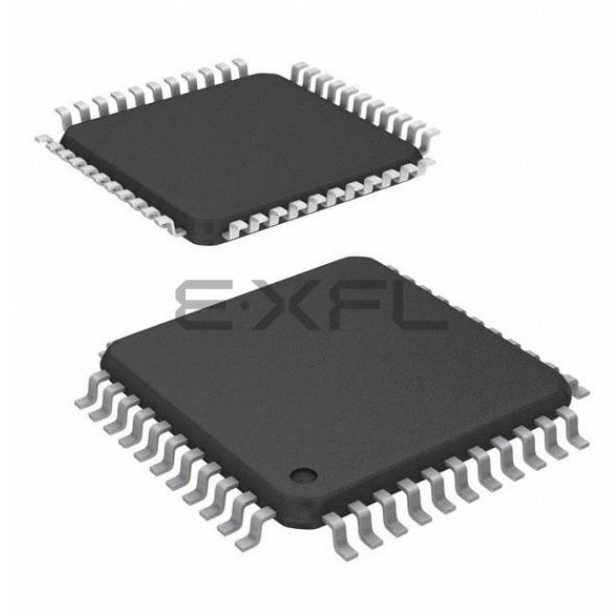


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

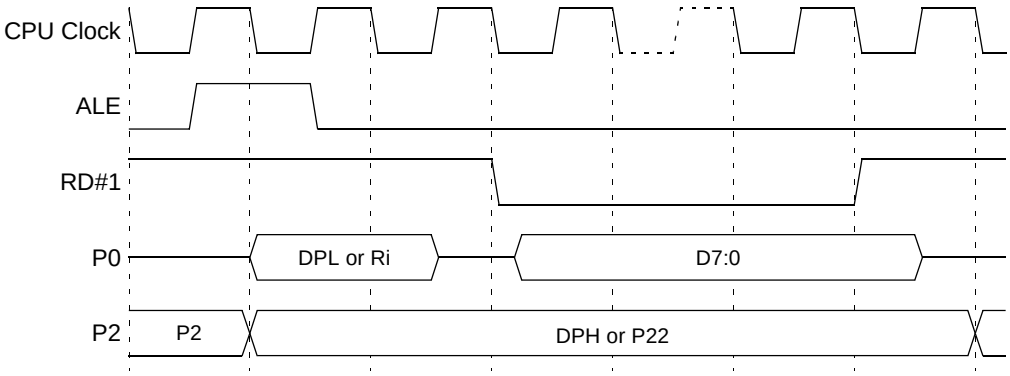
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	80C51
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-VQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51cc03u-rltim

Pin Name	Type	Description
VSS	GND	**Circuit ground**
TESTI	I	**Must be connected to VSS**
VCC		**Supply Voltage**
VAREF		Reference Voltage for ADC
VAGND		Reference Ground for ADC
P0.0:7	I/O	**Port 0:** Is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pull-ups when emitting 1's. Port 0 also outputs the code Bytes during program validation. External pull-ups are required during program verification.
P1.0:7	I/O	**Port 1:** Is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins can be used for digital input/output or as analog inputs for the Analog Digital Converter (ADC). Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 1 pins that are being pulled low externally will be the source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 1 pins are assigned to be used as analog inputs via the ADCCF register (in this case the internal pull-ups are disconnected). As a secondary digital function, port 1 contains the Timer 2 external trigger and clock input; the PCA external clock input and the PCA module I/O. P1.0/AN0/T2 Analog input channel 0, External clock input for Timer/counter2. P1.1/AN1/T2EX Analog input channel 1, Trigger input for Timer/counter2. P1.2/AN2/ECI Analog input channel 2, PCA external clock input. P1.3/AN3/CEX0 Analog input channel 3, PCA module 0 Entry of input/PWM output. P1.4/AN4/CEX1 Analog input channel 4, PCA module 1 Entry of input/PWM output. P1.5/AN5/CEX2 Analog input channel 5, PCA module 2 Entry of input/PWM output. P1.6/AN6/CEX3 Analog input channel 6, PCA module 3 Entry of input/PWM output. P1.7/AN7/CEX4 Analog input channel 7, PCA module 4 Entry ot input/PWM output. Port 1 receives the low-order address byte during EPROM programming and program verification. It can drive CMOS inputs without external pull-ups.
P2.0:7	I/O	**Port 2:** Is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 2 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 2 emits the high-order address byte during accesses to the external Program Memory and during accesses to external Data Memory that uses 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1's. During accesses to external Data Memory that use 8 bit addresses (MOVX @Ri), Port 2 transmits the contents of the P2 special function register. It also receives high-order addresses and control signals during program validation. It can drive CMOS inputs without external pull-ups.

