
30	RSTIN	71	P0.4/AD4
31	P1.0/CT0I/INT2	72	P0.3/AD3
32	P1.1/CT1I/INT3	73	P0.2/AD2
33	P1.2/CT2I/INT4	74	P0.1/AD1
34	P1.3/CT3I/INT5	75	P0.0/AD0
35	P1.4/T2	76	AV_{DD2}
36	P1.5/RT2	77	AV_{SS2}
37	P1.6	78	XTAL3
38	P1.7	79	XTAL4
39	SCL	80	SELXTAL1
40	SDA		
41	P3.0/RXD/DATA		



* only 89CE558 with alternative function

PIN DESCRIPTION

SYMBOL	PIN	DESCRIPTION
AV_{ref-} AV_{ref+}	1 2	**Low end** of analog to digital conversion reference resistor **High end** of analog to digital conversion reference resistor.
AV_{SS1} AV_{DD1}	3 4	**Analog** ground for ADC **Analog** power supply (+5V) for ADC
AV_{SS2} AV_{DD2}	77 76	**Analog** ground; for PLL oscillator **Analog** power supply; (+ 5V) for PLL oscillator
P5.7– P5.0	5– 12	**Port 5** 8–bit input port Port pin Alternative function P5.0–P5.7 Eight input channels to ADC (ADC0–ADC7)
V_{DD1-4}		**Digital power supply:** +5V power supply pins during normal operation and power reduction modes. All pins must be connected.
V_{SS1-4}		**Digital ground:** circuit ground potential. All pins must be connected.
V_{DD1} V_{SS1}	14 13	**Digital power supply**
V_{DD2} V_{SS2}	28 29	**Digital power supply**
V_{DD3} V_{SS3}	53 54	**Digital power supply**
V_{DD4} V_{SS4}	66 67	**Digital power supply**
ADEXS	15	**Start ADC operation:** Input starting analog to digital conversion triggered by a programmable edge (ADC operation can also be started by software). This pin must not float
$\overline{PWM0}$	16	**Pulse width modulation** output 0
$\overline{PWM1}$	17	**Pulse width modulation** output 1
$\overline{EW}$	18	**Enable watchdog timer:** Enable for T3 watchdog timer and disable Power–down mode. This pin must not float.
P4.0 – P4.7	19 – 22 24 – 27	**Port 4** 8–bit quasi–bidirectional I/O port Port pin Alternative function P4.0 CMSR0 } P4.1 CMSR1 } P4.2 CMSR2 } Timer T2: compare and set/reset P4.3 CMSR3 } outputs on a match with timer T2 P4.4 CMSR4 } P4.5 CMSR5 } P4.6 CMT0 } Timer T2: compare and toggle outputs P4.7 CMT1 } on a match with timer T2
RSTIN	30	**Reset:** Input to reset the 8xCE558.
RSTOUT	23	**Reset:** Output of the 8xCE558 for resetting peripheral devices during initialization and Watchdog Timer overflow.
P1.0 – P1.7	31 – 38	**Port 1** 8–bit quasi–bidirectional I/O port Port pin Alternative function P1.0 CT0I/INT2} P1.1 CT1I/INT3} : Capture timer inputs for P1.2 CT2I/INT4} timer T2 or external interrupt inputs P1.3 CT3I/INT5} P1.4 T2 : T2 event input, rising edge triggered P1.5 RT2 : T2 timer reset input, rising edge triggered P1.6 P1.7
SCL	39	**I^2C–bus serial clock** I/O port

PIN DESCRIPTION (Continued)

SYMBOL	PIN	DESCRIPTION
SDA	40	**I²C–bus serial data** I/O port If SCL and SDA are not connected to I²C–buslines, their input levels should be defined otherwise (e.g. by a pull–up resistor to V_{DD}) in order to prevent these port input stages from floating, which could affect the total power consumption. Especially during Power–down Mode extra power consumption by floating inputs is an adverse effect.
P3.0 – P3.7	41 – 48	8–bit quasi–bidirectional I/O port Port pin Alternative function P3.0 RXD : Serial input port P3.1 TXD : Serial output port P3.2 $\overline{INT0}$: External interrupt P3.3 $\overline{INT1}$: External interrupt P3.4 T0 : Timer 0 external input P3.5 T1 : Timer 1 external input P3.6 $\overline{WR}$: External data memory write strobe P3.7 $\overline{RD}$: External data memory read strobe
XTAL2	51	**Crystal pin 2**: output of the inverting amplifier that forms the oscillator. Left open–circuit when an external oscillator clock is used.
XTAL1	52	**Crystal pin 1**: input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator clock signal when an external oscillator is used. Must be connected to logic HIGH if the PLL oscillator is selected (SELXTAL1 = LOW)
P2.0 – P2.7	55 – 62	**Port2:** 8–bit quasi–bidirectional I/O port with internal pull–ups. During access to external memories (RAM/ROM) that use 16–bit addresses (MOVX@DPTR) Port 2 emits the high order address byte. The alternative function of P2.7 for the 89CE558 is the output enable signal for verify/read modes (active low). Port 2 can sink/source one TTL (=4 LSTTL) input. It can drive CMOS inputs without external pull–ups.
$\overline{PSEN}$	63	**Program Store Enable output:** read strobe to the external program memory via Port 0 and 2. Is activated twice each machine cycle during fetches from external program memory. When executing out of external program memory two activations of $\overline{PSEN}$ are skipped during each access to external data memory. $\overline{PSEN}$ is not activated (remains HIGH) during no fetches from external program memory. $\overline{PSEN}$ can sink/source 8 LSTTL inputs. It can drive CMOS inputs without external pull–ups.
ALE/$\overline{WE}$	64	**Address Latch Enable output**: latches the low byte of the address during access of external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE/$\overline{WE}$ can sink/–source 8 LSTTL inputs. It can drive CMOS inputs without an external pull–up. The alternative function for the 89CE558 is the programming pulse input $\overline{WE}$. To prohibit the toggling of ALE pin (RFI noise reduction) the bit RFI in the PCON Register (PCON.5) must be set by software. This bit is cleared on RESET and can be set and cleared by software. When set, ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX. ALE will retain its normal high value during Idle mode and a low value during Power–down mode while in the "RFI" mode. Additionally during internal access ($\overline{EA}$ = 1) ALE will toggle normally when the address exceeds the internal program memory size. During external access ($\overline{EA}$ = 0) ALE will always toggle normally, whether the flag "RFI" is set or not.
$\overline{EA}$	65	**External Access Input:** When, during RESET, $\overline{EA}$ is held at a TTL HIGH level the CPU executes out of the internal program memory, provided the program counter is less than 32768. When $\overline{EA}$ is held at a TTL low level during RESET, the CPU executes out of external program memory via Port 0 and Port 2. $\overline{EA}$ is not allowed to float. $\overline{EA}$ is latched during RESET and don't care after RESET.
P0.7–P0.0	68 –75	**Port 0:** 8–bit open drain bidirectional I/O port. It is also the multiplexed low–order address and data bus during accesses to external memory (during theses accesses internal pull–ups are activated). Port 0 can sink/source 8 LSTTL inputs.
XTAL3	78	**Crystal pin**, output of the inverting amplifier that forms the 32 KHz oscillator
XTAL4	79	**Crystal pin**, input to the inverting amplifier that forms the 32 KHz oscillator, must be connected to logic LOW if the PLL oscillator is not selected (SELXTAL1 = HIGH).
SELXTAL1	80	**Must be connected to logic HIGH level to select the HF oscillator, using the XTAL1/XTAL2 crystal. If pulled low the PLL is selected for clocking of the controller, using the XTAL3/ XTAL4 crystal.**

To avoid a 'latch–up' effect at Power–on, the voltage on any pin at any time must not be higher or lower than V_{DD}+ 0.5V or V_{SS}– 0.5V respectively.

ELECTROMAGNETIC COMPATIBILITY (EMC) IMPROVEMENTS

Primary attention was paid on the reduction of electromagnetic emission of the microcontroller 8xCE558.

The following features effect in reducing the electromagnetic emission and additionally improve the electromagnetic susceptibility:

- Four supply voltage pins (V_{DD}) and four ground pins (V_{SS}) with a pair of V_{DD} and V_{SS} at two adjacent pins in the center at one side of the package and at two adjacent pins at the opposite side of the package and one more V_{SS} pin at each of the other two sides of the package
- Separated V_{DD} pins for the internal logic and the port buffers
- Internal decoupling capacitance improves the EMC radiation behavior and the EMC immunity
- External capacitors are to be located as close as possible between pins V_{DD1} and V_{SS1}, V_{DD2} and V_{SS2}, V_{DD3} and V_{SS3} as well as V_{DD4} and V_{SS4} ; ceramic chip capacitors are recommended (100nF).

Useful in applications that require no external memory or temporarily no external memory:

- The ALE output signal (pulses at a frequency of $f_{CLK}/6$) can be disabled under software control (bit 5 in the SFR PCON: "RFI"); if disabled, no ALE pulse will occur. ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE (external data memory is accessed). ALE will retain its normal HIGH value during Idle mode and a LOW value during Power-down mode while in the "RFI" reduction mode. Additionally during internal access ($\overline{EA}$ = 1) ALE will toggle normally when the address exceeds the internal program memory size. During external access ($\overline{EA}$ = 0) ALE will always toggle normally, whether the flag "RFI" is set or not.

18.0 FUNCTIONAL DESCRIPTION

General

The 8xCE558 is a stand-alone high-performance microcontroller designed for use in real time applications such as instrumentation, industrial control, medium to high-end consumer applications and specific automotive control applications.

In addition to the 80C51 standard functions, the device provides a number of dedicated hardware functions for these applications.

The 8xCE558 is a control-oriented CPU with on-chip program and data memory. It can be extended with external program memory up to 64 K bytes. It can also access up to 64 K bytes of external data memory. For systems requiring extra capability, the 8xCE558 can be expanded using standard memories and peripherals.

The 8xCE558 has two software selectable modes of reduced activity for further power reduction- – Idle and Power-down. The Idle mode freezes the CPU while allowing the RAM, timers, serial ports and interrupt system to continue functioning. The Power–down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative. The Power–down mode can be terminated by an external Reset and by any one of the two external interrupts. (see description Wake–up from Power–down mode).

1.1 Memory organization

The central processing unit (CPU) manipulates operands in three memory spaces; these are the 64 K-byte external data memory, 1024 byte internal data memory (consisting of 256 bytes standard RAM and 768 bytes AUX-RAM) and the 64 K-byte internal and external program memory (see Figure 1).

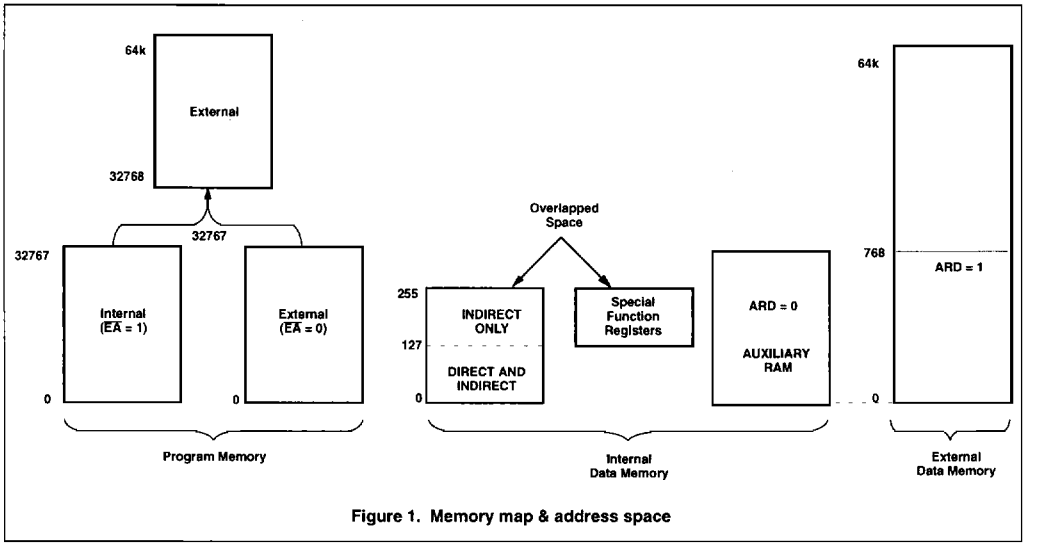


Figure 1. Memory map & address space

1.1.1 Program Memory

The program memory of the 8xCE558 consists of 32 Kbyte ROM resp. FEEPROM ("Flash Memory") on-chip, externally expandable up to 64 Kbyte. If the $\overline{EA}$ pin is held HIGH, the 8xCE558 executes out of the internal program memory unless the address exceeds 7FFFH. Locations 8000H through 0FFFFH are then fetched from the external program memory. If the $\overline{EA}$ pin was set LOW during RESET the 8xCE558 fetches all instructions from the external program memory. The $\overline{EA}$ input is latched during RESET and is don't care after RESET.

By setting a mask programmable security bit (ROM) resp. by software programmable security byte (FEEPROM) the internal program memory content is protected i.e. it cannot be read out at any time by any test mode or by any instruction in the external program memory space. The MOVC instructions are the only which have access to program code in the internal or external program memory. The $\overline{EA}$ input is latched during RESET and is 'don't care' after RESET. This implementation prevents from reading internal program code by switching from external program memory to internal program memory during MOVC instruction or an instruction that handles immediate data.

Table 1 lists the access to the internal and external program memory with MOVC instructions when the security feature has been activated.

1.1.2 Internal Data Memory

The internal data memory is divided into three physically separated parts:

256 byte of RAM, 768 byte of AUX-RAM, and a 128-byte special function area. These can be addressed each in a different way (see also 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 registerbank.
- RAM 128 to 255 can only be addressed indirectly. Address pointers are R0 and R1 of the selected registerbank.
- AUX-RAM 0 to 767 is indirectly addressable as external DATA MEMORY locations 0 to 767 via MOVX-Datapointer instruction, unless it is disabled by setting ARD = 1.
- AUX-RAM 0 to 767 is indirectly addressable via pageregister (XRAMP) and MOVX-Ri instructions, unless it is disabled by setting ARD = 1 (see Figure 2). When executing from internal program memory, an access to AUX-RAM 0 to 767 will not affect the ports P0, P2, P3.6 and P3.7. An access to external DATA MEMORY locations higher than 767 will be performed with the MOVX @ DPTR instructions in the same way as in the 80C51 structure, so with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read timing signals. Note that the external DATA MEMORY cannot be accessed with R0 and R1 as address pointer if the AUX-RAM is enabled (ARD = 0, default).
- The Special Function Registers (SFR) can only be addressed directly in the address range from 128 to 255 (see table 4).
- Four 8-register banks occupy locations 0 through 31 in the lower RAM area. 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 (see Figure 3). All registers except the program counter and the four 8-register banks reside in the Special Function Register address space.

Table 1.

	Access to Internal Program Memory	Access to External Program Memory
MOVC in internal program memory	YES	YES
MOVC in external program memory	NO	YES

If the security feature has not been activated, there are no restrictions for MOVC instructions.

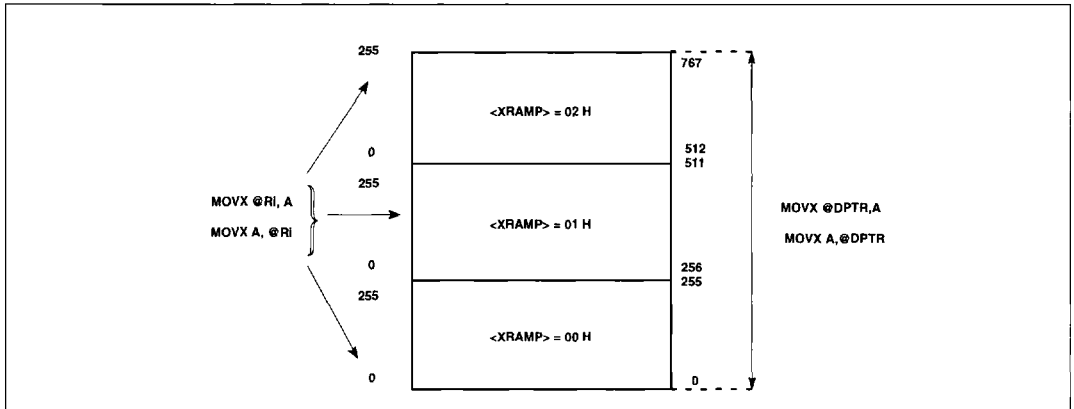


Figure 2. Indirect addressing of AUX-RAM (768 Bytes), ARD bit in PCON = LOW

Table 2.

LOCATION		ADDRESSED
RAM	0 to127	Direct and Indirect
AUX-RAM	0 to767	Indirect Only with MOVX
RAM	128 to255	Indirect Only
SFR	128 to 255	Direct Only

AUX-RAM Page Register XRAMP

The AUX-RAM Page Register is used to select one of three 256-byte pages of the internal 768-byte AUX-RAM for MOVX-accesses via R0 or R1. Its reset value is (XXXXXX00).

	7	6	5	4	3	2	1	0
XRAMP (0FAH)	x	x	x	x	x	x	XRAMP1	XRAMP0

x: undefined during read, a write operation must write "0" to these location

BIT	SYMBOL	FUNCTION
XRAMP.7	-	(reserved for future use [1])
XRAMP.6	-	(reserved for future use [1])
XRAMP.5	-	(reserved for future use [1])
XRAMP.4	-	(reserved for future use [1])
XRAMP.3	-	(reserved for future use [1])
XRAMP.2	-	(reserved for future use [1])
XRAMP.1	XRAMP1	AUX-RAM page select bit 1 AUX-RAM page select bit 0
XRAMP.0	XRAMP0	

NOTES:

1. User software should not write 1s to reserved bits. These bits may be used in future 80C51 family products to invoke new features. In that case, the reset or inactive value of the new bit will be LOW, and its active value will be HIGH. The value read from a reserved bit is indeterminate.

Table 3 shows the memory locations for all possible MOVX-accesses:

Table 3.

ARD[1]	XRAMP1	XRAMP0	MOVX @Ri,A and MOVX A,@Ri instructions access:
0	0	0	AUX-RAM locations 0 .. 255 (reset cond.)
0	0	1	AUX-RAM locations 256 .. 511
0	1	0	AUX-RAM locations 512 .. 767
0	1	1	*** no valid memory access ***
1	X	X	External RAM locations 0 .. 255

ARD[1]	XRAMP1	XRAMP0	MOVX @DPTR,A and MOVX A,@DPTR instructions access:
0	X	X	AUX-RAM locations 0 .. 767 (reset cond.) External RAM locations 768 .. 65535
1	X	X	External RAM locations 0 .. 65535

NOTES:

1. ARD (AUX-RAM Disable) is a bit in Special Function Register PCON

Table 4. Special Function Register Memory Map and Reset Values

	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							
2	DPL 00000000							
3	DPH 00000000							
4								
5								
6	ADRSL0 # XXXXXXXX	ADRSL1 # XXXXXXXX	ADRSL2 # XXXXXXXX	ADRSL3 # XXXXXXXX	ADRSL4 # XXXXXXXX	ADRSL5 # XXXXXXXX	ADRSL6 # XXXXXXXX	ADRSL7 # XXXXXXXX
7	PCON 00000000				P5 # XXXXXXXX	ADCON 00000000	ADPSS 00000000	ADRSH # 000000XX
8	TCON % 00000000	S0CON % 00000000	IEN0 % 00000000	IP0 % X0000000	TM2IR % 00000000	S1CON % 00000000	IEN1 % 00000000	IP1 % 00000000
9	TMOD 00000000	S0BUF XXXXXXXX	CML0 00000000		CMH0 00000000	S1STA # 11111000		PLLCON 00001101
A	TL0 00000000		CML1 00000000		CMH1 00000000	S1DAT 00000000	TM2CON 00000000	XRAMP XXXXXX00
B	TL1 00000000		CML2 00000000		CMH2 00000000	S1ADR 00000000	CTCON 00000000	FMCON * 000X0000
C	TH0 00000000		CTL0 # XXXXXXXX		CTH0 # XXXXXXXX		TML2 # 00000000	PWM0 00000000
D	TH1 00000000		CTL1 # XXXXXXXX		CTH1 # XXXXXXXX		TMH2 # 00000000	PWM1 00000000
E			CTL2 # XXXXXXXX		CTH2 # XXXXXXXX		STE 11000000	PWMP 00000000
F			CTL3 # XXXXXXXX		CTH3 # XXXXXXXX		RTE 00000000	T3 00000000

% = Bit addressable register
# = Read only register
X = Undefined
* = only in 89CE558

1.2 Addressing

The 8xCE558 has five modes for addressing:

- 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-register banks through Register, Direct or Register-Indirect addressing
- 1024 bytes of internal RAM through Direct or Register-Indirect addressing. Bytes 0-127 of internal RAM may be addressed directly/indirectly. Bytes 128-255 of internal RAM share their address location with the SFRs and so may only be addressed indirectly as data RAM. Bytes 0-767 of AUX-RAM can only be addressed indirectly via MOVX.
- Special Function Register through direct addressing at address locations 128-255 (see Figure 4).
- External data memory through Register-Indirect addressing
- Program memory look-up tables through Base- Register plus Index-Register-Indirect addressing

Byte Address (Hex)	Bit Address (Hex)								Byte Address (Decimal)
FFH	(MSB)							(LSB)	255
2FH	7F	7E	7D	7C	7B	7A	79	78	47
2EH	77	76	75	74	73	72	71	70	46
2DH	6F	6E	6D	6C	6B	6A	69	68	45
2CH	67	66	65	64	63	62	61	60	44
2BH	5F	5E	5D	5C	5B	5A	59	58	43
2AH	57	56	55	54	53	52	51	50	42
29H	4F	4E	4D	4C	4B	4A	49	48	41
28H	47	46	45	44	43	42	41	40	40
27H	3F	3E	3D	3C	3B	3A	39	38	39
26H	37	36	35	34	33	32	31	30	38
25H	2F	2E	2D	2C	2B	2A	29	28	37
24H	27	26	25	24	23	22	21	20	36
23H	1F	1E	1D	1C	1B	1A	19	18	35
22H	17	16	15	14	13	12	11	10	34
21H	0F	0E	0D	0C	0B	0A	09	08	33
20H	07	06	05	04	03	02	01	00	32
1FH	Bank 3								31
18H									24
17H	Bank 2								23
10H									16
0FH	Bank 1								15
08H									8
07H	Bank 0								7
00H									0

Figure 3. RAM Bit Addresses

Direct Byte Address (Hex)	Bit Address (Hex)								Register Mnemonic
FFH	(MSB)							(LSB)	
	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	
F8H	FF	FE	FD	FC	FB	FA	F9	F8	IP1
F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
	ET2	ECM2	ECM	ECM0	ECT3	ECT2	ECT1	ECT0	
E8H	EF	EE	ED	EC	EB	EA	E9	E8	IEN1
E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	
D8H	DF	DE	DD	DC	DB	DA	D9	D8	S1CON
	CY	AC	F0	RS1	RS0	OV	F1	P	
D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
	T2OV	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	
C8H	CF	CE	CD	CC	CB	CA	C9	C8	TM2IR
C0H	C7	C6	C5	C4	C3	C2	C1	C0	P4
	-	PAD	PS1	PS0	PT1	PX1	PT0	PX0	
B8H	BF	BE	BD	BC	BB	BA	B9	B8	IP0
B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	
A8H	AF	AE	AD	AC	AB	AA	A9	A8	IEN0
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9F	9E	9D	9C	9B	9A	99	98	S0CON
90H	97	96	95	94	93	92	91	90	P1
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

Figure 4. Special Function Register Bit Addresses

1.3 I/O facilities

The 8xCE558 has six 8-bit ports. Ports 0 to 3 are the same as in the 80C51, with the exception of the additional functions of Port 1. The parallel I/O function of Port 4 is equal to that of Ports 1, 2 and 3. Port 5 has a parallel input port function, but has no function as an output port.

The SDA and SCL lines serve the serial port SI01 (I^2C). Because the I^2C-bus may be active while the device is disconnected from V_{DD}, these pins, are provided with open drain drivers.

Ports 0, 1, 2, 3, 4 and 5 perform the following alternative functions:

- Port 0: provides the multiplexed low-order address and data bus used for expanding the 8xCE558 with standard memories and peripherals.
- Port 1: Port 1 is used for a number of special functions:
  - 4 capture inputs (or external interrupt request inputs if capture information is not utilized)
  - external counter input
  - external counter reset input
- Port 2: provides the high-order address bus when the 8xCE558 is expanded with external Program Memory and/or external Data Memory.
- Port 3: pins can be configured individually to provide:
  - external interrupt request inputs
  - counter inputs
  - receiver input and transmitter output of serial port SIO 0 (UART)
  - control signals to read and write external Data Memory

- Port 4: can be configured to provide signals indicating a match between timer counter T2 and its compare registers.
- Port 5: may be used in conjunction with the ADC interface. Unused analog inputs can be used as digital inputs. As Port 5 lines may be used as inputs to the ADC, these digital inputs have an inherent hysteresis to prevent the input logic from drawing too much current from the power lines when driven by analog signals. Channel to channel crosstalk should be taken into consideration when both digital and analog signals are simultaneously input to Port 5 (see DC characteristics).

All ports are bidirectional with the exception of Port 5 which is an input port.

Pins of which the alternative function is not used may be used as normal bidirectional I/Os.

The generation or use of a Port 1, Port 3 or Port 4 pin as an alternative function is carried out automatically by the 8xCE558 provided the associated Special Function Register bit is set HIGH.

The pull-up arrangements of Ports 1 - 5 are shown in Figure 5.

1.4 Pulse Width Modulated Outputs

The 8XCE558 contains two pulse width modulated output channels (see Figure 6). These channels generate pulses of programmable length and interval. The repetition frequency is defined by an 8-bit prescaler PWMP, which supplies the clock for the counter. The prescaler and counter are common to both PWM channels. 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. Provided the contents of either of these registers is greater than the counter value, the corresponding $\overline{PWM0}$ or $\overline{PWM1}$ 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. The pulse-width-ratio is in the range of 0/255 to 255/255 and may be programmed in increments of 1/255.

Buffered PWM outputs may be used to drive DC motors. The rotation speed of the motor would be proportional to the contents of PWMn. The PWM outputs may also be configured as a dual DAC. In this application, the PWM outputs must be integrated using conventional operational amplifier circuitry. If the resulting output voltages have to be accurate, external buffers with their own analog supply should be used to buffer the PWM outputs before they are integrated. The repetition frequency fpwm, at the PWMn outputs is give by:

$$fpwm = \frac{f_{CLK}}{2 \times (1 + PWMP) \times 255}$$

This gives a repetition frequency range of 123Hz to 31.4kHz (f_{CLK} = 16MHz). 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) 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. Both $\overline{PWMn}$ output pins are driven by push-pull drivers. These pins are not used for any other purpose.

Prescaler frequency control register PWMP

PWMP (FEH)

7	6	5	4	3	2	1	0
MSB							LSB

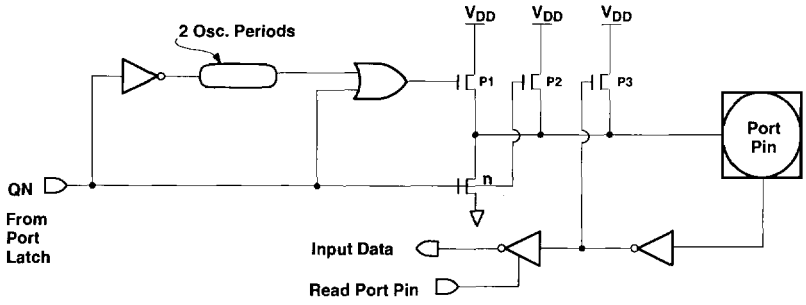
Bit	Function
PWMP.0-7	Prescaler division factor = (PWMP) + 1.

PWM0 (FCH)
PWM1 (FDH)

7	6	5	4	3	2	1	0
MSB							LSB

Reading PWMP gives the current reload value. The actual count of the prescaler cannot be read.

Bit	Function
PWM0.0-7 PWM1.0-7	Low/high ratio of $\frac{\overline{PWMn}}{255 - (PWMn)} = \frac{(PWMn)}{255 - (PWMn)}$



P1 is turned on for 2 oscillator periods after QN makes a 1-to-0 transition.
During this time, P1 also turns on P3 through the inverter to form an additional pull up.

Figure 5. I/O buffers in the 8xCE558 (Ports 1, 2, 3 and 4).

1.5 Analog / Digital Converter

In order to have a minimum of ADC service overhead in the microcontroller program, the ADC is able to operate autonomously by it's configurable autoscan feature and a set of 8 buffer registers (10 bit), which store the result of the conversion of each analog input channel. The functional diagram of the ADC is shown in Figure 7.

Feature Overview:

- 10-bit resolution.
- 8 multiplexed analog inputs.
- Programmable autoscan of the analog inputs.
- Bit oriented 8-bit scan-select register to select analog inputs.
- Continuous scan or one time scan configurable from 1 to 8 analog inputs.
- Start of a conversion by software or with an external signal.
- Programmable prescaler (div by 2, 4, 6, 8) to adapt to different oscillator frequencies.
- Eight 10-bit buffer registers, one register for each analog input channel.
- Conversion time for one A/D conversion: 15µs ... 50µs
- Differential non-linearity : DLe 1 LSB.
- Integral non-linearity : ILe 2 LSB.
- Offset error : OSe 2 LSB.
- Gain error : Ge 0.4 %.
- Absolute voltage error : Ae 3 LSB.
- Channel to channel matching : Mctc 1 LSB.
- Crosstalk between analog inputs : Ct < -60dB. @10 0 kHz.
- Monotonic and no missing codes.
- Separated analog (AVDD, AVSS) and digital (VDD, VSS) supply voltages.
- Reference voltage at two special pins : AVREF- and AVREF+.

1.5.1 Functional description:

Table 5.

ADCON	A/D control register	read/ write
ADPSS	Analog port scan-select register	read/ write
ADRSLn	8 A/D result registers, contain the 8 lower bits	read only
ADRSH	A/D result register, contains the 2 higher bits	read only
P5	Digital input port (shared with ana-log inputs)	read only

After a RESET of the microcontroller the ADCON and ADPSS register bits are initialized to zero, registers ADRSLn and ADRSH are undefined.

ADRSLn and ADRSH are specified as read only registers to prevent conflict situations between software write operation and writing the A/D conversion result in ADRSn by the hardware.

1.5.2 Idle and Power-down Mode for the A/D converter

If the Idle or Power-down Mode is activated, then an A/D conversion in progress is aborted, the ADSST flag is cleared and the internal clock is halted. The interrupt flag ADINT will not be set. The ADRSn registers 10 bit buffer are not affected.

Table 6 shows resultant conversion times (tconv) at external clock frequencies (f_{CLK}) and ADC prescaler divisors (m), which are programmable by the bits ADCON.7 and ADCON.6. For conversion times outside the specified range the specified characteristics are not guaranteed; those conversion times are put in brackets.

Table 6. Configuration time configurable examples (tconv/ms)

f_{CLK} m	6MHz	8MHz	12MHz	16MHz
2	26	19.5	[13]	[9.75]
4	50	37.5	25	18.75
6	[74]	[55.5]	37	27.75
8	[98]	[73.5]	49	36.75

conversion time = 6 x m + 1 machine cycles

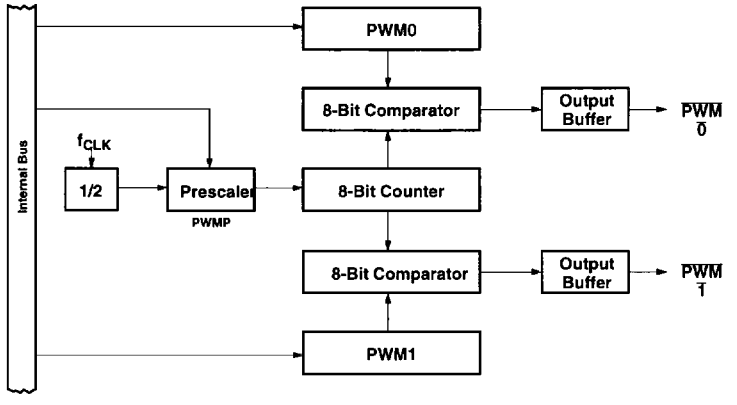


Figure 6. Functional Diagram of Pulse Width Modulated Outputs

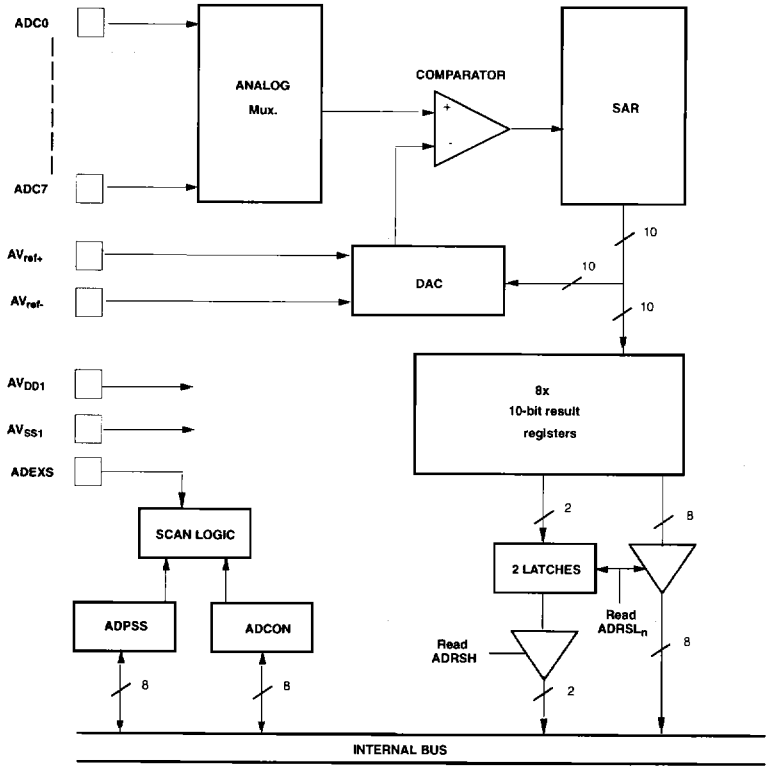


Figure 7. Functional Diagram of AD converter

1.5.3 A/D Control Register ADCON

	7	6	5	4	3	2	1	0
ADCON (D7H)	ADPR1	ADPR0	ADPOS	ADINT	ADSST	ADCSA	ADSRE	ADSFE

Bit	Symbol	Function
ADCON.7	ADPR1	Control bit for the prescaler.
ADCON.6	ADPR0	Control bit for the prescaler. ADPR1=0 APPR0=0 Prescaler divides by 2. ADPR1=0 ADPR0=1 Prescaler divides by 4. ADPR1=1 ADPR0=0 Prescaler divides by 6. ADPR1=1 ADPR0=1 Prescaler divides by 8.
ADCON.5	ADPOS	ADPOS is reserved for future use. Must be "0" if ADCON is written.
ADCON.4	ADINT	ADC interrupt flag. This flag is set when all selected analog inputs are converted, as well in continuous scan as in one time scan mode. An interrupt is invoked if this interrupt is enabled. ADINT must be cleared by software. It cannot be set by software.
ADCON.3	ADSST	ADC start and status. Setting this bit by software or by hardware (via ADEXS input) starts the A/D conversion of the selected analog inputs. ADSST stays a 'one' in continuous scan mode. In one time scan mode, ADSST is cleared when the last selected analog input channel has been converted. As long as ADSST is a one, new start commands to the ADC- block are ignored. An A/D conversion in progress is aborted if ADSST is cleared by software.
ADCON.2	ADCSA	1 = Continuous scan of the selected analog input after a start of A/D conversion. 0 = One time scan of the selected analog inputs after a start of A/D conversion.
ADCON.1	ADSRE	1 = A rising edge at inputs ADEXS will start the A/D conversion and generate a capture signal. 0 = A rising edge at input ADEXS has no effect.
ADCON.0	ADSFE	1 = A falling edge at input ADEXS will start the A/D conversion and generate a capture signal. 0 = A falling edge at input ADEXS has no effect.

1.5.4 A/D Input Port Scan-Select Register ADPSS

	7	6	5	4	3	2	1	0
ADPSS (E7H)	ADPSS7	ADPSS6	ADPSS5	ADPSS4	ADSS3	ADPSS2	ADPSS1	ADPSS0

ADPSS7-0 For each individual bit position:
- 0 = The corresponding analog input is skipped in the auto-scan loop.
- 1 = The corresponding analog input is included in the auto-scan loop.

If all bits are 'zero' then no A/D conversion can be started. If ADPSS is written while an A/D conversion is in progress (ADSST in the ADCON register is 'one') then the auto-scan loop with the previous selected analog inputs is completed first. The next auto-scan loop is performed with the new selected analog inputs.

1.5.5 A/D Result Registers ADRSLn and ADRSH:

There are 8 ADRSLn registers and one ADRSH register. Reading an ADRSLn register by software copies at the same time the two highest bits of the corresponding 10-bit conversion value in two latches. These two latches form bit position 0 and bit position 1 of register ADRSH. The upper 6 bits of ADRSH are all 0's if read. ADRSLn and ADRSH are read only registers.

1.6 Timer / Counters

The 8xCE558 contains three 16-bit timer/event counters: Timer 0, Timer 1 and Timer T2 and one 8-bit timer, T3. 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

1.6.1 Timer 0 and Timer 1

Timers 0 and 1 each have a control bit in SFR TMOD 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 1-to-0 transition at the corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show 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 1- to-0 transition. There are no restrictions on the duty cycle of the external input signal, but to insure 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:

- Mode 0: 8-bit timer or 8-bit counter each with divide by-32 prescaler
- Mode 1: 16-bit time-interval or event counter
- Mode 2: 8-bit time-interval or event counter with automatic reload upon overflow
- Mode 3: -Timer 0: one 8-bit time-Interval or event counter and one 8-bit time-Interval counter
  -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 or generate an interrupt. However the overflow from Timer 1 can be used to pulse the serial port baud-rate generator.