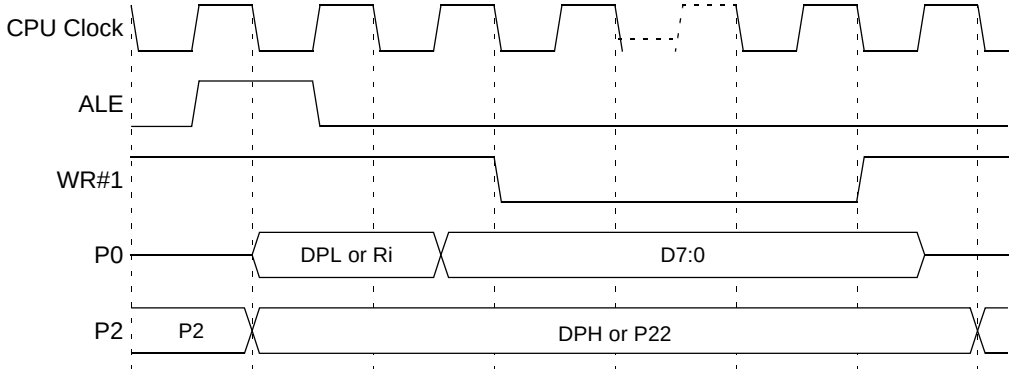
Pin Name	Type	Description
P3.0:7	I/O	**Port 3:** Is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 3 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. The output latch corresponding to a secondary function must be programmed to one for that function to operate (except for TxD and $\overline{WR}$). The secondary functions are assigned to the pins of port 3 as follows: P3.0/RxD: Receiver data input (asynchronous) or data input/output (synchronous) of the serial interface P3.1/TxD: Transmitter data output (asynchronous) or clock output (synchronous) of the serial interface P3.2/$\overline{INT0}$: External interrupt 0 input/timer 0 gate control input P3.3/$\overline{INT1}$: External interrupt 1 input/timer 1 gate control input P3.4/T0: Timer 0 counter input P3.5/T1/$\overline{SS}$: Timer 1 counter input SPI Slave Select P3.6/$\overline{WR}$: External Data Memory write strobe; latches the data byte from port 0 into the external data memory P3.7/$\overline{RD}$: External Data Memory read strobe; Enables the external data memory. It can drive CMOS inputs without external pull-ups.
P4.0:4	I/O	**Port 4:** Is an 2-bit bi-directional I/O port with internal pull-ups. Port 4 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 4 pins that are being pulled low externally will be a source of current (IIL, on the datasheet) because of the internal pull-up transistor. The output latch corresponding to a secondary function RxDC must be programmed to one for that function to operate. The secondary functions are assigned to the two pins of port 4 as follows: P4.0/TxDC: Transmitter output of CAN controller P4.1/RxDC: Receiver input of CAN controller. P4.2/MISO: Master Input Slave Output of SPI controller P4.3/SCK: Serial Clock of SPI controller P4.4/MOSI: Master Ouput Slave Input of SPI controller It can drive CMOS inputs without external pull-ups.

Figure 11. External Data Read Waveforms



Notes: 1. RD# signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.

Figure 12. External Data Write Waveforms



Notes: 1. WR# signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.

Registers

Table 6. PSW Register

PSW (S:8Eh)
Program Status Word Register

7	6	5	4	3	2	1	0
CY	**AC**	**F0**	**RS1**	**RS0**	**OV**	**F1**	**P**

Bit Number	Bit Mnemonic	Description
7	CY	**Carry Flag** Carry out from bit 1 of ALU operands.
6	AC	**Auxiliary Carry Flag** Carry out from bit 1 of addition operands.
5	F0	**User Definable Flag 0.**
4-3	RS1:0	**Register Bank Select Bits** Refer to Table 4 for bits description.
2	OV	**Overflow Flag** Overflow set by arithmetic operations.
1	F1	**User Definable Flag 1**
0	P	**Parity Bit** Set when ACC contains an odd number of 1's. Cleared when ACC contains an even number of 1's.

Reset Value = 0000 0000b

Table 7. AUXR Register

AUXR (S:8Eh)
Auxiliary Register

7	6	5	4	3	2	1	0
-	**-**	**M0**	**XRS2**	**XRS1**	**XRS0**	**EXTRAM**	**A0**

Bit Number	Bit Mnemonic	Description
7-6	-	**Reserved** The value read from these bits are indeterminate. Do not set this bit.
5	M0	**Stretch MOVX control:** the RD/ and the WR/ pulse length is increased according to the value of M0. **M0** **Pulse length in clock period** 0 6 1 30

Table 9. Pin Conditions in Special Operating Modes

Mode	Port 0	Port 1	Port 2	Port 3	Port 4	ALE	PSEN#
Reset	Floating	High	High	High	High	High	High
Idle (internal code)	Data	Data	Data	Data	Data	High	High
Idle (external code)	Floating	Data	Data	Data	Data	High	High
Power-Down(internal code)	Data	Data	Data	Data	Data	Low	Low
Power-Down (external code)	Floating	Data	Data	Data	Data	Low	Low

Cross Flash Memory Access

Code executing from		Action	FM0 (user Flash)	FM1 (boot Flash)
	FM0 (user Flash)	Read	ok	-
		Load column latch	ok	-
		Write	-	-
	FM1 (boot Flash)	Read	ok	ok
		Load column latch	ok	-
		Write	ok	-
	External memory EA = 0	Read	(a)	-
		Load column latch	-	-
		Write	-	-

(a) Depend upon general lock bit configuration.

Operation Cross Memory Access

Space addressable in read and write are:

- RAM
- ERAM (Expanded RAM access by movx)
- XRAM (eXternal RAM)
- EEPROM DATA
- FM0 (user flash)
- Hardware byte
- XROW
- Boot Flash
- Flash Column latch

The table below provide the different kind of memory which can be accessed from different code location.

Table 18. Cross Memory Access

	Action	RAM	XRAM ERAM	Boot FLASH	FM0	E² Data	Hardware Byte	XROW
boot FLASH	Read			OK	OK	OK	OK	-
	Write			-	OK(1)	OK(1)	OK(1)	OK(1)
FM0	Read			OK	OK	OK	OK	-
	Write			-	OK (idle)	OK(1)	-	OK
External memory EA = 0 or Code Roll Over	Read			-	-	OK	-	-
	Write			-	-	OK(1)	-	-

Note: 1. RWW: Read While Write

Boot Loader Jump Bit (BLJB):
- This bit indicates if on RESET the user wants to jump to this application at address @0000h on FM0 or execute the boot loader at address @F800h on FM1.
- BLJB = 0 on parts delivered with bootloader programmed.
- To read or modify this bit, the APIs are used.

Boot Vector Address (SBV):
- This byte contains the MSB of the user boot loader address in FM0.
- The default value of SBV is FCh (no user boot loader in FM0).
- To read or modify this byte, the APIs are used.

Extra Byte (EB) and Boot Status Byte (BSB):
- These Bytes are reserved for customer use.
- To read or modify these Bytes, the APIs are used.

Hardware Boot Process

At the falling edge of RESET, the bit ENBOOT in AUXR1 register is initialized with the value of Boot Loader Jump Bit (BLJB).

Further at the falling edge of RESET if the following conditions (called Hardware condition) are detected:

- PSEN low,
- EA high,
- ALE high (or not connected).
  - After Hardware Condition the FCON register is initialized with the value 00h and the PC is initialized with F800h (FM1).

The Hardware condition makes the bootloader to be executed, whatever BLJB value is.

If no hardware condition is detected, the FCON register is initialized with the value F0h.

Check of the BLJB value.

- If bit BLJB = 1:
  User application in FM0 will be started at @0000h (standard reset).
- If bit BLJB = 0:
  Boot loader will be started at @F800h in FM1.

Note:
1. As PSEN is an output port in normal operating mode (running user applications or bootloader applications) after reset it is recommended to release PSEN after the falling edge of Reset is signaled.
   The hardware conditions are sampled at reset signal Falling Edge, thus they can be released at any time when reset input is low.
2. To ensure correct microcontroller startup, the PSEN pin should not be tied to ground during power-on.

4. Interrupt routine

// Save the current CANPAGE

// Find the first message object which generate an interrupt in CANSIT1 and CANSIT2

// Select the corresponding message object

// Analyse the CANSTCH register to identify which kind of interrupt is generated

// Manage the interrupt

// Clear the status register CANSTCH = 00h;

// if it is not a channel interrupt but a general interrupt
// Manage the general interrupt and clear CANGIT register

// restore the old CANPAGE

Table 54. CANEN1 Register

CANEN1 (S:CEh Read Only)
CAN Enable Message Object Registers 1

7	6	5	4	3	2	1	0
-	ENCH14	ENCH13	ENCH12	ENCH11	ENCH10	ENCH9	ENCH8

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The values read from this bit is indeterminate. Do not set this bit.
6-0	ENCH14:8	**Enable Message Object** These bits provide the availability of the MOb. It is set to one when the MOb is enabled. Once TXOK or RXOK is set to one (TXOK for automatic reply), the corresponding ENMOB is reset. ENMOB is also set to zero configuring the MOb in disabled mode, applying abortion or standby mode. 0 - message object disabled: MOb available for a new transmission or reception. 1 - message object enabled: MOb in use. This bit is resetable by re-writing the CANCONCH of the corresponding message object.

Reset Value = x000 0000b

Table 55. CANEN2 Register

CANEN2 (S:CFh Read Only)
CAN Enable Message Object Registers 2

7	6	5	4	3	2	1	0
ENCH7	ENCH6	ENCH5	ENCH4	ENCH3	ENCH2	ENCH1	ENCH0

Bit Number	Bit Mnemonic	Description
7-0	ENCH7:0	**Enable Message Object** These bits provide the availability of the MOb. It is set to one when the MOb is enabled. Once TXOK or RXOK is set to one (TXOK for automatic reply), the corresponding ENMOB is reset. ENMOB is also set to zero configuring the MOb in disabled mode, applying abortion or standby mode. 0 - message object disabled: MOb available for a new transmission or reception. 1 - message object enabled: MOb in use. This bit is resetable by re-writing the CANCONCH of the corresponding message object.