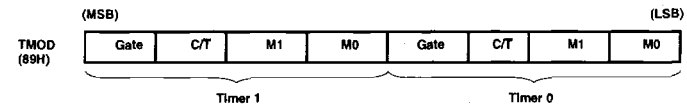
With a 16 MHz crystal, the counting frequency of these timer/counters is as follows:

- In the timer function, the timer is incremented at a frequency of 1,33 MHz - a division by 12 of the oscillator frequency
- 0 Hz to an upper limit of 0.66 MHz (1/24 of the oscillator requency) when programmed for external inputs

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

When configured as a counter, the register is incremented on every falling edge on the corresponding input pin, T0 or T1. The incremented register value can be read earliest during the second machine cycle after that one, during which the incrementing pulse occurred.

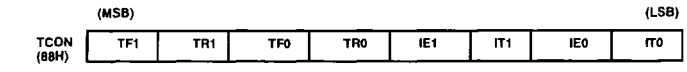
The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all HIGHs to all LOWs (or automatic reload value), with the exception of mode 3 as previously described.



Gate	Gating control when set. Timer/Counter "x" is enabled only while "INTx" pin is high and "TRx" control pin is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.
C/T	Timer or Counter Selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from "Tx" input pin).

M1	M0	Operating
0	0	8048 Timer "TLx" serves as 5-bit prescaler.
0	1	16-bit Timer/Counter "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload 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 only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

Figure 8. Timer/Counter Mode Control (TMOD) Register



Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 Run 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 by hardware when processor vectors to interrupt routine.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.

Symbol	Position	Name and Significance
IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Figure 9. Timer/Counter Control (TCON) Register

1.6.2 Timer T2

Timer T2 is a 16 bit timer/counter which has capture and compare facilities.

The operational diagram is shown in Figure 8.

The 16 bit timer/counter is clocked via a prescaler with a programmable division factor of 1, 2, 4 or 8. The input of the prescaler is clocked with 1/12 of the clock frequency, or by an external source connected to the T2 input, or it is switched off. The maximum repetition rate of the external clock source is $f_{CLK}/12$, twice that of Timer 0 and Timer 1. The prescaler is incremented on a rising edge. It is cleared if its division factor or its input source is changed, or if the timer/counter is reset (see also Figure 11: TM2CON). T2 is readable 'on the fly', without any extra read latches; this means that software precautions have to be taken against misinterpretation at overflow from least to most significant byte while T2 is being read. T2 is not loadable and is reset by the RST signal or at the positive edge of the input signal RT2, if enabled. In the Idle mode the timer/counter and prescaler are reset and halted.

T2 is connected to four 16-bit Capture Registers: CT0, CT1, CT2 and CT3. A rising or falling edge on the inputs CT0I, CT1I, CT2I or CT3I (alternative function of Port 1) results in loading the contents of T2 into the respective Capture Registers and an interrupt request.

Using the Capture Register CTCON (see Figure 12), these inputs may invoke capture and interrupt request on a positive, a negative edge or on both edges. If neither a positive nor a negative edge is selected for capture input, no capture or interrupt request can be generated by this input.

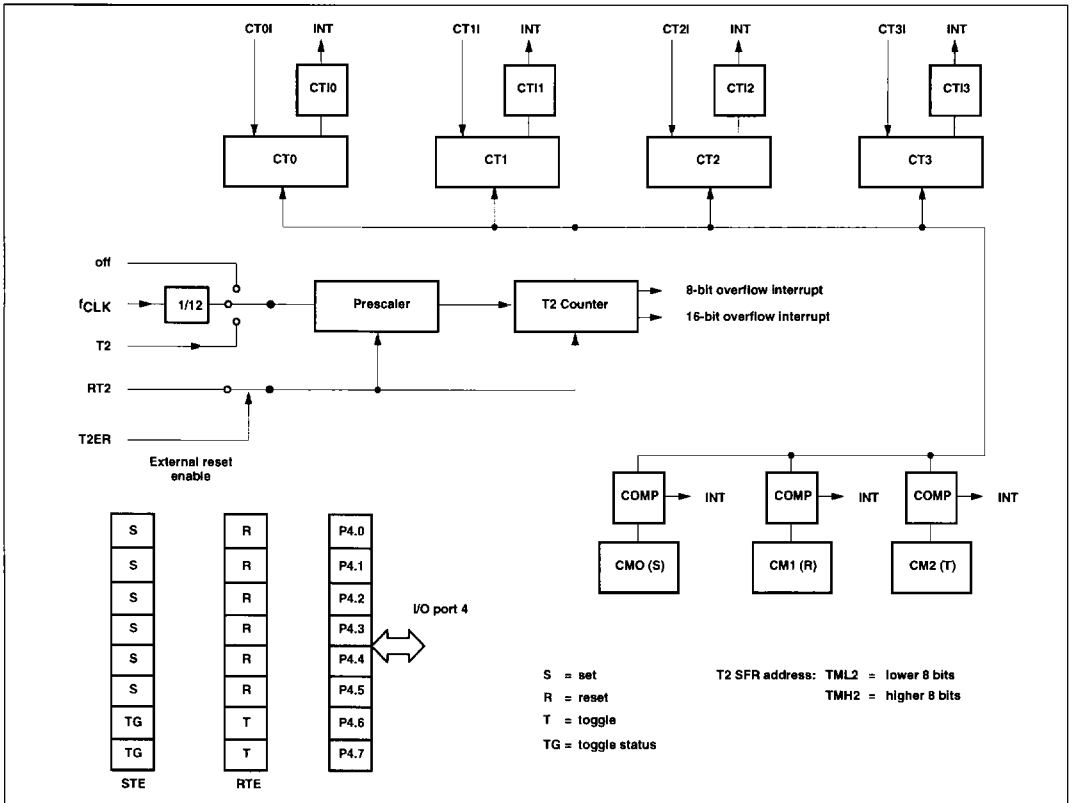
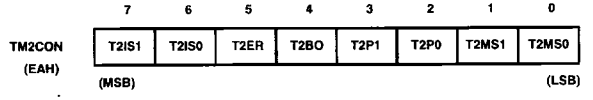


Figure 10. Block Diagram of Timer 2



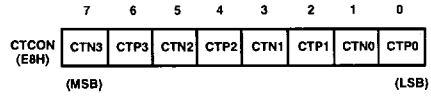
Bit	Symbol	Function
TM2CON.7	T2IS1	Timer T2 16-bit overflow interrupt select
TM2CON.6	T2IS0	Timer T2 byte overflow interrupt select
TM2CON.5	T2ER	Timer T2 external reset enable. When this bit is set, Timer T2 may be reset by a rising edge on RT2 (P1.5).
TM2CON.4	T2BO	Timer T2 byte overflow interrupt flag
TM2CON.3	T2P1	Timer T2 prescaler select
TM2CON.2	T2P0	

T2P1	T2P0	Timer T2 Clock
0	0	Clock source
0	1	Clock source/2
1	0	Clock source/4
1	1	Clock source/8

TM2CON.1	T2MS1	Timer T2 mode select
TM2CON.2	T2MS0	

T2MS1	T2MS0	Mode Selected
0	0	Timer T2 halted (off)
0	1	T2 clock source = $f_{CLK}/12$
1	0	Test mode; do not use
1	1	T2 clock source = pin T2

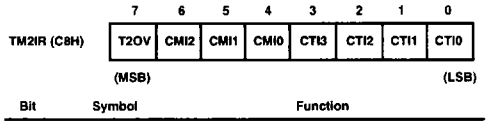
Figure 11. T2 Control Register (TM2CON)



Bit	Symbol	Capture/Interrupt on:
CTCON.7	CTN3	Capture Register 3 triggered by a falling edge on CT3I
CTCON.6	CTP3	Capture Register 3 triggered by a rising edge on CT3I
CTCON.5	CTN2	Capture Register 2 triggered by a falling edge on CT2I
CTCON.4	CTP2	Capture Register 2 triggered by a rising edge on CT2I
CTCON.3	CTN1	Capture Register 1 triggered by a falling edge on CT1I
CTCON.2	CTP1	Capture Register 1 triggered by a rising edge on CT1I
CTCON.1	CTN0	Capture Register 0 triggered by a falling edge on CT0I
CTCON.0	CTP0	Capture Register 0 triggered by a rising edge on CT0I

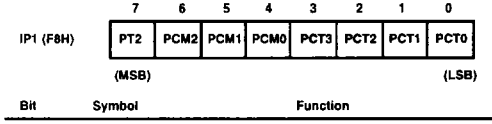
Figure 12. Capture Control Register (CTCON)

The contents of the Compare Registers CM0, CM1 and CM2 are continuously compared with the counter value of Timer T2. When a match occurs, an interrupt may be invoked. A match of CM0 sets the bits 0-5 of Port 4, a CM1 match resets these bits and a CM2 match toggles bits 6 and 7 of Port 4, provided these functions are enabled by the STE resp. RTE registers. A match of CM0 and CM1 at the same time results in resetting bits 0-5 of Port 4. CM0, CM1 and CM2 are reset by the RSTIN signal.



Bit	Symbol	Function
TM2IR.7	T2OV	Timer T2 16-bit overflow interrupt flag
TM2IR.6	CMI2	CM2 interrupt flag
TM2IR.5	CMI1	CM1 interrupt flag
TM2IR.4	CMI0	CM0 interrupt flag
TM2IR.3	CTI3	CT3 interrupt flag
TM2IR.2	CTI2	CT2 interrupt flag
TM2IR.1	CTI1	CT1 interrupt flag
TM2IR.0	CTI0	CT0 interrupt flag

Interrupt Flag Register (TM2IR)

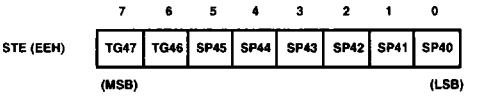


Bit	Symbol	Function
IP1.7	PT2	Timer T2 overflow interrupt(s) priority level
IP1.6	PCM2	Timer T2 comparator 2 interrupt priority level
IP1.5	PCM1	Timer T2 comparator 1 interrupt priority level
IP1.4	PCM0	Timer T2 comparator 0 interrupt priority level
IP1.3	PCT3	Timer T2 capture register 3 interrupt priority level
IP1.2	PCT2	Timer T2 capture register 2 interrupt priority level
IP1.1	PCT1	Timer T2 capture register 1 interrupt priority level
IP1.0	PCT0	Timer T2 capture register 0 interrupt priority level

Timer 2 Interrupt Priority Register (IP1)

Figure 13. Interrupt Flag Register (TM2IR) and Timer T2 Interrupt Priority Register (IP1)

For more information concerning the TM2CON, CTCON, TM2IR and the STE/RTE registers see IC20 handbook 1994.



Bit	Symbol	Function
STE.7	TG47	If "1" then P4.7 is reset on the next toggle, if LOW P4.7 is set on the next toggle
STE.6	TG46	If "1" then P4.6 is reset on the next toggle, if LOW P4.6 is set on the next toggle
STE.5	SP45	If "1" then P4.5 is set on a match between CM0 and Timer T2
STE.4	SP44	If "1" then P4.4 is set on a match between CM0 and Timer T2
STE.3	SP43	If "1" then P4.3 is set on a match between CM0 and Timer T2
STE.2	SP42	If "1" then P4.2 is set on a match between CM0 and Timer T2
STE.1	SP41	If "1" then P4.1 is set on a match between CM0 and Timer T2
STE.0	SP40	If "1" then P4.0 is set on a match between CM0 and Timer T2

Figure 14. Set Enable Register (STE)

Port 4 can be read and written by software without affecting the toggle, set and reset signals. At a byte overflow of the least significant byte, or at a 16-bit overflow of the timer/counter, an interrupt sharing the same interrupt vector is requested. Either one or both of these overflows can be programmed to request an interrupt.

All interrupt flags must be reset by software.

1.7 Watchdog Timer (T3) (see Figure 16)

In addition to Timer T2 and the standard timers, a watchdog timer consisting of an 11-bit prescaler and an 8-bit timer is also incorporated.

The timer is incremented every 1,5 ms, derived from the oscillator frequency of 16 MHz by the following:

$$f_{timer} = \frac{f_{CLK}}{12 \times 2048}$$

When a timer overflow occurs, the microcontroller - but not the PLL - is reset and a reset output pulse is generated at pin RSTOUT.

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 produce a reset upon overflow thus preventing the processor running out of control. The watchdog timer can only be reloaded if the condition flag WLE = PCON.4 has been previously set by software.

At the moment the counter is loaded the condition flag is automatically cleared.

The time interval between the timer's reloading and the occurrence of a reset depends on the reloaded value. For example, this may range from 1,5 ms to 0,375 s when using an oscillator frequency of 16 MHz.

In the Idle state the watchdog timer and reset circuitry remain active.

The watchdog timer is controlled by the watchdog enable pin ($\overline{EW}$). A LOW level enables the watchdog timer and disables the Power-down mode. A HIGH level disables the watchdog timer and enables the Power-down mode.

	7	6	5	4	3	2	1	0
RTE (EFH)	TP47	TP46	RP45	RP44	RP43	RP42	RO41	RP40
	(MSB)							(LSB)

Bit	Symbol	Function
RTE.7	TP47	If "1" then P4.7 toggles on a match between CM2 and Timer T2
RTE.6	TP46	If "1" then P4.6 toggles on a match between CM2 and Timer T2
RTE.5	RP45	If "1" then P4.5 is reset on a match between CM1 and Timer T2
RTE.4	RP44	If "1" then P4.4 is reset on a match between CM1 and Timer T2
RTE.3	RP43	If "1" then P4.3 is reset on a match between CM1 and Timer T2
RTE.2	RP42	If "1" then P4.2 is reset on a match between CM1 and Timer T2
RTE.1	RP41	If "1" then P4.1 is reset on a match between CM1 and Timer T2
RTE.0	RP40	If "1" then P4.0 is reset on a match between CM1 and Timer T2

Figure 15. Reset/Toggle Enable Register (RTE)

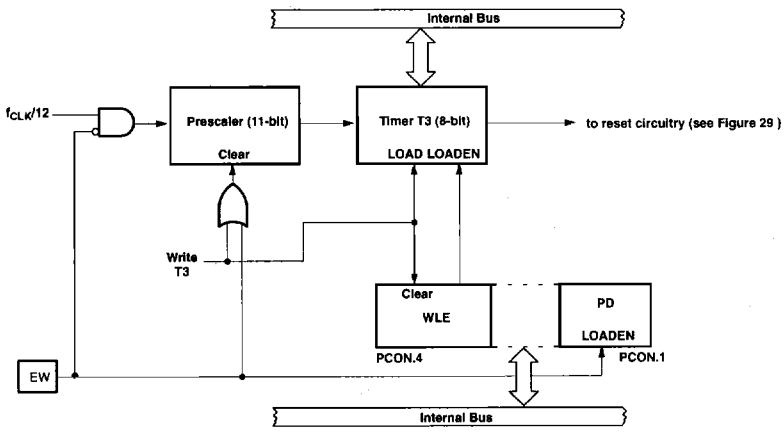


Figure 16. Watchdog Timer

1.8 Serial I/O

The 8xCE558 is equipped with two independent serial ports: SIO0 and SI01. SIO0 is the full duplex UART port, identical to the PCB80C51 serial port. SIO1 is an I^2C-bus serial I/O interface with byte oriented master and slave functions.

1.8.1 SIO0 (UART)

SIO 0 is a full duplex serial I/O port - it can transmit and receive simultaneously. This serial port is also receive-buffered. It can commence reception of a second byte before the previously received byte has been read from the receive register. If, however, the first byte has still not been read by the time reception of the second byte is complete, one of the bytes will be lost. The SIO0 receive and transmit registers are both accessed via the S0BUF special function register. Writing to S0BUF loads the transmit register, and reading S0BUF accesses to a physically separate receive register. SIO0 can operate in 4 modes:

Mode 0: Serial data is transmitted and received through RXD. TXD outputs the shift clock. 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 0: 10 bits are transmitted via TXD or received through RXD: a start bit (0), 8 data bits (LSB first), and a stop bit(1). On receive, the stop bit is put into RB8 (S0CON special function register). The baud rate is variable.

Mode 0: 11 bits are transmitted through TXD or received through RXD: a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in S0CON) can be assigned the value of 0 or 1. With nominal software, TB8 can be the parity bit (P in PSW). During a receive, the 9th data bit is stored in RB8 (S0CON), and the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 0: 11 bits are transmitted through TXD or received through RXD: a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). Mode 3 is the same as Mode 2 except the baud rate which is variable in Mode 3.

In all four modes, transmission is initiated by any instruction that writes to the S0BUF function register. Reception is initiated in Mode 0 when RI = 0 and REN = 1. In the other three modes, reception is initiated by the incoming start bit provided that REN = 1.

Modes 2 and 3 are provided for multiprocessor communications. In these modes, 9 data bits are received with the 9th bit written to RB8. The 9th bit is followed by the stop bit. The port can be programmed so that with receiving the stop bit, the serial port interrupt will be activated if, and only if RB8 = 1.

This feature is enabled by setting bit SM2 in S0CON. This feature may be used in multiprocessor systems.

For more information about how to use the UART in combination with the registers S0CON, PCON, IEN0, S0BUF and Timer register refer to the 80C51 Data Handbook IC20 1992.

	(MSB)							(LSB)
S0CON (98H)	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

Where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	$f_{CLK}/12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{CLK}/64$ or $f_{CLK}/32$
1	1	3	9-bit UART	variable

- SM2 enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1,, then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1, then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
- REN enables serial reception. Set by software to enable reception. Clear by software to disable reception.
- TB8 is 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, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stopbit that was received in mode 0, RB8 is not used.
- TI is the 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 is the 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.

Figure 17. Serial Port Control (S0CON) Register

1.8.2 SIO1 (I²C-bus Interface)

The I²C-bus is a simple bidirectional 2-wire bus for efficient inter-IC data exchange. Out-standing features of the I²C-bus are:

- Only two bus lines are required: a serial clock line (SCL) and a serial data line (SDA)
- Each device connected to the bus is software addressable by a unique address
- Masters can operate as Master-transmitter or as Master-receiver
- It's a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer
- Serial clock synchronization allows devices with different bit rates to communicate via the same serial bus
- ICs can be added to or removed from an I²C-bus system without affecting any other circuit on the bus
- Fault diagnostics and debugging are simple; malfunctions can be immediately traced

The on-chip I²C logic provides a serial interface that meets the I²C-bus specification, supporting all I²C-bus modes of operation, they are:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

The SI01 logic performs a byte oriented data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware. Via two pins the external I²C-bus is interfaced to the SIO1 logic:

SCL serial clock I/O and SDA (serial data I/O), (see Special Function Register bit S1CON.6/ENS1 for enabling the SIO1 logic).

The SIO1 logic handles byte transfer autonomously. It keeps track of the serial transfers, and a status register (S1STA) reflects the status of SIO1 and the I²C-bus.

Via the following four Special Function Registers the CPU interfaces to the I²C logic.

S1CON	control register. Bit addressable by the CPU
S1STA	status register whose contents may be used as a vector to service routines.
S1DAT	data shift register. The data byte is stable as long as S1CON.3/SI=1.
S1ADR	slave address register. It's LSB enables/ disables general call address recognition.

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 I²C bus. The STO bit is also cleared when ENS1 = "0".

S1CON (D8H)

7	6	5	4	3	2	1	0
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.

ENS1 = "1": When ENS1 is "1", SIO1 is enabled.

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

ENS1 should not be used to temporarily release SIO1 from the I²C-bus since, when ENS1 is reset, the I2C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

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 I2C 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.

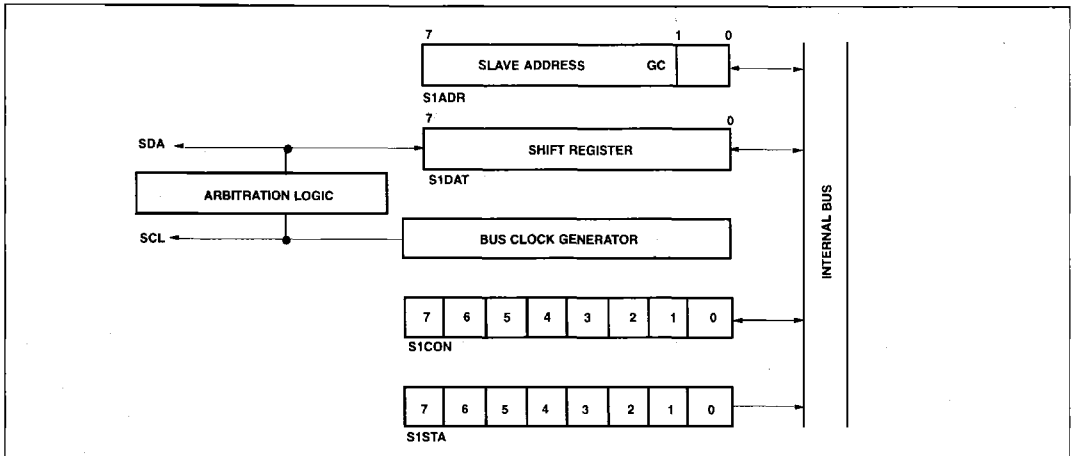


Figure 18. Block diagram of I²C serial I/O interface

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 the following events occur:

- A START condition is generated in MST mode.
- The own slave address has been received during AA = logic 1.
- The general call address has been received while S1ADR.0 and AA = logic 1
- A data byte has been received or transmitted as selected slave.
- A STOP or START condition is received as selected slave receiver or transmitter.

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 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 7.

Table 7. Serial Clock Rates

			BIT FREQUENCY (kHz) AT f_{CLK}	
CR2	CR1	CR0	12MHz	16MHz
1	0	0	50	66.7
1	0	1	3.75	5
1	1	0	75	100
1	1	1	100	-
0	0	0	200	266.7
0	0	1	7.5	10
0	1	0	300	400
0	1	1	400	-
0				

The frequencies shown in Table 7 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100kHz.

Except from the bit rate selection (see Table 7) and the timing of the SCL and SDA signals (see AC electrical characteristics in section 10 of this data sheet) the SIO circuit is the same as described in detail in the 8xC552 section of this handbook.

AA

Assert acknowledge bit, When this bit is set, an acknowledge is returned after any one of the following conditions:

- Own slave address is received
- General call address is received (S1ADR.0 = logic 1)
- A data byte is received, while the device is programmed to be a master receiver
- A data byte is received, while the device is a selected slave receiver

When the bit is reset, no acknowledge is returned. Consequently, no interrupt is requested when the own address or general is received.

SI

SIO1 interrupt flag. This is set, and an interrupt request is generated, after any of the following events occur:

- A START condition is generated in MST mode
- The own slave address has been received during AA = logic 1
- The general call address has been received while S1ADR.0 and AA = logic 1
- A data byte has been received or transmitted in MST mode (even if arbitration is lost)
- A data byte has been received or transmitted as selected slave
- A STOP or START condition is received as selected slave receiver or transmitter

STO

STOP flag. When in master mode, and this bit is set a STOP condition is generated. A STOP condition detected on the I^2C-bus clears this bit. This bit may also be set in slave mode in order to recover from an error condition. Then no STOP condition is generated to the I^2C-bus, but the hardware releases the SDA and SCL lines and switches to the not selected receiver mode. The STOP flag is cleared by the hardware.

STA
START flag. When this bit is set in slave mode, the hardware checks the I²C-bus and generates a START condition if the bus is free or after the bus becomes free. If the device operates in master mode it will generate a repeated START condition.

ENS1
0 = Serial I/O Disabled and reset. SDA and SCL outputs are open drain.

1 = Serial I/O Enabled.

Serial status register S1STA (S1STA is a read only register)

S1STA (D9H)

7	6	5	4	3	2	1	0
SC4	SC3	SC2	SC1	SC0	0	0	0

S1STA.3 - S1STA.7 hold a status code. S1STA.0 - S1STA.2 are held LOW. The contents of the status register may be used as a vector to a service routine. This optimize the response time of the software and consequently that of the I²C-bus.

Abbreviations used:

SLA : 7-bit slave address
R : Read bit
W : Write bit
ACK : Acknowledgement (acknowledge bit = logic 0)
$\overline{ACK}$: Not acknowledgement (acknowledge bit = logic 1)
DATA : 8b-t data byte to or from I²C-bus
MST : Master
SLV : Slave
TRX : Transmitter
REC : Receiver

The following is a list of the status codes:

MST/TRX mode
S1STA value

08H - A START condition has been transmitted
10H - A repeated START condition has been transmitted
18H - SLA and W have been transmitted, ACK has been received
20H - SLA and W have been transmitted, $\overline{ACK}$ received
28H - DATA and S1DAT has been transmitted, ACK received
30H - DATA and S1DAT has been transmitted, $\overline{ACK}$ received
38H - Arbitration lost in SLA, R/W or DATA

MST/REC mode
S1STA value

38H - Arbitration lost while returning $\overline{ACK}$
40H - SLA and R have been transmitted, ACK received
48H - SLA and R have been transmitted, $\overline{ACK}$ received
50H - DATA has been received, ACK returned
58H - DATA has been received, $\overline{ACK}$ returned

SLV/REC mode
S1STA value

60H - Own SLA and W have been received, ACK returned
68H - Arbitration lost in SLA, R/W as MST. Own SLA and W have been received, $\overline{ACK}$ returned
70H - General CALL has been received, ACK returned
78H - Arbitration lost in SLA, R/W as MST. General call has been received
80H - Previously addressed with own SLA. DATA byte received, ACK returned
88H - Previously addressed with own SLA. DATA byte received, $\overline{ACK}$ returned
90H - Previously addressed with general call. DATA byte has been received, ACK has been returned
98H - Previously addressed with general call. DATA byte has been received, $\overline{ACK}$ has been returned
A0H - A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

SLV/TRX mode
S1STA value

A8H - Own SLA and R have been received, ACK returned
B0H - Arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned
B8H - DATA byte has been transmitted, ACK returned
C0H - DATA byte has been transmitted, $\overline{ACK}$ returned
C8H - Last DATA byte has been transmitted (AA = logic 0), ACK received

Miscellaneous
S1STA value

00H - Bus error during MST mode or selected (AA = logic 0), ACK received

The data shift register S1DAT
S1DAT (DAH)

7	6	5	4	3	2	1	0

This register contains the serial data to be transmitted or data which has been received. Bit 7 is transmitted or received first; i.e. data is shifted from right to left.

The address register S1ADR
S1ADR (DBH)

7	6	5	4	3	2	1	0

S1ADR.0, GC : 0 = general call address is not recognized
1 = general call address recognized

S1ADR.7 - 1 : own slave address

This 8-bit register may be loaded with the 7-bit slave address to which the controller will respond when programmed as a slave receiver/transmitter. The LSB (GC) is used to determine whether the general call address is recognized.

1.9 Interrupt System

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronously to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response time in a single-interrupt system is in the range from 2.25us to 6.75 us when using a 16 MHz crystal. The latency time depends on the sequence of instructions executed directly after an interrupt request.

The 8xCE558 acknowledges interrupt requests from 15 sources as follows (see Figure 19):

- UART serial I/O port receive/transmit interrupt
- I²C-bus interface serial I/O interrupt
- ADC autoscan completion interrupt
- 'Seconds' timer interrupt SEC (ORed with INT1)
- $\overline{INT0}$ and $\overline{INT1}$ external interrupts
- Timer 0 and Timer 1 internal timer/counter interrupts
- Timer 2 internal timer/counter byte and/or 16-bit overflow, 3 compare and 4 capture interrupts (or 4 additional external interrupts). (1)

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated then the interrupt request flag remains set until the external interrupt pin $\overline{INTx}$ goes high. Consequently the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated. As these external interrupts are active LOW a "wire-ORing" of several interrupt sources to one input pin allows expansion.

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.

The eight Timer/Counter T2 Interrupt sources are: 4 capture Interrupts ([1]), 3 compare interrupts and an overflow interrupt. The appropriate interrupt request flags must be cleared by software.

The UART Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware. The service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared by software.

The I^2C Interrupt is generated by bit SI in register S1CON. This flag has to be cleared by software.

The ADC Interrupt is generated by bit ADINT, which is set when of all selected analog inputs to be scanned, the conversion is finished. ADINT must be cleared by software. It cannot be set by software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware (except the ADC interrupt request flag ADINT, which cannot be set by software). That is, interrupts can be generated or pending interrupts can be cancelled in software.

The Interrupts X0, T0, X1, T1, SEC, S0 and S1 are capable to terminate the Idle Mode.

Interrupt Enable Registers

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function registers IEN0 and IEN1. All interrupt sources can also be globally enabled or disabled by setting or clearing bit EA in IEN0. The interrupt enable registers are described in Figures 20 and 21.

Interrupt Priority Structure

Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function registers IP0 and IP1. IP0 and IP1 are described in Figures 22 and 23.

Interrupt priority levels are as follows:
"0"—low priority
"1"—high priority

A low priority interrupt may be interrupted by a high priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority occur simultaneously, the high priority level request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence. This second priority structure is shown in Table 8.

Interrupt Handling

The interrupt sources are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the previous machine cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware- generated LCALL is not blocked by any of the following conditions:

1. An interrupt of higher or equal priority level is already in progress.
2. The current machine cycle is not the final cycle in the execution of the instruction in progress. (No interrupt request will be serviced until the instruction in progress is completed.)
3. The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers. (No interrupt will be serviced after RETI or after a read or write to IP0, IP1, IE0, or IE1 until at least one other instruction has been subsequently executed.)

The polling cycle is repeated with every machine cycle, and the values polled are the values present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but is not being responded to because of one of the above conditions, and if the flag is inactive when the blocking condition is removed, then the blocked interrupt will not be serviced. Thus, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate service routine. In some cases it also clears the flag which generated the interrupt, and in others it does not. It clears the Timer 0, Timer 1, and external interrupt flags. An external interrupt flag (IE0 or IE1) is cleared only if it was transition-activated. All other interrupt flags

are not cleared by hardware and must be cleared by the software. The LCALL pushes the contents of the program counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 9.

Execution proceeds from the vector address until the RETI instruction is encountered. The RETI instruction clears the "priority level active" flip-flop that was set when this interrupt was acknowledged. It then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it was interrupted.

Note 1): If a capture register is unused and it's contents is of no interest, then the corresponding input pin CTnI/P1.n (n: 0...3) may be used as a (configurable) positive and/or negative edge triggered additional external interrupt input (INT2, INT3, INT4, INT5).

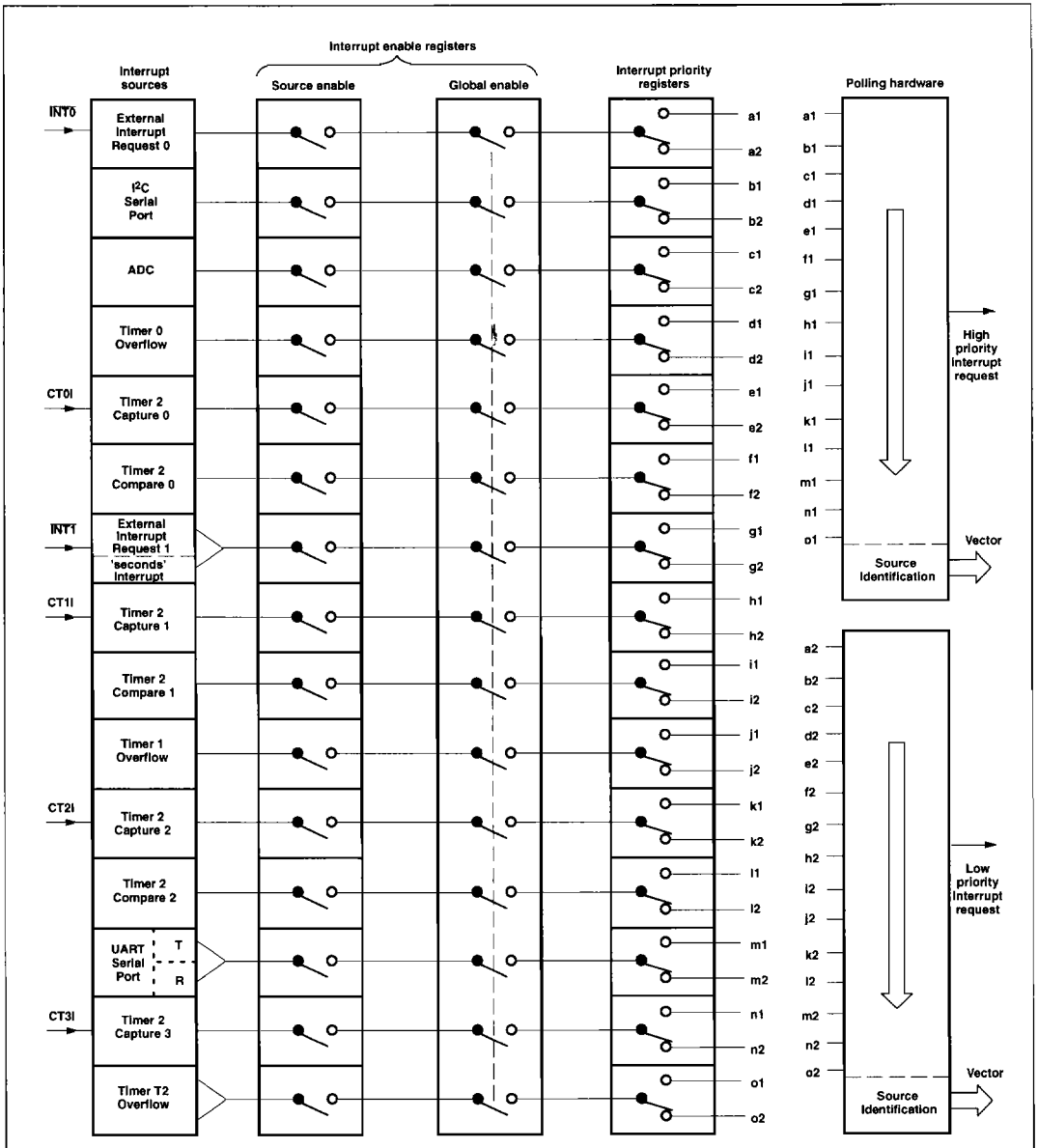


Figure 19. The Interrupt System

	7	6	5	4	3	2	1	0
IEN0 (A8H)	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0

(MSB) (LSB)

Bit	Symbol	Function
IEN0.7	EA	Global enable/disable control 0 = No interrupt is enabled 1 = Any individually enabled interrupt will be accepted
IEN0.6	EAD	Enable ADC interrupt
IEN0.5	ES1	Enable SIO1 (I^2C) interrupt
IEN0.4	ES0	Enable SIO0 (UART) interrupt
IEN0.3	ET1	Enable Timer 1/Seconds interrupt
IEN0.2	EX1	Enable External interrupt 1
IEN0.1	ET0	Enable Timer 0 interrupt
IEN0.0	EX0	Enable External interrupt 0

Figure 20. Interrupt Enable Register (IEN0)

	7	6	5	4	3	2	1	0
IEN1 (E8H)	ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0

(MSB) (LSB)

Bit	Symbol	Function
IEN1.7	ET2	Enable T2 overflow interrupt(s)
IEN1.6	ECM2	Enable T2 comparator 2 interrupt
IEN1.5	ECM1	Enable T2 comparator 1 interrupt
IEN1.4	ECM0	Enable T2 comparator 0 interrupt
IEN1.3	ECT3	Enable T2 capture register 3 interrupt
IEN1.2	ECT2	Enable T2 capture register 2 interrupt
IEN1.1	ECT1	Enable T2 capture register 1 interrupt
IEN1.0	ECT0	Enable T2 capture register 0 interrupt

If the enable bit is 0, then the interrupt is disabled,

If the enable bit is 1, then the interrupt is enabled.

Figure 21. Interrupt Enable Register (IEN1)

	7	6	5	4	3	2	1	0
IP0 (B8H)	-	PAD	PS1	PS0	PT1	PX1	PT0	PX0

(MSB) (LSB)

Bit	Symbol	Function
IP0.7	-	Reserved for future use
IP0.6	PAD	ADC interrupt priority level
IP0.5	PS1	SIO1 (I^2C) interrupt priority level
IP0.4	PS0	SIO0 (UART) interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1/Seconds interrupt priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

Figure 22. Interrupt Priority Register (IP0)

	7	6	5	4	3	2	1	0
IP1 (F8H)	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0

(MSB) (LSB)

Bit	Symbol	Function
IP1.7	PT2	T2 overflow interrupt(s) priority level
IP1.6	PCM2	T2 comparator 2 interrupt priority level
IP1.5	PCM1	T2 comparator 1 interrupt priority level
IP1.4	PCM0	T2 comparator 0 interrupt priority level
IP1.3	PCT3	T2 capture register 3 interrupt priority level
IP1.2	PCT2	T2 capture register 2 interrupt priority level
IP1.1	PCT1	T2 capture register 1 interrupt priority level
IP1.0	PCT0	T2 capture register 0 interrupt priority level

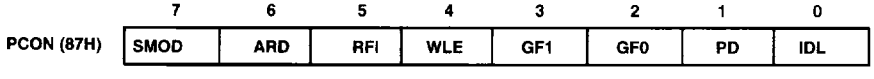
Figure 23. Interrupt Priority Register (IP1)

Table 8. Interrupt Priority Structure

SOURCE	NAME	PRIORITY WITHIN LEVEL
		(highest) ↑
External interrupt 0	X0	
SIO1 (I^2C)	S1	
ADC completion	ADC	
Timer 0 overflow	T0	
T2 capture 0	CT0	
T2 compare 0	CM0	
External interrupt 1/seconds interrupt	X1/SEC	
T2 capture 1	CT1	
T2 compare 1	CM1	
Timer 1 overflow	T1	
T2 capture 2	CT2	
T2 compare 2	CM2	
SIO0 (UART)	S0	
T2 capture 3	CT3	
Timer T2 overflow	T2	↓ (lowest)

Table 9. Interrupt Vector Addresses

SOURCE	NAME	VECTOR ADDRESS
External interrupt 0	X0	0003H
Timer 0 overflow	T0	000BH
External interrupt 1/seconds interrupt	X1/SEC	0013H
Timer 1 overflow	T1	001BH
SIO0 (UART)	S0	0023H
SIO1 (I^2C)	S1	002BH
T2 capture 0	CT0	0033H
T2 capture 1	CT1	003BH
T2 capture 2	CT2	0043H
T2 capture 3	CT3	004BH
ADC completion	ADC	0053H
T2 compare 0	CM0	005BH
T2 compare 1	CM1	0063H
T2 compare 2	CM2	006BH
T2 overflow	T2	0073H



Bit	Symbol	Function
PCON.7	SMOD	Double Baud rate bit. When set to logic 1 the baud rate is doubled when the serial port SIO0 is being used in modes 1, 2, or 3.
PCON.6	ARD	AUX-RAM disable bit. When set to a 1 the internal 768-byte AUX-RAM is disabled, so that all MOVX-Instructions access the external data memory - as it is with the standard PCB80C51.
PCON.5	RFI	Reduced radio frequency interference bit. When set to a 1 the toggling of ALE pin is prohibited. This bit is cleared on RESET (see also sections Features (EMC) and Pinning).
PCON.4	WLE	Watchdog load enable. This flag must be set by software prior to loading timer T3 (watchdog timer). It is cleared when timer T3 is loaded.
PCON.3	GF1	General-purpose flag bit
PCON.2	GF0	General-purpose flag bit
PCON.1	PD	Power-down bit. Setting this bit activates the power-down mode. It can only be set if input $\overline{EW}$ is high.
PCON.0	IDL	Idle mode bit. Setting this bit activates the idle mode.