Reset Value = 0000 0000b

Error Conditions

The following flags in the SPSCR register indicate the SPI error conditions:

Mode Fault Error (MODF)

Mode Fault error in Master mode SPI indicates that the level on the Slave Select ($\overline{SS}$) pin is inconsistent with the actual mode of the device.

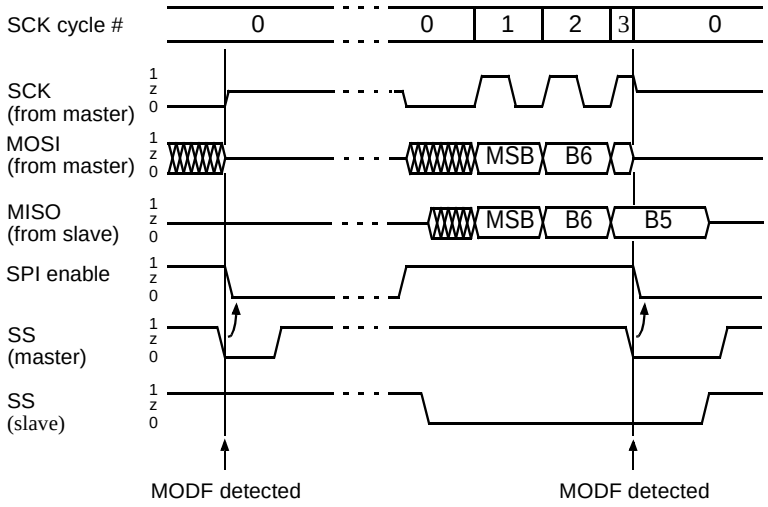
- Mode fault detection in Master mode:

MODF is set to warn that there may be a multi-master conflict for system control. In this case, the SPI system is affected in the following ways:

  – An SPI receiver/error CPU interrupt request is generated
  – The SPEN bit in SPCON is cleared. This disables the SPI
  – The MSTR bit in SPCON is cleared

Clearing the MODF bit is accomplished by a read of SPSCR register with MODF bit set, followed by a write to the SPCON register. SPEN Control bit may be restored to its original set state after the MODF bit has been cleared.

Figure 64. Mode Fault Conditions in Master Mode (Cpha =’1’/Cpol =’0’)



Note: When SS is discarded (SS disabled) it is not possible to detect a MODF error in master mode because the SPI is internally unselected and the SS pin is a general purpose I/O.

- Mode fault detection in Slave mode

In slave mode, the MODF error is detected when SS goes high during a transmission.
A transmission begins when SS goes low and ends once the incoming SCK goes back to its idle level following the shift of the eighteen data bit.
A MODF error occurs if a slave is selected (SS is low) and later unselected (SS is high) even if no SCK is sent to that slave.

At any time, a ’1’ on the SS pin of a slave SPI puts the MISO pin in a high impedance state and internal state counter is cleared. Also, the slave SPI ignores all incoming SCK clocks, even if it was already in the middle of a transmission. A new transmission will be performed as soon as SS pin returns low.

Bit Number	Bit Mnemonic	Description
4	MODF	**Mode Fault** - Set by hardware to indicate that the $\overline{SS}$ pin is in inappropriate logic level (in both master and slave modes). - Cleared by hardware when reading SPSCR When MODF error occurred: - In slave mode: SPI interface ignores all transmitted data while $\overline{SS}$ remains high. A new transmission is perform as soon as SS returns low. - In master mode: SPI interface is disabled (SPEN=0, see description for SPEN bit in SPCON register).
3	SPTE	**Serial Peripheral Transmit register Empty** - Set by hardware when transmit register is empty (if needed, SPDAT can be loaded with another data). - Cleared by hardware when transmit register is full (no more data should be loaded in SPDAT).
2	UARTM	**Serial Peripheral UART mode** Set and cleared by software: - Clear: Normal mode, data are transmitted MSB first (default) - Set: UART mode, data are transmitted LSB first.
1	SPTEIE	**Interrupt Enable for SPTE** Set and cleared by software: - Set to enable SPTE interrupt generation (when SPTE goes high, an interrupt is generated). - Clear to disable SPTE interrupt generation Caution: When SPTEIE is set no interrupt generation occurred when SPIF flag goes high. To enable SPIF interrupt again, SPTEIE should be cleared.
0	MODFIE	**Interrupt Enable for MODF** Set and cleared by software: - Set to enable MODF interrupt generation - Clear to disable MODF interrupt generation

Reset Value = 00X0 XXXXb

Not Bit addressable

Serial Peripheral DATa Register (SPDAT)

The Serial Peripheral Data Register (Table 94) is a read/write buffer for the receive data register. A write to SPDAT places data directly into the shift register. No transmit buffer is available in this model.