If logic 1s are written to PD and IDL at the same time, PD takes precedence.

Figure 24. Power Control Register (PCON)

1.10 Power reduction modes

Two software-selectable modes of reduced power consumption are implemented. These are the Idle Mode and the Power-down Mode.

Idle Mode operation permits the interrupt, serial ports and timer blocks T0, T1 and T3 to function while the CPU is halted. The following functions are switched off when the microcontroller enters the Idle Mode:

- CPU (halted)
- Timer 2 (stopped and reset)
- PWM0, PWM1 (reset, output = HIGH)
- ADC (aborted if conversion in progress)

The following functions remain active during Idle Mode. These functions may generate an interrupt or reset and thus terminate the Idle Mode:

- Timer 0, Timer 1, Timer 3 (Watchdog timer)
- UART
- I^2C
- External interrupt
- Second Timer

In Power-down Mode the system clock is halted. If the PLL oscillator is selected (SELXTAL1 = 0) and the RUN32 bit is set, the 32 kHz oscillator keeps running, otherwise its stopped. If the HF-oscillator (SELXTAL1 = 1) is selected, it is freezed after setting the bit PD in the PCON register.

1.10.1 Power Control Register

The modes Idle and Power-down are activated by software via the Special Function Register PCON. Its hardware address is 87h. PCON is not bit addressable. The reset value of PCON is (00000000).

1.10.2 Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode.

The status of external pins during Idle mode is shown in Table 10.

There are three ways to terminate the Idle mode:

Activation of any enabled interrupt X0, T0, X1, SEC, T1, S0 or S1 will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with an external hardware reset. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 HF oscillator periods) to complete the reset operation if the HF oscillator is selected.

Table 10. External Pin Status During Idle and Power-Down Modes

MODE	MEMORY	ALE	$\overline{PSEN}$	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	SCL/SDA	$\overline{PWM0/PWM1}$
Idle	Internal	1	1	data	data	data	data	data	operative (1)	HIGH
Idle	External	1	1	high - Z	data	Address	data	data	operative (1)	HIGH
Power-down	Internal	0	0	data	data	data	data	data	high-Z	HIGH
Power-down	External	0	0	high - Z	data	data	data	data	high-Z	HIGH

NOTE:

1. In Idle Mode SCL and SDA can be active as outputs only if SIO1 is enabled; if SIO1 is disabled (S1CON.6/ENS1 = 0) these pins are in a high-impedance state.

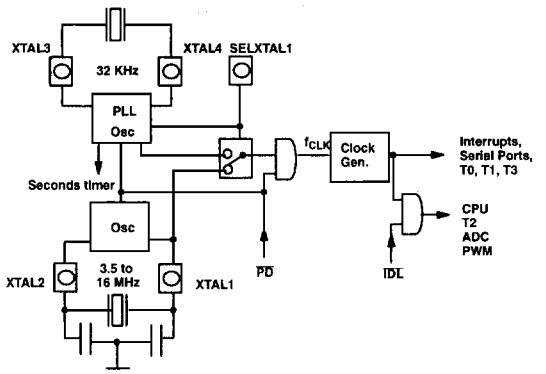


Figure 25. Idle and Power Down Hardware

When the PLL oscillator is selected a hardware reset of ≥1 µsec is required and the microcontroller restarts within 63 msec after the reset has finished.

The third way of terminating the Idle mode is by internal watchdog reset. The microcontroller restarts after 3 machine cycles in all cases.

1.10.3 Power-down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the HF oscillator is stopped. The 32 KHz oscillator may stay running. The content of the on-chip RAM and the Special Function Registers are not saved. Note that the Power-down mode can not be entered when the watchdog has been enabled.

The Power-down mode can be terminated by an external RESET in the same way as in the 80C51 (RAM is saved, SFRs are cleared due to RESET) or in addition by any one of the external interrupts, $\overline{INT0}$, $\overline{INT1}$ or Seconds interrupt. (see description Wake-up from Power-down mode)

The status of the external pins during Power-down mode is shown in Table 10. If the Power-down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a logic1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor P1 (see Figure 5)

1.10.4 Wake-up from Power-down Mode

The Power-down mode of the 8xCE558 can also be terminated by any one of the three enabled interrupts, $\overline{INT0}$, $\overline{INT1}$ or Seconds interrupt.

A termination with these interrupts do not affect the internal data memory and does not affect the Special Function Registers. This gives the possibility to exit Power-down without changing the port output levels. To terminate the Power-down mode with an external interrupt, $\overline{INT0}$ or $\overline{INT1}$ must be switched to be level-sensitive and must be enabled. The external interrupt input signal $\overline{INT0}$ or $\overline{INT1}$ must be kept LOW till the oscillator has restarted and stabilized (see Figure 26) in order to prevent any interrupt priority problems during wake-up. A Seconds interrupt will terminate the Power-down mode if enabled and $\overline{INT1}$ is level sensitive. Wake-up time is 60 ms in this case. The priority of the desired wake-up interrupt should be higher than the priorities of all other enabled interrupt sources. The instruction following the one that put the device into the Power-down mode will be the first one which will be executed after the wake-up.

1.11 Oscillator circuits

The input signal SELXTAL1 selects for XTAL1, 2 oscillator (standard 80C51) or the PLL oscillator. The not selected oscillator is halted and its XTAL pin must be not connected.

1.11.1 XTAL1, 2 Oscillator circuit (standard 80C51)

The oscillator circuit of the 8xCE558 is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between the XTAL1 and XTAL2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. Both are operated in parallel resonance. XTAL1 is the high gain amplifier input, and XTAL2 is the output (see Figure 27). To drive the 8xCE558 externally, XTAL1 is driven from an external source and XTAL2 left open-circuit (see Figure 28).

1.11.2 XTAL3, 4 circuitry, PLL oscillator

See appendix 1, "Specification PLL oscillator for 80C51 derivatives."

1.12 Reset Circuitry

The reset input pin RSTIN is connected to a Schmitt trigger for noise reduction (see Figure 29). Is the HF-oscillator selected a reset is accomplished by holding the RSTIN pin HIGH for at least 2 machine cycles (24 system clock periods). Is the PLL-oscillator selected the RSTIN-pulse must have a width of 1 µs at least, independent of the 32 kHz-oscillator is running or not (see PLL description). The CPU responds by executing an internal reset. The RSTOUT pin represents the signal resetting the CPU and can be used to reset peripheral devices.

The RSTOUT level also could be high due to a Watchdog timer overflow.

The length of the output pulse from T3 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

During reset, ALE and $\overline{PSEN}$ output a HIGH level. In order to perform a correct reset, this level must not be affected by external elements.

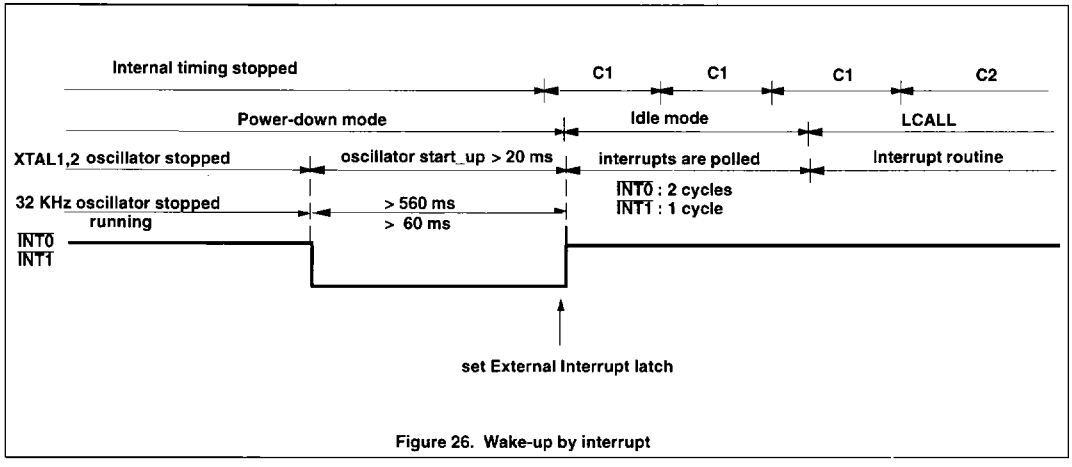


Figure 26. Wake-up by interrupt

A reset leaves the internal registers as follows:

REGISTER	CONTENT	
ACC	0000	0000
ADCON	xx00	0000
ADRSH	xxxx	xxxx
ADPSS	0000	0000
ADRSL0-7	xxxx	xxxx
B	0000	0000
CML0-CML2	0000	0000
CMH0-CMH2	0000	0000
CTCON	0000	0000
CTL0-CTL3	xxxx	xxxx
CTH0-CTH3	xxxx	xxxx
DPL	0000	0000
DPH	0000	0000
FMCON	000x	0000
IEN0	0000	0000
IEN1	0000	0000
IP0	x000	0000
IP1	0000	0000
PCON	0000	0000
PLLCON	0000	1101
PSW	0000	0000
PWM0	0000	0000
PWM1	0000	0000
PWMP	0000	0000
P0-P4	1111	1111
P5	xxxx	xxxx
RTE	0000	0000
S0BUF	xxxx	xxxx
S0CON	0000	0000
S1ADR	0000	0000
S1CON	0000	0000
S1DAT	0000	0000
S1STA	1111	1000
SP	0000	0111
STE	1100	0000
TCON	0000	0000
TH0, TH1	0000	0000
TMH2	0000	0000
TL0, TL1	0000	0000
TML2	0000	0000
TMOD	0000	0000
TM2CON	0000	0000
TM2IR	0000	0000
T3	0000	0000
XRAMP	xxxx	xx00

The internal RAM is not affected by reset. At power-on, the RAM content is indeterminate.

1.13 Power-on Reset

An automatic reset can be obtained by switching on V_{DD}, if the RSTIN pin is connected to V_{DD} via a capacitor, as shown in Figure 30. Is the HF oscillator selected the V_{DD} rise time must not exceed 10 ms and the capacitor should be at least 2.2 µF. The decrease of the RSTIN pin voltage depends on the capacitor and the internal resistor R_{RST}. That voltage must remain above the lower threshold for at minimum the HF-oscillator start-up time plus 2 machine cycles. Is the PLLC-oscillator selected a 0.1 nF capacitor is sufficient to obtain an automatic reset.

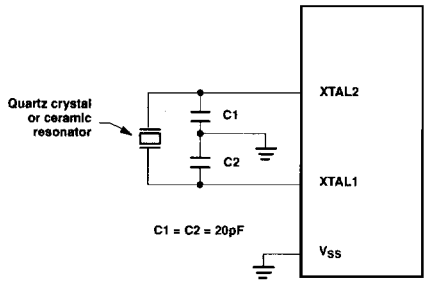


Figure 27. Using the On-Chip Oscillator

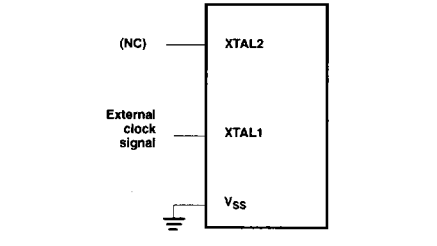


Figure 28. Using an external clock

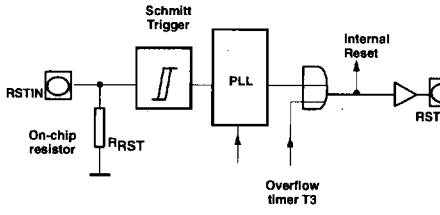


Figure 29. On-chip Reset Configuration

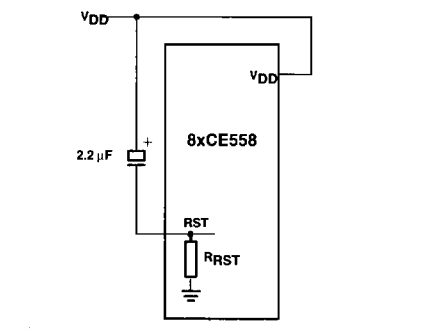


Figure 30. Power-on Reset

2.0 INSTRUCTION SET

The 8xCE558 uses the powerful instruction set of the PCB80C51. It consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. Using a 16 MHz quartz, 64 of the instructions are executed in 0.75 µs, 45 in 1,5 µs and the multiply, divide instructions in 3 µs.

A summary of the instruction set is given in Table 11.

The 8xCE558 has additional Special Function Registers to control the on-chip peripherals.

2.1 Addressing Modes

Most instructions have a "destination, source" field that specifies the data type, addressing modes and operands involved. For all these instructions, except for MOVs, the destination operand is also the source operand (e.g. ADD A,R7).

There are five kinds of addressing modes:

- Register Addressing
  - R0 - R7 (4 banks)
  - A,B,C (bit), AB (2 bytes), DPTR (double byte)
- Direct Addressing
  - lower 128 bytes of internal Main RAM (including the 4 R0-R7 register banks)
  - Special Function Registers 1
  - 28 bits in a subset of the internal Main RAM
  - 128 bits in a subset of the Special Function Registers
- Register-Indirect Addressing
  - internal Main RAM (@R0, @R1, @SP [PUSH/POP])
  - internal Auxiliary RAM (@R0, @R1, @DPTR)
  - external Data Memory (@R0, @R1, @DPTR)
- Immediate Addressing
  - Program Memory (in-code 8 bit or 16 bit constant)
- Base-Register-plus Index-Register-Indirect Addressing
  - Program Memory look-up table (@DPTR+A, @PC+A)

The first three addressing modes are usable for destination operands.

80C51 FAMILY INSTRUCTION SET

Table 11. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings[1]

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ANL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@Ri	8-bit internal data RAM location addressed indirectly through register R1 or R0 of the actual register bank..
#data	8-bit constant included in the instruction.
#data 16	16-bit constant included in the instruction
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

Table 11. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with carry	1	1
ADDC	A,direct	Add direct byte to Accumulator with carry	2	1
ADDC	A,@Ri	Add indirect RAM to Accumulator with carry	1	1
ADDC	A,#data	Add immediate data to A_{CC} with carry	2	1
SUBB	A,Rn	Subtract Register from A_{CC} with borrow	1	1
SUBB	A,direct	Subtract direct byte from A_{CC} with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A_{CC} with borrow	1	1
SUBB	A,#data	Subtract immediate data from A_{CC} with borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement Register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1

Table 11. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
LOGICAL OPERATIONS (Continued)				
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	1
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator left	1	1
RLC	A	Rotate Accumulator left through the carry	1	1
RR	A	Rotate Accumulator right	1	1
RRC	A	Rotate Accumulator right through the carry	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	RN,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A_{CC}	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A_{CC}	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr) to A_{CC}	1	2
MOVX	A@DPTR	Move external RAM (16-bit addr) to A_{CC}	1	2
MOVX	A,@Ri,A	Move A_{CC} to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A_{CC} to external RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	1
XCHD	A,@Ri	Exchange low-order digit indirect RAM with A_{CC}	1	1

Table 11. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear carry	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry	2	1
MOV	bit,C	Move carry to direct bit	2	2
JC	rel	Jump if carry is set	2	2
JNC	rel	Jump if carry not set	2	2
JB	rel	Jump if direct bit is set	2	2
JNB	rel	Jump if direct bit is not set	2	2
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2
PROGRAM BRANCHING				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is zero	2	2
JNZ	rel	Jump if Accumulator is not zero	2	2
CJNE	A,direct,rel	Compare direct byte to A_{CC} and jump if not equal	3	2
CJNE	A,#data,rel	Compare immediate to A_{CC} and jump if not equal	3	2
CJNE	RN,#data,rel	Compare immediate to register and jump if not equal	3	2
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	2
DJNZ	Rn,rel	Decrement register and jump if not zero	2	2
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	2
NOP		No operation	1	1

3.0 FLASH EEPROM

3.1 General

- 32 kbyte electrically erasable internal program memory with Block-and Page-Erase option ("Flash Memory").
- Internal fixed boot ROM.
- Up to 32 kbyte external program memory in combination with the internal FEEPROM ($\overline{EA}$=1).
- Up to 64 kbyte external program memory if the internal program memory is switched off ($\overline{EA}$=0).

The FEEPROM can be read and written byte-wise. Full Erase, Block Erase, and Page erase will erase 32 kbyte, 256 bytes and 32 bytes respectively. In-circuit programming and out-of-circuit programming is possible. On-chip erase and write timing generation and on chip high voltage generation contribute to a user friendly interface.

3.2 Features

- Read: byte-wise
- Write: byte-wise within 2.5 ms. (previously erased by a page, block or full erase).
- Erase: Page Erase (32 bytes) within 5 ms. Block Erase (256 bytes) within 5 ms. Full Erase (32 kbyte) within 5 ms. Erased bytes contain FFH.
- Endurance: 100 erase and write cycles each byte at T_{amb} = 22°C
- Retention: 10 years
- Out-of-circuit programming: Parallel programming with 87C51 compatible hardware Interface to programmer.
- In-circuit programming: Serial programming via RS232 interface under boot ROM program control. Auto baud rate selection. Intel Hex Object file Format.
- The user program can call routines in the boot ROM for erase, write and verify of the FEEPROM.
- High programming voltage generation: on chip
- Zero point on-chip oscillator and timer to generate the write and erase time durations.
- Programmable security for the code in the FEEPROM to prevent software piracy. The Security Byte is located in the highest address (7FFFH) of the FEEPROM.
- Supply voltage monitoring circuit on-chip to prevent loss of information in the FEEPROM during power-on and power-off.

3.3 Memory map

Figure 31 shows the memory map of the user program memory and the boot ROM. They are located in the same program address space. Two bits UBS1 and UBS0 of the FEEPROM control special function register FMCON select between the two memory blocks.

User program memory selection
If UBS1 and UBS0 are both 0, then the user program memory is mapped into the 64k program memory space and the boot ROM cannot be selected. This is the situation after a reset when $\overline{PSEN}$ and ALE have not been pulled down during reset. Program execution starts at 0000H in the internal FEEPROM or in the external program memory dependent on the level of $\overline{EA}$ during reset.

Boot ROM selection
If UBS1 and UBS0 are both '1' then the boot ROM is mapped into the 64k program memory space and the user program memory cannot be selected. This is the situation after a reset when during reset $\overline{PSEN}$ and $\overline{EA}$ are pulled down while ALE stay high. Program execution starts at 0000H of the boot ROM. The boot ROM size is 1 kbyte and its address map is repeated after each 1k addresses. Besides serial the in-circuit programming routines the Boot ROM contains the routines for erase write and verify of the FEEPROM which can be called by the user program (LJMP/LCALL to the address space between 63k and 64k).

Switching between user program memory and boot ROM
Switching between user program memory (internal or external) and boot ROM is possible if UBS1 and UBS0 are 0,1 or 1,0. Then in the program memory address space between 0 and 63k the user program memory is selected and in the memory space between 63k and 64k the boot ROM is selected.

To switch from user program memory to boot ROM first UBS0 must be set (UBS1 stay 0) and a jump or call instruction to a location >63k must be executed. At the moment of crossing the 63k address border by a call or jump instruction the switching between user and boot memory is performed without timing problems.

To switch from boot ROM to user program memory the boot program will be running between 63k and 64k. UBS1 is set and UBS0 is cleared and a jump or return instruction to a location <63k must be executed. At the moment of crossing the 63k address border by a jump or return instruction the switching from boot ROM to user memory (internal or external) is performed. After crossing the 63k address border UBS1 is immediately cleared by hardware and the total 64k memory space is mapped as user program memory. By clearing UBS1 by hardware, no special requirements to the user program are necessary to do that after a boot routine or erase or write routine.

A small restriction for memory switching is that no memory switching is allowed from or to the address space between 63k and 64k of the user program memory because the UBS bits must stay 0 in this range and no 63k address crossing would take place. This restriction can be avoided if the memory switching is always done by a subroutine in the address range between 0 and 63k.

Description
The user program code in the FEEPROM is executed as in the standard 80C51 microcontroller. Erase and write cycles in the FEEPROM are always performed under control of the boot program in the boot ROM in the address space between 63k and 64k. Address and data parameters are passed via DPTR and accumulator A respectively. During an erase or write cycle in the FEEPROM no other access or program execution in the FEEPROM is possible. All interrupts must be disabled when the user program calls a user routine in the boot ROM.

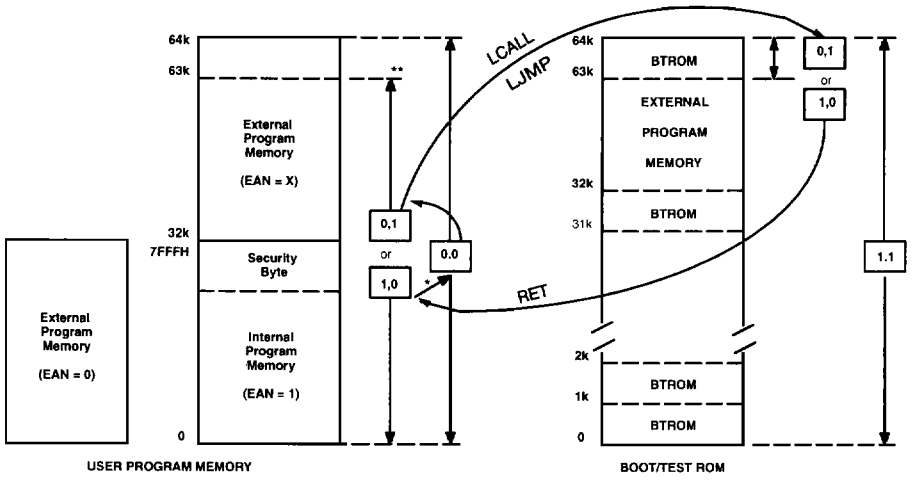
The boot routine for serial programming takes care of addressing, data transfer, blank-check, verify, high voltage control, error message, interrupt disabling/enabling and return to the user program memory. It also contains the serial communication routine.

The FEEPROM control register FMCON is a special function register. It contains the control bits for verify, write, erase and boot ROM switching.

FMCON (FBH)

7	6	5	4	3	2	1	0
UBS1	UBS0	HV	_1	FCB3	FCB2	FCB1	FCB0

Note 1): Reserved for future use; a write operation must write "0" to the location



X,X = UBS1, UBS0 User-Boot selection bits

* UBS1 is cleared by hardware if program is executed below the 63k border and UBS0=0.
** In the program execution between 63k and 64k in the user program no call or jump to the BTROM is allowed.

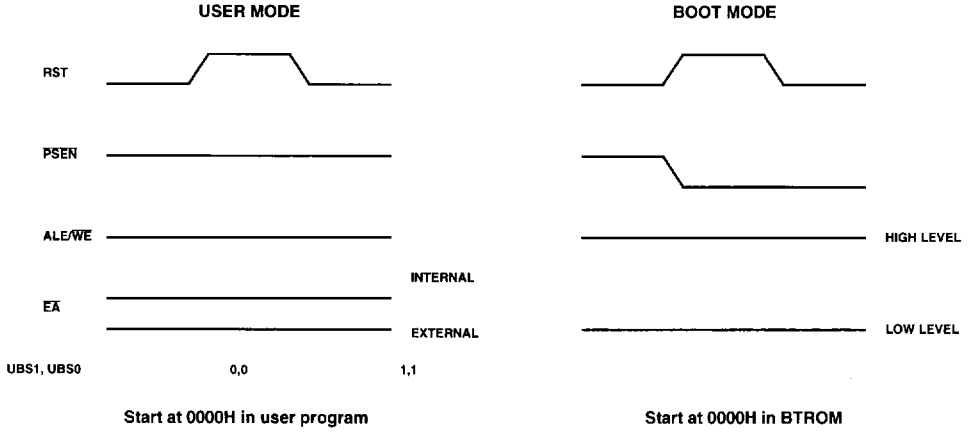


Figure 31. Program memory map and operation modes

UBS1	UBS0	User - Boot selection bits
0	0	User memory mapped from 0 to 64k
0	1	User memory mapped from 0 to 63k Boot ROM mapped from 63k to 64k
1	0	User memory mapped from 0 to 63k, but UBS1 bit cleared by hardware in this user address range. Boot ROM mapped from 63k to 64k.
1	1	Boot ROM mapped from 0 to 64k.

HV	**High voltage indication bit.** Read only. Is 1 as long as the high voltage for an erase or write operation is present.

FCB3	FCB2	FCB1	FCB0	Function Code Bits
0	0	0	0	Value after reset.
0	1	0	1	Byte Write or byte read (verify)
1	1	0	0	Page Erase (32 byte boundaries).
0	0	1	1	Block Erase (256 byte boundaries).
1	0	1	0	Full Erase (32k byte).

The four FCB bits are by write protected if the security feature is activated. Then only instructions in the internal program memory (FEEPROM) are able to write FCB3-0, boot ROM and external program memory instructions cannot change FCB3-0 except the full erase code can be loaded.

The duration of a write or erase operation is determined by the FEEPROM timer. This timer includes a zero point RC oscillator and cannot be controlled by software.

For calling a user routine in the Boot ROM first all interrupts must be disabled and the DPTR and A have to be loaded with the desired values. After setting UBS0 = 1 and UBS1 = 0 and selecting the function via FCB-bits the respective user routine will be called.

The table below lists the boot ROM user routines, which can be called by the user program. The content of FMCON, A and DPTR before calling is by (IN) and by (OUT) after described return. The boot ROM user routines do not change other registers or Data memory.

X = don't care or not defined
V = verified byte (read back)
1) = 5 LSB's of DPTR are don't care
2) = 5 LSB's of DPTR are "0"
3) = 8 LSB's of DPTR are don't care
4) = 8 LSB's of DPTR contain 08H.

Example of user software (internal or external) that calls the Page Erase routine in the boot-ROM to erase a page in the FEEPROM (32 bytes) starting at address location 1260H.

```
CLR EA                ; Disable all interrupts
MOV DPTR, # 1260H     ; Load page-address
MOV FMCON, # 4CH      ; Load Page-Erase code
LCALL 0FFAAH          ; Call Page-Erase routine
                      ; in boot-ROM (5 ms)
SETB EA               ; Enable interrupts again
```

Example of user software (internal or external) that calls the Byte-Write routine in the boot-ROM to write the content of R5 into the FEEPROM address location 1263H.

```
CLK EA                ; Disable all interrupts
MOV DPTR, # 1263H     ; Load byte address
MOV A, R5             ; Load byte to be written
MOV FMCON, # 45H      ; Load byte-write code
LCALL 0FFADH          ; Call byte-write routine
                      ; in boot-ROM (2.5 ms)
SETB EA ; Enable interrupts again
XRL A, R5             ; Compare the "read-back" byte
JNZ                   ; Jump if verify error
```

BOOT-ROM ROUTINE	CALL ADDRESS	FMCON (IN)	FMCON (OUT)	A (IN)	A (OUT)	DPTR (IN)	DPTR (OUT)
BYTE_READ	FFBAH	45H	15H	XXH	BYTE	BYTE ADDRESS	BYTE ADDRESS
BYTE_WRITE	FFADH	45H	15H	BYTE	BYTE (V)	BYTE ADDRESS	BYTE ADDRESS
PAGE_ERASE	FFAAH	46H	16H	XXH	08H	PAGE ADDRESS 1)	PAGE ADDRESS 2)
BLOCK_ERASE	FFA5H	43H	13H	XCXH	02H	BLOCK ADDRESS 3)	BLOCK ADDRESS 4)
FULL_ERASE	FFA0H	4AH	1AH	XXH	0AH	XXXXH	0018H

3.4 Security

The security feature protects against software piracy and prevents that the content of the FEEPROM can be read undesirable. The Security Byte is located in the highest address location 7FFFH of the FEEPROM and is the same as all other bytes of the FEEPROM.

The Security Byte should be 01010000B to activate and 00H or FFH to activate the security feature. This security code is chosen in such a way that single bit failures will not change the security mode.

If the security feature is deactivated, then there are no access restrictions to the FEEPROM.

If the security feature is activated, then the external program memory has no access to the FEEPROM with the MOVC instructions. Also bits FCB3-0 of FMCON cannot be written by external program code or boot ROM code. This prevents in-circuit programming and verification. Only the Full Erase code can be written to FCB0-3 of FMCON. At the end of a full erase operation the security feature is deactivated. Also parallel programming and verify is inhibited if the security feature is activated, only a full erase is possible. Note that the security mode does not change immediately when the security code is written into the security byte 7FFFH, but after a reset or power-on. This allows the verification of the loaded code in the FEEPROM, including the Security Byte.

3.5 Parallel programming

Unlike standard EPROM programming, no high programming supply voltage must be applied to the $\overline{EA}$ pin and only one programming pulse must be applied to the ALE/$\overline{WE}$ pin. The parallel programming mode is entered with the steady signals RST=1, PSEN=0, $\overline{EA}$=1 and SELTAL1 = 1. The XTAL1, 2 clock must have a frequency between 4 and 6MHz. The following table shows the logic levels for programming, erasing, verifying and read signature.

MODE	ALE/$\overline{WE}$	P2.7	P2.6	P3.7	P3.6
Full erase	⊓⊔	1	1	0	1
Program FEEPROM	⊓⊔	1	0	1	1
Verifiy FEEPROM	1	0	0	1	1
Read signature	1	0	0	0	0

ALE/$\overline{WE}$ Write Enable signal (program/erase), active low

P2.6, P2.7, P3.6, P3.7 output enable signal for verify/read modes, active low

Data and address bits:

P0.0-P0.7: D0 - D7 Program data input / verify or read data output

P1.0-P1. : A0 - A7 Input low order address bits.

P2.0-P2.5, P3.4 : A8 - A14 Input high order address bits.

The 89CE558 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. These bytes are read by the same procedure as for a normal verification of locations 30H and 31H, except that P3.6 and P3.7 need to be pulled to LOW.

ADDRESS	CONTENT	MEANING
30H	15H	Philips
31H	B5H	89CE558

The 89CE558 has a Security Byte in location 7FFFH (the highest address) of the FEEPROM, which should be programmed to 01010000B to activate the security feature.

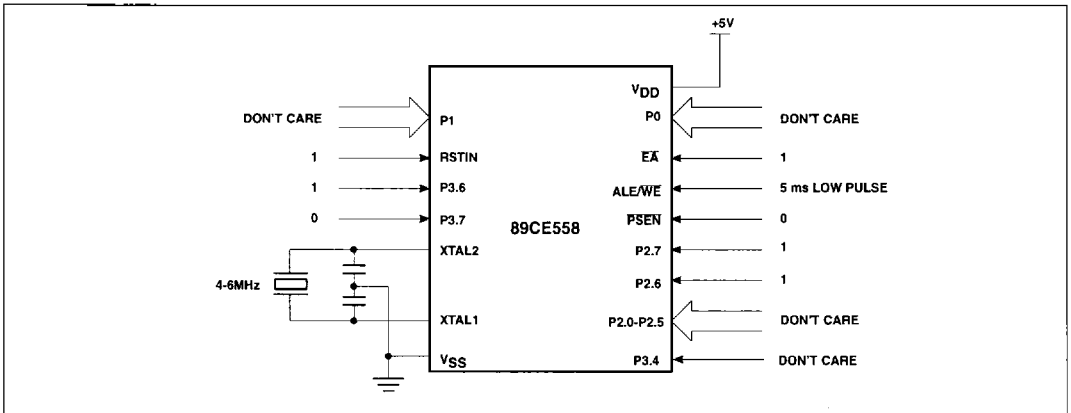


Figure 32. Erase Configuration

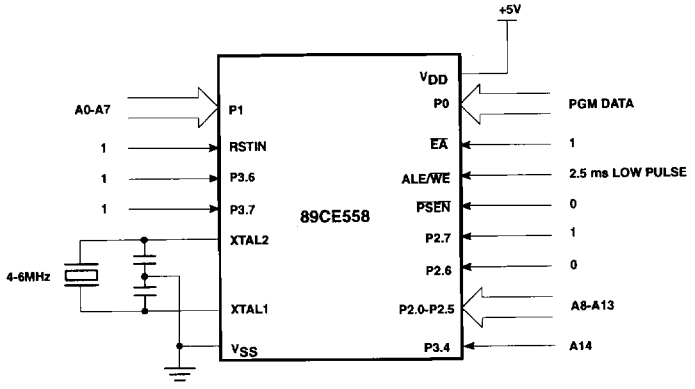


Figure 33. Programming Configuration

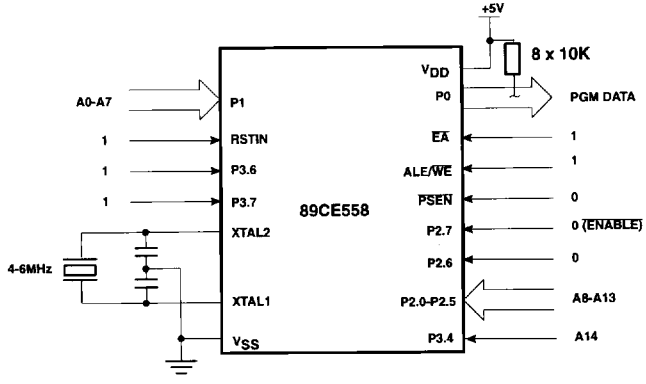


Figure 34. Program Verification

FEEPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T_{amb} = -40°C to +85°C, V_{DD} = 5V±10%, V_{SS} = 0V (See Figure 35)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$1/t_{CLCL}$	Oscillator frequency (standard oscillator)	4	6	MHz
t_{AVWL}	Address setup to $\overline{WE}$ LOW	$48t_{CLCL}$	-	
t_{WHAX}	Address hold after $\overline{WE}$ HIGH	$48t_{CLCL}$	-	
t_{DVWL}	Data setup to $\overline{WE}$ LOW	$48t_{CLCL}$	-	
t_{WHDX}	Data hold after $\overline{WE}$ HIGH	$48t_{CLCL}$	-	
t_{EHWL}	P2.7 ($\overline{ENABLE}$) HIGH to $\overline{WE}$ LOW	$48t_{CLCL}$	-	
t_{WHEL}	$\overline{WE}$ HIGH to P2.7 ($\overline{ENABLE}$) LOW	$48t_{CLCL}$	-	
t_{WLWHp}	$\overline{WE}$ width (programming)	2.25	2.75	ms
t_{WLWHe}	$\overline{WE}$ width (erase)	4.5	5.5	ms
t_{AVQV}	Address to data valid	-	$48t_{CLCL}$	
t_{ELQV}	P2.7 ($\overline{ENABLE}$) Low to data valid	-	$48t_{CLCL}$	
t_{EHQZ}	Data float after P2.7 ($\overline{ENABLE}$) HIGH	0	$48t_{CLCL}$	

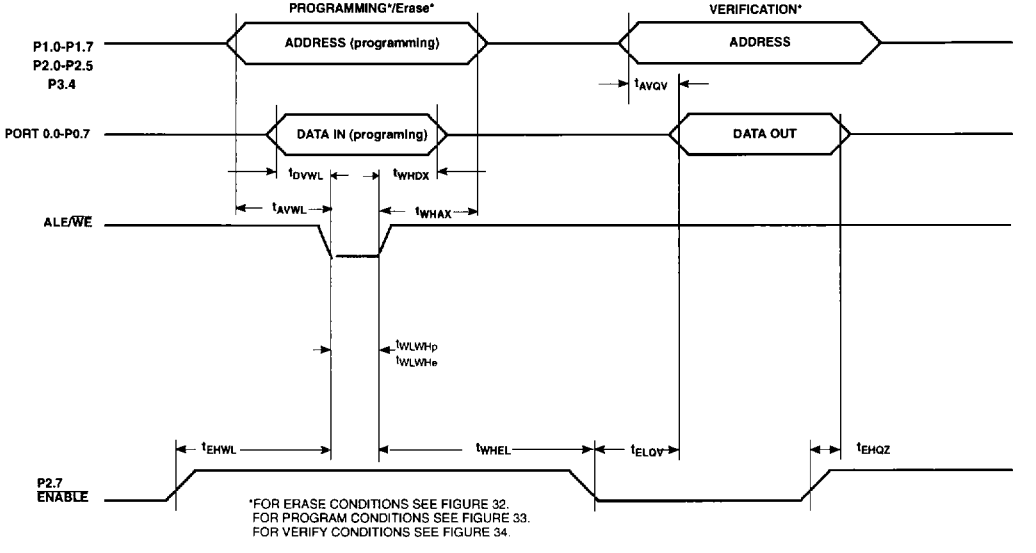


Figure 35. FEEPROM Programming/Erase and Verification Waveforms

3.6 Serial programming of FEEPROM

Serial in-circuit programming (boot-mode) is entered if during and after RESET $\overline{PSEN}$ and $\overline{EA}$ are pulled down, $\overline{PSEN}$ via a resistor of 2 kOhm to VSS. The two UBS bits are set to 1 by hardware and program execution starts at 0000H of the boot ROM. P3.0 (RXD) and P3.1 (TXD) form the serial RS232 interface. A baud rate of 4800 or 9600 Baud is possible after reset (µC clock is 11.01 MHz) if the PLL oscillator is selected. The receive and transmit channel have the same baud rate. The boot routine inputs the Intel Hex Object Format. The baud rate will be selected automatically after reception of the first character (:) of the object file. No other characters are allowed to precede the first (:) character. Programming is only started if the first received record has the right type indication (TT). If the security feature is activated then the programming starts with a Full Erase, otherwise only the addressed page(s) will be erased and the not alterated bytes are rewritten. During the erase or write operation the next string of bytes can be received. Xon and Xoff handshake codes are used to control the serial transfer. At the end of the programming a message that indicates a successful or not successful programming, will be returned over the RS232 interface channel. If the programming was successful then the user program can be started up at 0000H in FEEPROM by a reset for user mode ($\overline{EA}$ = high, $\overline{PSEN}$ not affected). If the programming was not successful the boot program halts and a retry can be started by a reset for the boot mode.