A Read of the SPDAT returns the value located in the receive buffer and not the content of the shift register.

Table 94. SPDAT Register

SPDAT - Serial Peripheral Data Register (0D6H)

7	6	5	4	3	2	1	0
R7	R6	R5	R4	R3	R2	R1	R0

Reset Value = Indeterminate

R7:R0: Receive data bits

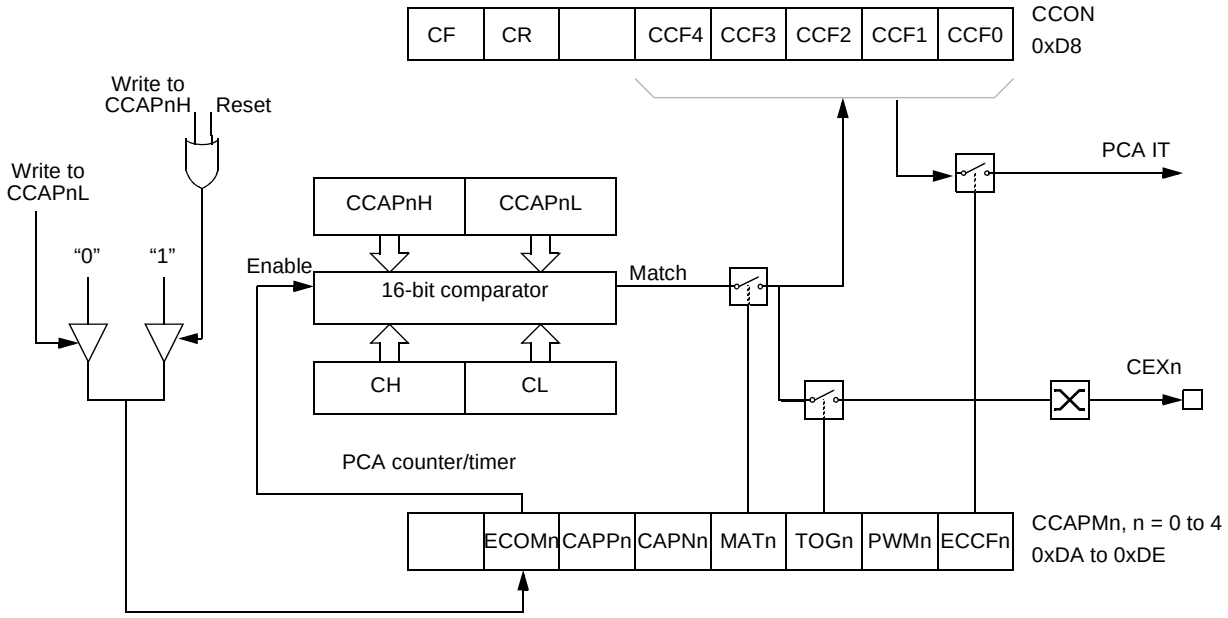
SPCON, SPSTA and SPDAT registers may be read and written at any time while there is no on-going exchange. However, special care should be taken when writing to them while a transmission is on-going:

- Do not change SPR2, SPR1 and SPR0
- Do not change CPHA and CPOL
- Do not change MSTR
- Clearing SPEN would immediately disable the peripheral
- Writing to the SPDAT will cause an overflow.

High Speed Output Mode

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set.

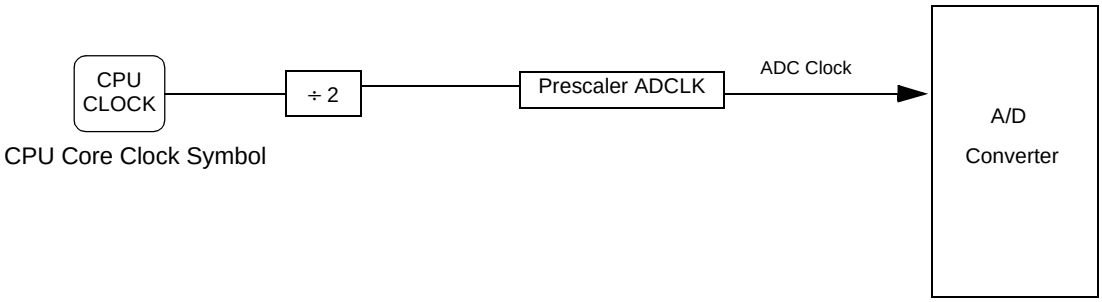
Figure 71. PCA High Speed Output Mode



Pulse Width Modulator Mode

All the PCA modules can be used as PWM outputs. The output frequency depends on the source for the PCA timer. All the modules will have the same output frequency because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than it, the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. the allows the PWM to be updated without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

Figure 75. A/D Converter Clock



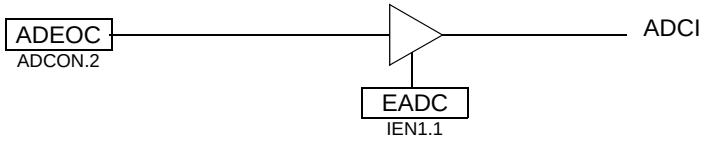
ADC Standby Mode

When the ADC is not used, it is possible to set it in standby mode by clearing bit ADEN in ADCON register. In this mode its power dissipation is about 1 µW.