3.6 Boot Routine

The boot routine transmits the next "one ASCII character" messages via the RS232 interface:

- " . " After each record type TT = 00H indication in the HEX file.
- " X " Checksum error of a record in the HEX file detected.
- " Y " Wrong record type received
- " Z " Buffer overflow error (Check Xon/Xoff of terminal)
- " R " Verification error (of last written byte)
- " V " End record received and programming of FEEPROM was successful

No messages are transmitted if the baud rate of the first character (:) can not be detected.

The boot routine can also be started by the internal or external user program (LJMP FC07H). FMCON must be loaded previously with 40H. Interrupt registers, stack pointer, Timer 0, UART, P3.0 and P3.1 must be in the reset state. $\overline{EA}$ and $\overline{PSEN}$ must not be affected. A reset is needed to restart the user program after programming.

The next baudrates will be detected automatically within the specified µC clock range in MHz.

Baudrate	f_{CLK} (min)	f_{CLK} (max)
1200	1 [1)]	3.6
2400	2 [1)]	7.3
4800	4	19.7 [11)]
9600	7.9	29.5 [1)]
19200	15.7	59 [1)]

Note 1): value outside the specified clock range

Note that the boot routines can (re) program any number of bytes from 1 byte to 32k bytes, independent in which order or at which location.

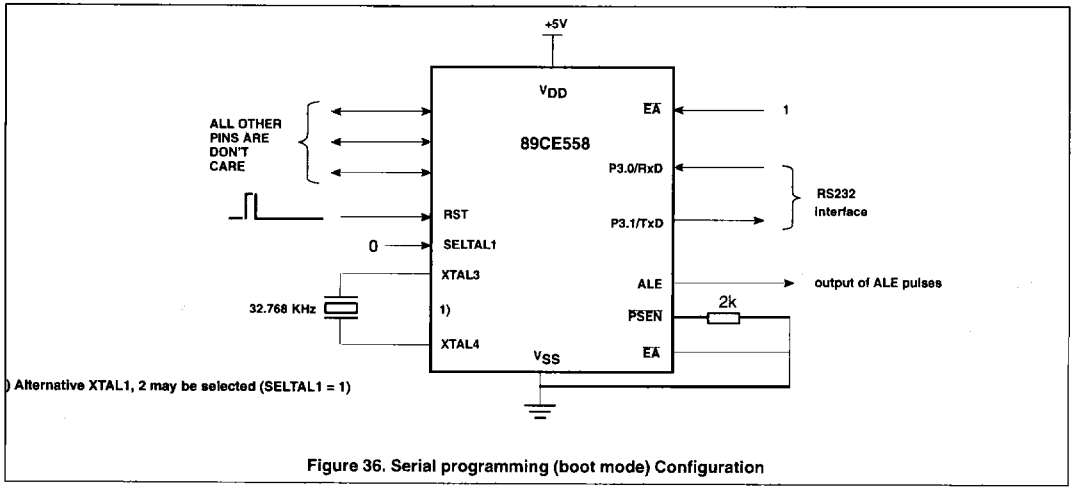


Figure 36. Serial programming (boot mode) Configuration

NOTE: In the following, each letter corresponds to one hexadecimal digit in ASCII representation (the digits 0 through 9, and the letters A through F).

Definitions:

: - Record start character

BC - Byte Count. The hexadecimal number of data bytes in the record. This may theoretically be any number from 0 to 255 prefer to deal with 16 data bytes per record (as shown in the example below).

AAAA - Load address in hexadecimal of first data byte in this record.

TT - Record type. The record type is 00 for data records and 01 for the end record.

HH - One hexadecimal data byte.

CC - Record checksum. This is the 2's complement of the summation of all of the bytes in the record from the byte count through the last data byte. While the summation is calculated, it is always truncated to a one byte result. Thus, if all of the bytes in the record are summed, including the checksum itself, the result will always be 00 if the record is valid.

Construction of data records (using the notation defined above) is as follows:

:: BCAAAATTHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC

:: BCAAATTHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC

The last record in a file is the end record and contains no data. Usually the end record will appear as shown in the first example below. However, in some cases a 16 bit checksum of all of the data bytes in the entire file may be inserted in the address field of the end record. This checksum would correspond to one generated by an EPROM programmer during file load, and its inclusion does not violate the rules for this format. This is shown in the second example.

:00000001FF

:00B12C0122

Successive hex records need not appear in sequential address order . For instance, a record for address 0000 might appear after a record for address 7FE0. All of the bytes in a single record, however, must be in sequence. Any characters that appear outside of a record (i.e. after a checksum, but before the next ":") should be ignored, if present.

An example of a valid hex file follows:

:10010000C2F0E53030E704F404D2F08531F030F786

:100110000763F0FF05F0B2F0A430F00A63F0FFF4DB

:0C0120002401500205F085F032F5332276

:00000001FF

4.0 LIMITING VALUES

ABSOLUTE MAXIMUM RATINGS[1, 2, 3]

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on V_{DD} to V_{SS}	-0.5 to +6.5	V
Voltage on any other pin to V_{SS}	-0.5 to V_{dd} + 0.5	V
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	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 is 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 are taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

5.0 DC CHARACTERISTICS

DC ELECTRICAL CHARACTERISTICS

V_{DD} = 5V (± 10%), V_{SS} = 0V, T_{amb} = **0°C to +70°C** (8xCE558EBx). All voltages with respect to V_{SS} unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS MIN	LIMITS MAX	UNIT
V_{DD}	Supply voltage		4.5	5.5	V
I_{DD}	Supply current operating : 89CE558 83CE558	See notes 1 and 2 f_{CLK} = 16MHz V_{DD} = 5.5 V		 50 40	 mA mA
I_{ID}	Idle mode : 89CE558 83CE558	See notes 1 and 3 f_{CLK} = 16MHz V_{DD} = 5.5 V		 10 8	 mA mA
I_{PD}	Power-down current	See note 4 2V < V_{PD} < V_{DDmax}		100	µA
Inputs					
V_{IL}	Input LOW voltage, except $\overline{EA}$, SCL, SDA		-0.5	$0.2V_{DD}$-0.1	V
V_{IL1}	Input LOW voltage to $\overline{EA}$		-0.5	$0.2V_{DD}$-0.3	V
V_{IL2}	Input LOW voltage to SCL, SDA [5]		-0.5	$0.3V_{DD}$	V
V_{IH}	Input HIGH voltage, except XTAL1, RST, SCL, SDA		$0.2V_{DD}$+0.9	V_{DD}+0.5	V
V_{IH1}	Input HIGH voltage, XTAL1, RST		$0.7V_{DD}$	V_{DD}+0.5	V
V_{IH2}	Input HIGH voltage, SCL, SDA [5]		$0.7V_{DD}$	6.0	V
I_{IL}	Input current LOW level, Ports 0, 1, 2, 3, 4	V_{IN} = 0.45V		-50	µA
I_{TL}	Transition current HIGH to LOW, Ports 0, 1, 2, 3, 4	See note 6		-650	µA
$\pm I_{IL1}$	Input leakage current, Port 0, $\overline{EA}$, ADEXS, $\overline{EW}$	0.45V < V_I < V_{DD}		10	µA
$\pm I_{IL2}$	Input leakage current, SCL, SDA	0V < V_I < 6V 0V < V_{DD} < 5.5V		10	µA
$\pm I_{IL3}$	Input leakage current, Port 5	0.45V < V_I < V_{DD}		1	µA
Outputs					
V_{OL}	Output low voltage, Ports 1, 2, 3, 4	I_{OL} = 1.6mA [7,17]		0.45	V
V_{OL1}	Output low voltage, Port 0, ALE, $\overline{PSEN}$, $\overline{PWM0}$, $\overline{PWM1}$	I_{OL} = 3.2mA [7,17]		0.45	V
V_{OL2}	Output low voltage, SCL, SDA	I_{OL} = 3.0mA [7,17]		0.4	V
V_{OH}	Output high voltage, Ports 1, 2, 3, 4	V_{DD}=5V±10% $-I_{OH}$ = 60µA $-I_{OH}$ = 25µA $-I_{OH}$ = 10µA	 2.4 $0.75V_{DD}$ $0.9V_{DD}$		 V V V
V_{OH1}	Output high voltage (Port 0 in external bus mode, ALE, $\overline{PSEN}$, RST, $\overline{PWM0}$, $\overline{PWM1}$) [8]	V_{DD}=5V±10% $-I_{OH}$ = 800µA $-I_{OH}$ = 300µA $-I_{OH}$ = 80µA	 2.4 $0.75V_{DD}$ $0.9V_{DD}$		 V V V
R_{RST}	Internal reset pull-down resistor		50	150	kΩ
C_{IO}	Pin capacitance	Test freq = 1MHz, T_{amb} = 25°C		10	pF

DC ELECTRICAL CHARACTERISTICS (Continued)

V_{DD} = 5V (± 10%), V_{SS} = 0V, T_{amb} = **-40°C to +85°C** (8xCE558EFx). DC parameters not included here are the same as in the 8xCE558EBB, DC electrical characteristics
All voltages with respect to V_{SS} unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS MIN	LIMITS MAX	UNIT
Inputs					
V_{IL}	Input LOW voltage, except $\overline{EA}$, SCL, SDA		-0.5	$0.2V_{DD}$-0.15	V
V_{IL1}	Input LOW voltage to $\overline{EA}$		-0.5	$0.2V_{DD}$-0.35	V
V_{IH}	Input HIGH voltage, except XTAL1, RST, SCL, SDA		$0.2V_{DD}$+1.0	V_{DD}+0.5	V
V_{IH1}	Input HIGH voltage, XTAL1, RST		$0.7V_{DD}$+0.1	V_{DD}+0.5	V
I_{IL}	Input current LOW level, Ports 1, 2, 3, 4	V_{IN} = 0.45V		-75	µA
I_{TL}	Transition current HIGH to LOW, Ports 1, 2, 3, 4	See note 6		-750	µA

DC ELECTRICAL CHARACTERISTICS ANALOG

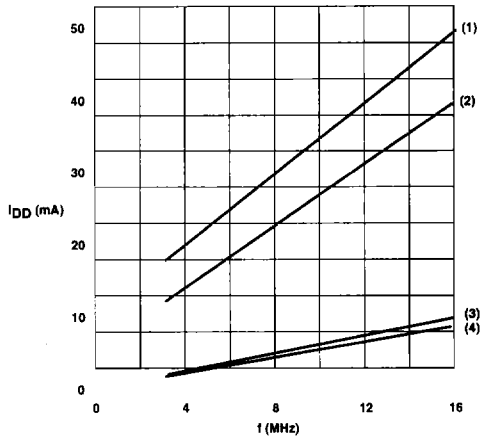
AV_{DD} = 5V (± 10%), AV_{SS} = 0V, T_{amb} = 0°C to +70°C (8xCE558EBx).
AV_{DD} = 5V (± 10%), AV_{SS} = 0V, T_{amb} = -40°C to +85°C (8xCE558EFx).
All voltages with respect to V_{SS} unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS MIN	LIMITS MAX	UNIT
AV_{DD}	Analog supply voltage	$AV_{DD} = V_{DD} \pm 0.2V$	4.5	5.5	V
AI_{DD}	Analog supply current	Port 5 = 0 to AV_{DD} see note 19		1.2	mA
AI_{ID}	Idle mode	see note 18		50	µA
AI_{PD}	Power-down mode	2V < AV_{PD} < AV_{DD} see note 18		50	µA
Analog Inputs					
AV_{IN}	Analog input voltage		AV_{SS}-0.2	AV_{DD}+0.2	V
AV_{REF}	Reference voltage: AV_{REF-} AV_{REF+}		AV_{SS}-0.2	 AV_{DD}+0.2	 V V
R_{REF}	Resistance between AV_{REF+} and AV_{REF-}		10	50	kΩ
C_{IA}	Analog input capacitance			15	pF
DL_e	Differential non-linearity [9, 10, 11]			±1	LSB
IL_e	Integral non-linearity [9, 12]			±2	LSB
OS_e	Offset error [9, 13]			±2	LSB
G_e	Gain error [9, 14]			±0.4	%
A_e	Absolute voltage error [9, 15]			±3	LSB
M_{CTC}	Channel to channel matching			±1	LSB
C_t	Crosstalk between inputs of port 5 [16]	0-100kHz		-60	dB

NOTES FOR DC ELECTRICAL CHARACTERISTICS:

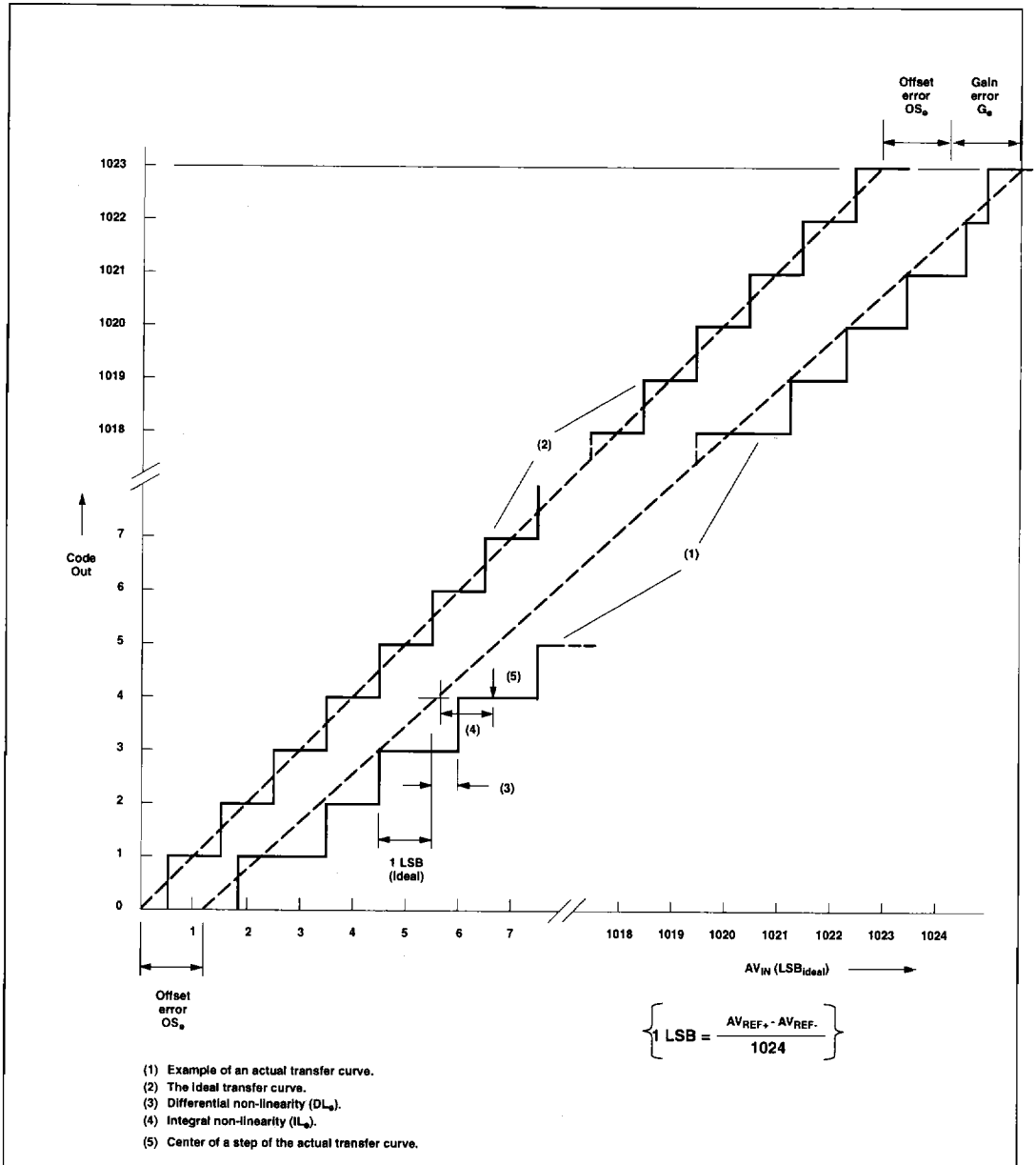
1. See Figures 37 and 39 through 41 for I_{DD} test conditions.
2. The operating supply current is measured with all output pins disconnected; XTAL1 driven with $t_r = t_f = 5ns$; $V_{IL} = V_{SS} + 0.5V$; $V_{IH} = V_{DD} - 0.5V$; XTAL2 not connected; $\overline{EA}$ = RST = Port 0 = $\overline{EW}$ = SCL = SDA = V_{DD}; ADEXS = V_{SS}.
3. The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with $t_r = t_f = 5ns$; $V_{IL} = V_{SS} + 0.5V$; $V_{IH} = V_{DD} - 0.5V$; XTAL2 not connected; Port 0 = $\overline{EW}$ = SCL = SDA = SELXTAL 1 = V_{DD}; $\overline{EA}$ = RST = ADEXS = V_{SS}.
4. The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = $\overline{EW}$ = SCL = SDA = SELXTAL 1 = V_{DD}; $\overline{EA}$ = RST = ADEXS = XTAL1 = V_{SS}.
5. The input threshold voltage of SCL and SDA (SIO1) meets the I^2C specification, so an input voltage below 0.3 V_{DD} will be recognized as a logic 0 while an input voltage above 0.7 V_{DD} will be recognized as a logic 1.
6. Pins of ports 1, 2, 3, and 4 source a transition current when they are being externally driven from HIGH to LOW. The transition current reaches its maximum value when V_{IN} is approximately 2V.
7. Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V_{OL} of ALE and ports 1, 3 and 4. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
8. Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the $0.9V_{DD}$ specification when the address bits are stabilizing.
9. Conditions: $AV_{REF-} = 0V$; $AV_{DD} = 5.0V$, $AV_{REF+} = 5.12V$. $V_{DD} = 5.0V$, $V_{SS} = 0V$, ADC is monotonic with no missing codes. Measurement by continuous conversion of AV_{IN} = -20mV to 5.12V in steps of 0.5mV, derivating parameters from collected conversion results of ADC. ADC prescaler programmed according to the actual oscillator frequency, resulting in a conversion time within the specified range for t_{conv} (15µs ... 50µs).
10. The differential non-linearity (DL_e) is the difference between the actual step width and the ideal step width.
11. The ADC is monotonic; there are no missing codes.
12. The integral non-linearity (IL_e) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error.
13. The offset error (OS_e) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error), and a straight line which fits the ideal transfer curve. The offset error is constant at every point of the actual transfer curve.
14. The gain error (G_e) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error), and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve.
15. The absolute voltage error (A_e) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve.
16. This should be considered when both analog and digital signals are simultaneously input to port 5.
17. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
    Maximum I_{OL} per port pin: 10 mA
    Maximum I_{OL} 8-bit port -
    Port 0 : 26 mA
    Ports 1, 2, 3 and 4 : 15 mA
    Maximum total I_{OL} for all output pins : 86 mA

    If I_{OL} exceeds the test conditions, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
18. AI_{ID} is measured with SELXTAL1 = V_{DD}.
19. AI_{ID} and AI_{PD} are measured with the 32 kHz oscillator being halted, XTAL3 not connected and XTAL4 = V_{SS}.



(1) Maximum operating mode 89CE558 : V_{DD} = 5.5V
(2) Maximum operating mode 83CE558/80CE558 : V_{DD} = 5.5V
(3) Maximum idle mode 89CE558 : V_{DD} = 5.5V
(4) Maximum idle mode;83CE558/80CE558 : V_{DD} = 5.5V

Figure 37. Supply Current (I_{DD}) as a Function of Frequency at XTAL1



(1) Example of an actual transfer curve.
(2) The ideal transfer curve.
(3) Differential non-linearity (DL_e).
(4) Integral non-linearity (IL_e).
(5) Center of a step of the actual transfer curve.

Figure 38. ADC Conversion Characteristic

6.0 AC CHARACTERISTICS

AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5V ± 10% (EBx), V_{SS} = 0V, T_{CLCL} min = 1/fmax (maximum operating frequency)
V_{DD} = 5V ± 10% (EFx), V_{SS} = 0V, T_{CLCL} min = 1/fmax (maximum operating frequency)
T_{amb} = 0°C to +70°C, T_{CLCL} min = 63 ns for 8xCE558EBx
T_{amb} = -40°C to +85°C, T_{CLCL} min = 63 ns for 8xCE558EFx
C1 = 100 pF for Port 0, ALE and $\overline{PSEN}$; C1 = 80 pF for all other outputs unless otherwise specified.

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	42	Oscillator					3.5	16	MHz
t_{LHLL}	42	ALE pulse width	127		85		$2t_{CLCL}$-40		ns
t_{AVLL}	42	Address valid to ALE LOW	28		8		t_{CLCL}-55		ns
t_{LLAX}	42	Address hold after ALE LOW	48		28		t_{CLCL}-35		ns
t_{LLIV}	42	ALE LOW to valid instruction in		234		150		$4t_{CLCL}$-100	ns
t_{LLPL}	42	ALE LOW to $\overline{PSEN}$ LOW	43		23		t_{CLCL}-40		ns
t_{PLPH}	42	$\overline{PSEN}$ pulse width	205		143		$3t_{CLCL}$-45		ns
t_{PLIV}	42	$\overline{PSEN}$ LOW to valid instruction in		145		83		$3t_{CLCL}$-105	ns
t_{PXIX}	42	Input instruction hold after $\overline{PSEN}$	0		0		0		ns
t_{PXIZ}	42	Input instruction float after $\overline{PSEN}$		59		38		t_{CLCL}-25	ns
t_{AVIV}	42	Address to valid instruction in		312		208		$5t_{CLCL}$-105	ns
t_{PLAZ}	42	$\overline{PSEN}$ LOW to address float		10		10		10	ns
Data Memory									
t_{LLAX}	43, 44	Address valid to ALE LOW	28		8		t_{CLCL}-55		ns
t_{RLRH}	43	$\overline{RD}$ pulse width	400		275		$6t_{CLCL}$-100		ns
t_{WLWH}	44	$\overline{WR}$ pulse width	400		275		$6t_{CLCL}$-100		ns
t_{RLDV}	43	$\overline{RD}$ LOW to valid data in		252		148		$5t_{CLCL}$-165	ns
t_{RHDX}	43	Data hold after $\overline{RD}$	0		0		0		ns
t_{RHDZ}	43	Data float after $\overline{RD}$		97		55		$2t_{CLCL}$-70	ns
t_{LLDV}	43	ALE LOW to valid data in		517		350		$8t_{CLCL}$-150	ns
t_{AVDV}	43	Address to valid data in		585		398		$9t_{CLCL}$-165	ns
t_{LLWL}	43, 44	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	200	300	138	238	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	43, 44	Address valid to $\overline{WR}$ LOW or $\overline{RD}$ LOW	203		120		$4t_{CLCL}$-130		ns
t_{QVWX}	44	Data valid to $\overline{WR}$ transition	23		3		t_{CLCL}-60		ns
t_{DW}	44	Data before $\overline{WR}$	433		288		$7t_{CLCL}$-150		ns
t_{WHQX}	44	Data hold after $\overline{WR}$	33		13		t_{CLCL}-50		ns
t_{RLAZ}	43	$\overline{RD}$ low to address float		0		0		0	ns
t_{WHLH}	43, 44	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	43	123	23	103	t_{CLCL}-40	t_{CLCL}+40	ns
UART Timing - Shift Register Mode (Test Conditions: T_{amb} = 0°C to +70°C; V_{SS} = 0V; Load Capacitance = 80pF)									
t_{XLXL}	46	Serial port clock cycle time	1.0		0.75		$12t_{CLCL}$		µs
t_{QVXH}	46	Output data setup to clock rising edge	700		492		$10t_{CLCL}$-133		ns
t_{XHQX}	46	Output data hold after clock rising edge	50		8		$2t_{CLCL}$-117		ns
t_{XHDX}	46	Input data hold after clock rising edge	0		0		0		ns
t_{XHDV}	46	Clock rising edge to input data valid		700		492		$10t_{CLCL}$-133	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	Standard-mode I²C-bus		UNIT
		MIN	MAX	
I²C Interface timing (refer to Figure 45)				
f_{SCL}	SCL clock frequency	0	100	kHz
t_{BUF}	Bus free time between a STOP and START condition	4.7	-	µs
$t_{HD;\ STA}$	Hold time (repeated) START condition. After this period, the first clock pulse is generated	4.0	-	µs
t_{LOW}	LOW period of the SCL clock	4.7	-	µs
t_{HIGH}	High period of the SCL clock	4.0	-	µs
$t_{SU;\ STA}$	Set-up time for a repeated START condition	4.7	-	µs
$t_{HD;\ DAT}$	Data hold time	0[1]		µs
$t_{HD;\ DAT1,2,3}$	Data set-up time	250	-	ns
t_{RD}, t_{RC}	Rise time of both SDA and SCL signals	-	1000	ns
t_{FD}, t_{FC}	Fall time of both SDA and SCL signals	-	300	ns
$t_{SU;\ STO}$	Set-up time for STOP condition	4.0	-	µs
C_b	Capacitive load for each bus line	-	400	pF

All values referred to V_{IH} and $V_{IL\ max}$ levels.

NOTES:

1. A device must internally provide a hold time of at least 300 ns from the SDA signal (referred to the $V_{IH\ min}$ of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.

For 62ns < t_{CLCL} < 570ns, 16MHz > f_{CLK} > 3.5MHz) the SI01 interface meets concerning AC parameters the I²C-bus specification for bit-rates up to 100 kbit/s.

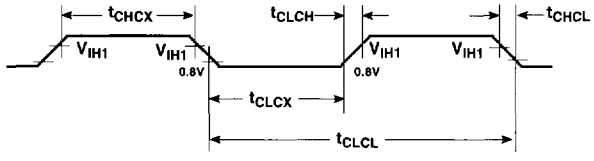
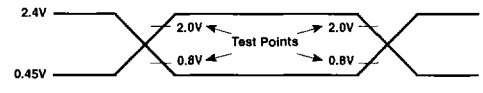
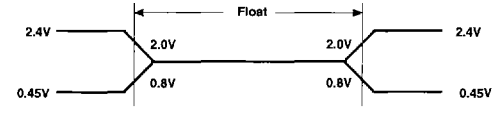


Figure 39. External Clock Drive waveform



NOTE:
AC inputs during testing are driven at 2.4V for a logic 'HIGH' and 0.45V for a logic 'LOW'. Timing measurements are made at 2.0V for a logic 'HIGH' and 0.8V for a logic 'LOW'.

Figure 40. AC Testing Input/Output



NOTE:
The float state is defined as the point at which a port 0 pin sinks 3.2mA or sources 400µA at the voltage test levels.

Figure 41. AC Testing, Float Waveform

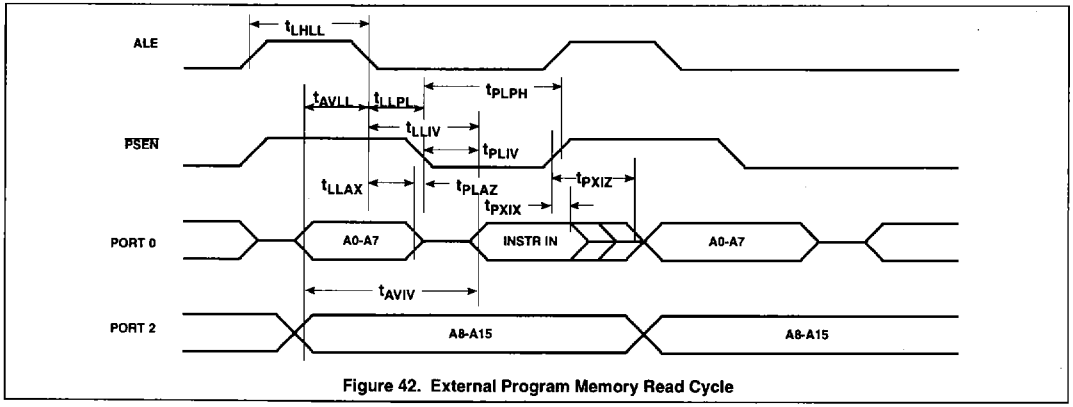


Figure 42. External Program Memory Read Cycle

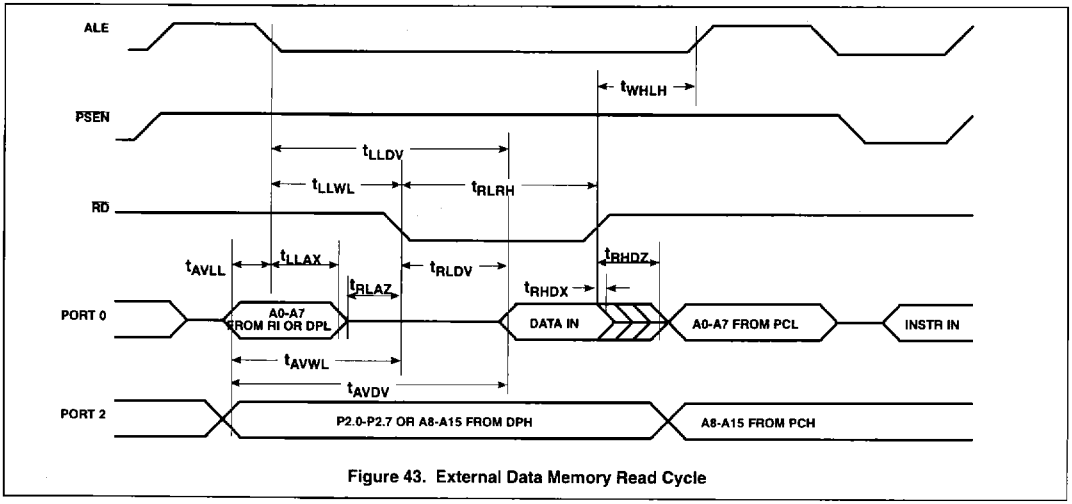


Figure 43. External Data Memory Read Cycle

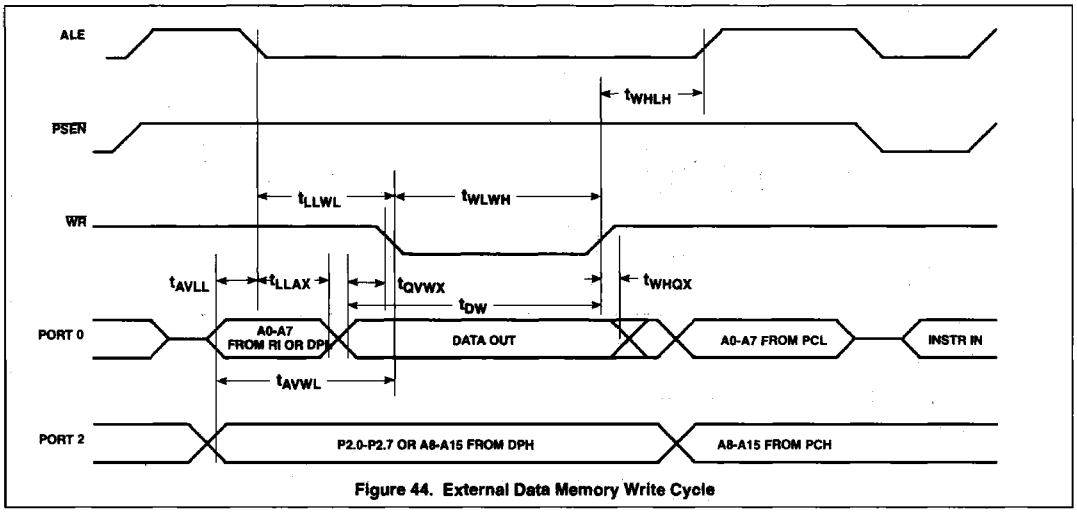


Figure 44. External Data Memory Write Cycle

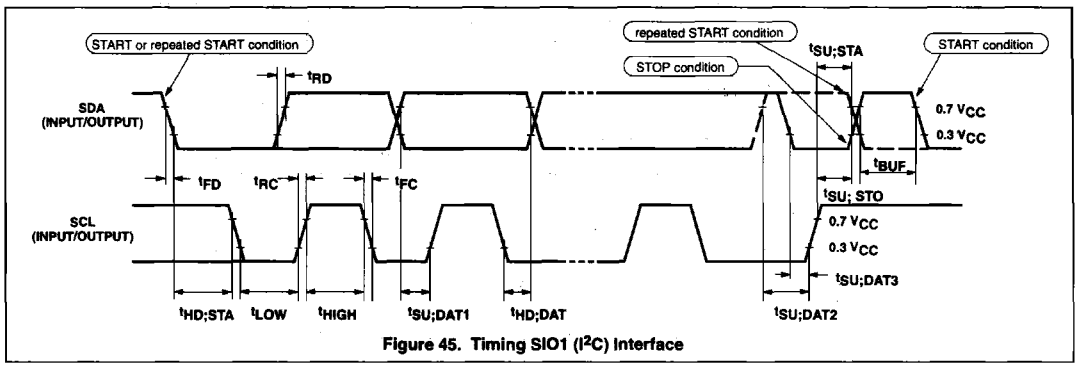


Figure 45. Timing SIO1 (I^2C) Interface

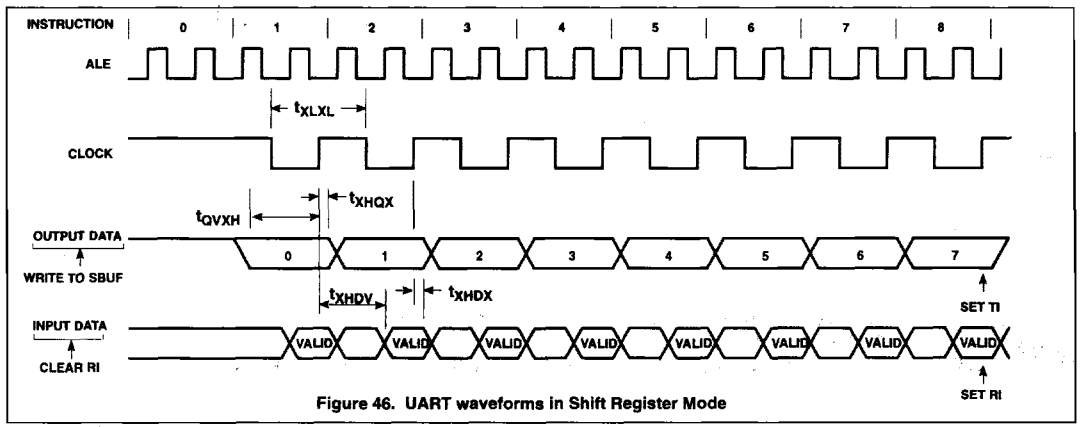


Figure 46. UART waveforms in Shift Register Mode

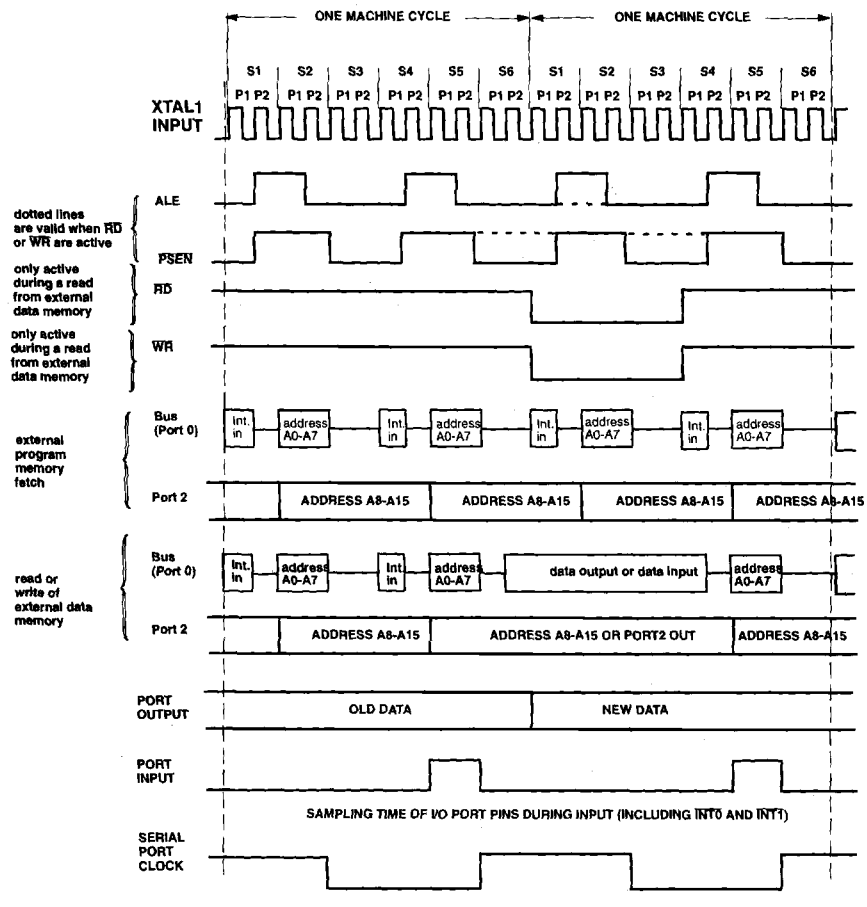


Figure 47. Instruction cycle timing

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.