IT ADC Management

An interrupt end-of-conversion will occurs when the bit ADEOC is activated and the bit EADC is set. For re-arming the interrupt the bit ADEOC must be cleared by software.

Figure 76. ADC Interrupt Structure



Routines examples

1. Configure P1.2 and P1.3 in ADC channels

```
// configure channel P1.2 and P1.3 for ADC
 ADCF = 0Ch

// Enable the ADC
 ADCON = 20h
```

2. Start a standard conversion

```
// The variable "channel" contains the channel to convert
// The variable "value_converted" is an unsigned int
// Clear the field SCH[2:0]
 ADCON and = F8h
// Select channel
 ADCON | = channel
// Start conversion in standard mode
 ADCON | = 08h
// Wait flag End of conversion
 while((ADCON and 01h)! = 01h)
// Clear the End of conversion flag
 ADCON and = EFh
// read the value
 value_converted = (ADDH << 2)+(ADDL)
```

3. Start a precision conversion (need interrupt ADC)

```
// The variable "channel" contains the channel to convert
// Enable ADC
```

External Program Memory Characteristics

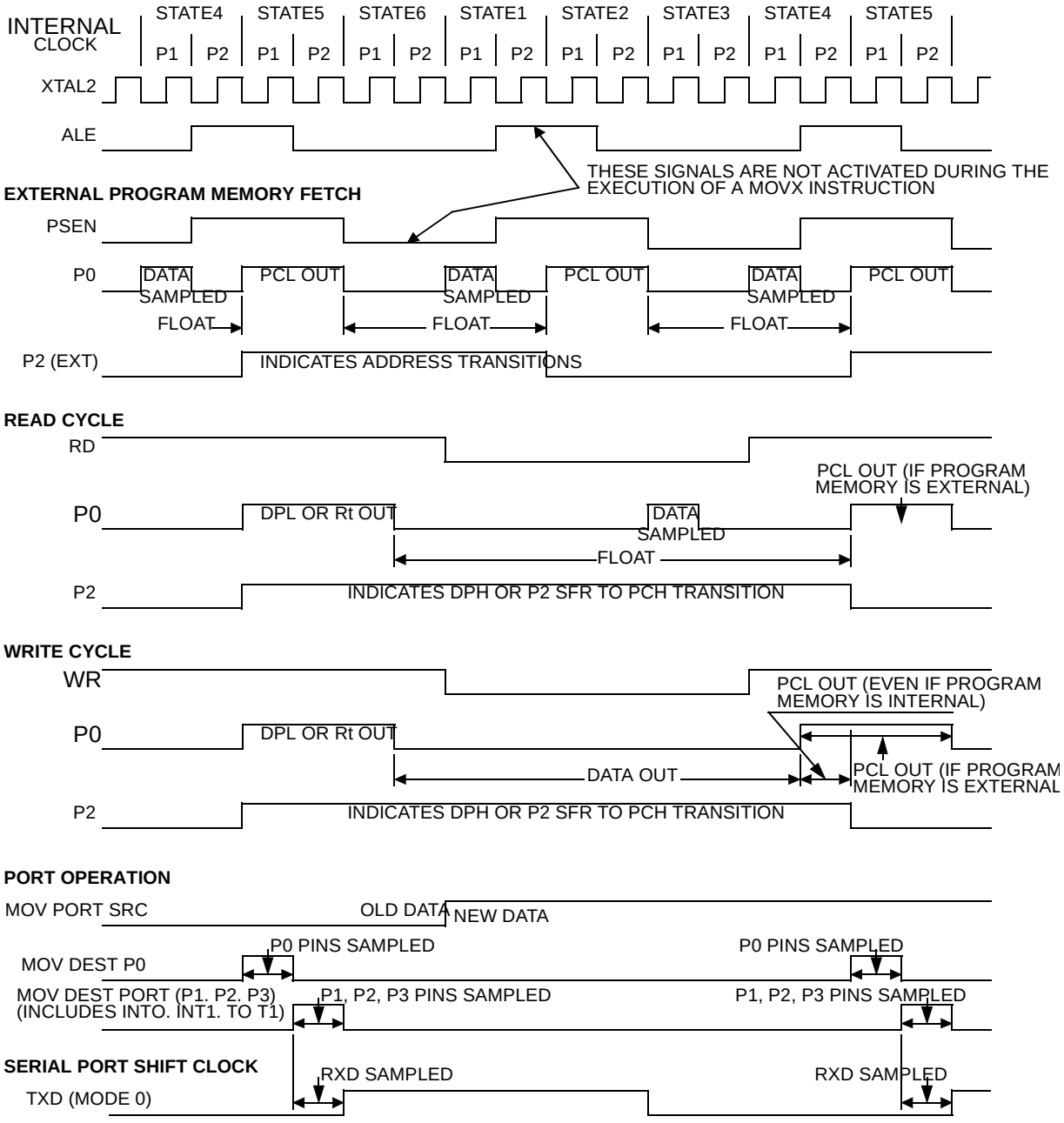
Table 118. Symbol Description

Symbol	Parameter
T	Oscillator clock period
T_{LHLL}	ALE pulse width
T_{AVLL}	Address Valid to ALE
T_{LLAX}	Address Hold After ALE
T_{LLIV}	ALE to Valid Instruction In
T_{LLPL}	ALE to $\overline{PSEN}$
T_{PLPH}	$\overline{PSEN}$ Pulse Width
T_{PLIV}	$\overline{PSEN}$ to Valid Instruction In
T_{PXIX}	Input Instruction Hold After $\overline{PSEN}$
T_{PXIZ}	Input Instruction Float After $\overline{PSEN}$
T_{AVIV}	Address to Valid Instruction In
T_{PLAZ}	$\overline{PSEN}$ Low to Address Float

Table 119. AC Parameters for a Fix Clock (F = 40 MHz)

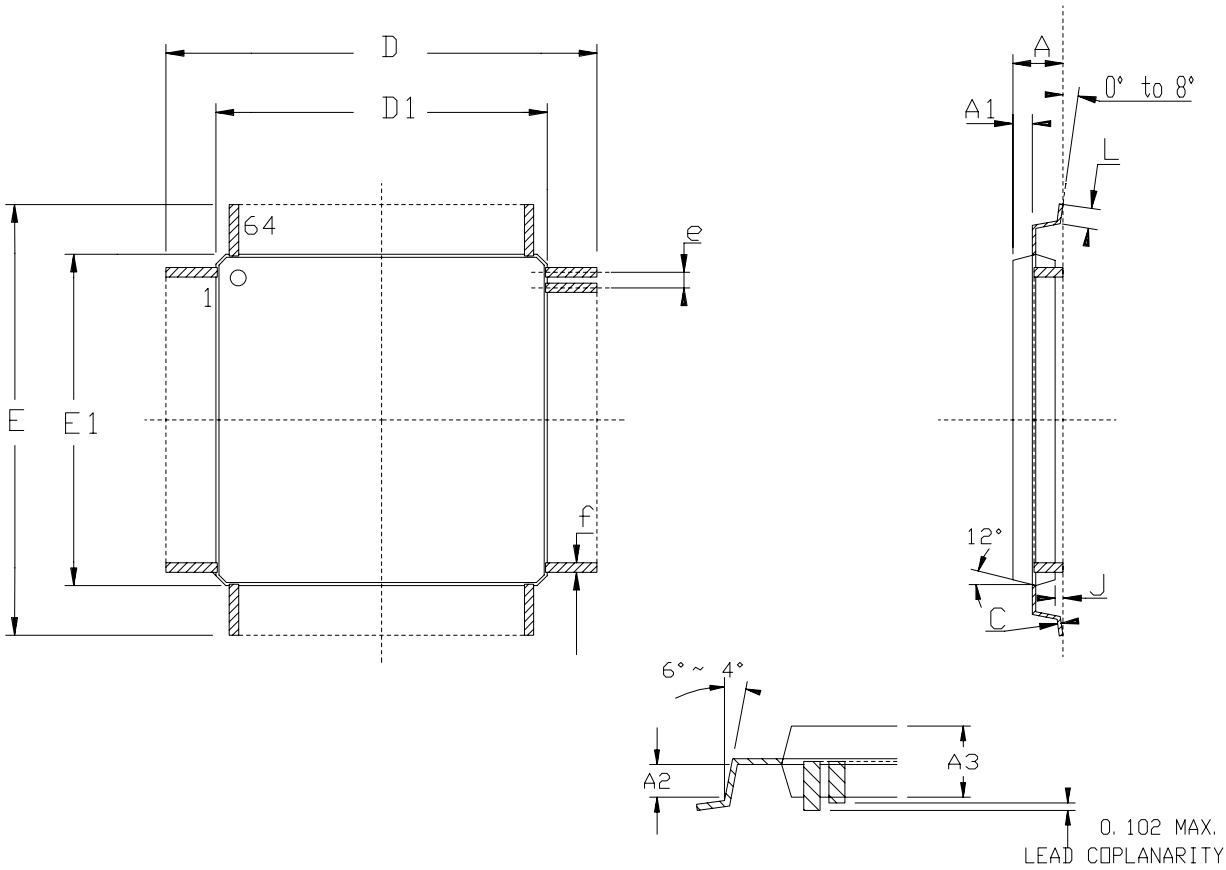
Symbol	Min	Max	Units
T	25		ns
T_{LHLL}	40		ns
T_{AVLL}	10		ns
T_{LLAX}	10		ns
T_{LLIV}		70	ns
T_{LLPL}	15		ns
T_{PLPH}	55		ns
T_{PLIV}		35	ns
T_{PXIX}	0		ns
T_{PXIZ}		18	ns
T_{AVIV}		85	ns
T_{PLAZ}		10	ns

Clock Waveforms Valid in normal clock mode. In X2 mode XTAL2 must be changed to XTAL2/2.



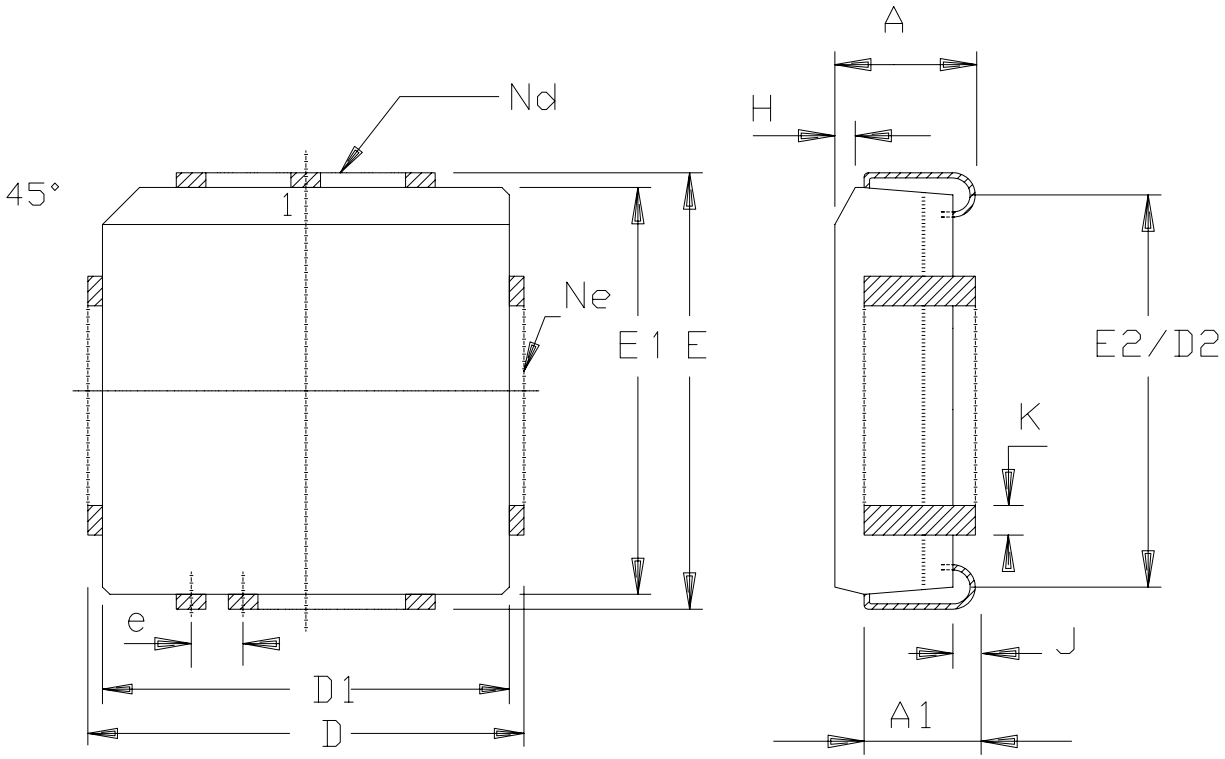
This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A=25°C fully loaded) RD and WR propagation delays are approximately 50ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

VQFP64



	MM		INCH	
	Min	Max	Min	Max
A	–	1.60	–	.063
A1	0.64 REF		.025 REF	
A2	0.64 REF		.025 REF	
A3	1.35	1.45	.053	.057
D	11.75	12.25	.463	.483
D1	9.90	10.10	.390	.398
E	11.75	12.25	.463	.483
E1	9.90	10.10	.390	.398
J	0.05	–	.002	–
L	0.45	0.75	.018	.030
e	0.50 BSC		.0197 BSC	
f	0.25 BSC		.010 BSC	

PLCC52



	MM		INCH	
A	4.20	4.57	.165	.180
A1	2.29	3.30	.090	.130
D	19.94	20.19	.785	.795
D1	19.05	19.25	.750	.758
D2	17.53	18.54	.690	.730
E	19.94	20.19	.785	.795
E1	19.05	19.25	.750	.758
E2	17.53	18.54	.690	.730
e	1.27	BSC	.050	BSC
H	1.07	1.42	.042	.056
J	0.51	–	.020	–
K	0.33	0.53	.013	.021
Nd	13		13	
Ne	13		13	
PKG STD	00			