Microchip Technology - AT89C51CC03UA-RDTUM Datasheet

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	Active
Core Processor	80C51
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, SPI, 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	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51cc03ua-rdtum

Data Memory

The AT89C51CC03 provides data memory access in two different spaces:

1. The internal space mapped in three separate segments:
- the lower 128 Bytes RAM segment.
- the upper 128 Bytes RAM segment.
- the expanded 2048 Bytes RAM segment (ERAM).
2. The external space.

A fourth internal segment is available but dedicated to Special Function Registers, SFRs, (addresses 80h to FFh) accessible by direct addressing mode.

Figure 8 shows the internal and external data memory spaces organization.

Figure 7. Internal Memory - RAM

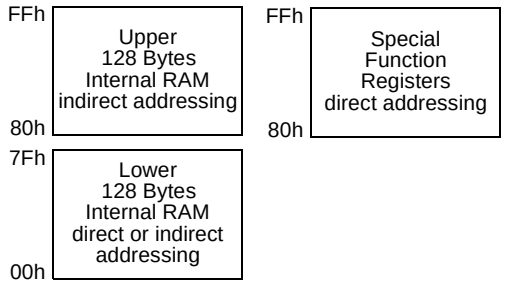
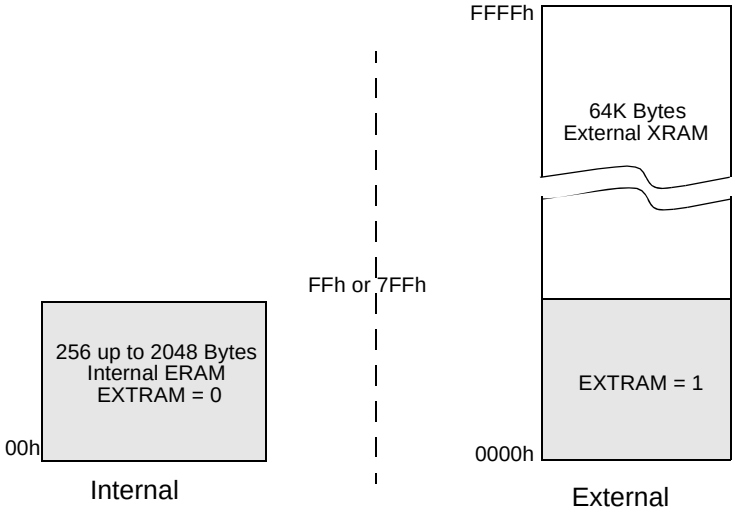


Figure 8. Internal and External Data Memory Organization ERAM-XRAM



Internal Space

Lower 128 Bytes RAM

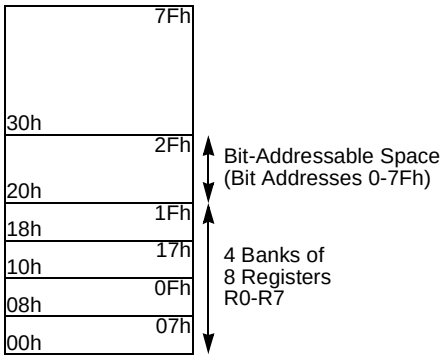
The lower 128 Bytes of RAM (see Figure 8) are accessible from address 00h to 7Fh using direct or indirect addressing modes. The lowest 32 Bytes are grouped into 4 banks of 8 registers (R0 to R7). Two bits RS0 and RS1 in PSW register (see Figure 6) select which bank is in use according to Table 4. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing, and can be used for context switching in interrupt service routines.

Table 4. Register Bank Selection

RS1	RS0	Description
0	0	Register bank 0 from 00h to 07h
0	1	Register bank 0 from 08h to 0Fh
1	0	Register bank 0 from 10h to 17h
1	1	Register bank 0 from 18h to 1Fh

The next 16 Bytes above the register banks form a block of bit-addressable memory space. The C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00h to 7Fh.

Figure 9. Lower 128 Bytes Internal RAM Organization



Upper 128 Bytes RAM

The upper 128 Bytes of RAM are accessible from address 80h to FFh using only indirect addressing mode.

Expanded RAM

The on-chip 2048 Bytes of expanded RAM (ERAM) are accessible from address 0000h to 07FFh using indirect addressing mode through MOVX instructions. In this address range, the bit EXTRAM in AUXR register is used to select the ERAM (default) or the XRAM. As shown in Figure 8 when EXTRAM = 0, the ERAM is selected and when EXTRAM = 1, the XRAM is selected.

The size of ERAM can be configured by XRS2-0 bit in AUXR register (default size is 2048 Bytes).

Note: Lower 128 Bytes RAM, Upper 128 Bytes RAM, and expanded RAM are made of volatile memory cells. This means that the RAM content is indeterminate after power-up and must then be initialized properly.

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

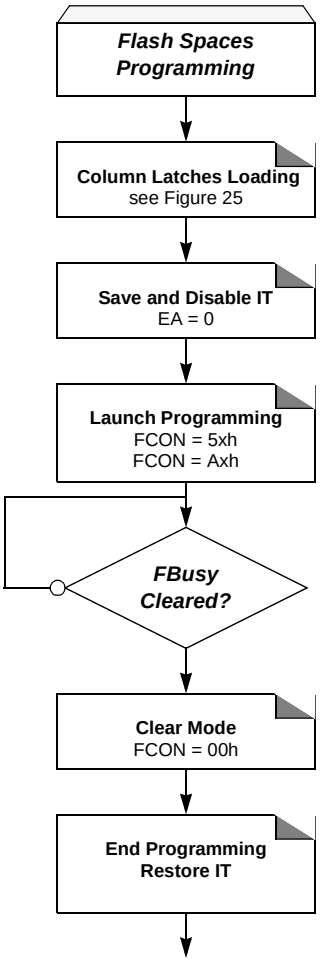
Program/Code Memory

The AT89C51CC03 implement 64K Bytes of on-chip program/code memory. Figure 20 shows the partitioning of internal and external program/code memory spaces depending on the product.

The Flash memory increases EPROM and ROM functionality by in-circuit electrical erasure and programming. Thanks to the internal charge pump, the high voltage needed for programming or erasing Flash cells is generated on-chip using the standard VDD voltage. Thus, the Flash Memory can be programmed using only one voltage and allows In-System Programming commonly known as ISP. Hardware programming mode is also available using specific programming tool.

Figure 20. Program/Code Memory Organization

Figure 26. Flash and Extra Row Programming Procedure

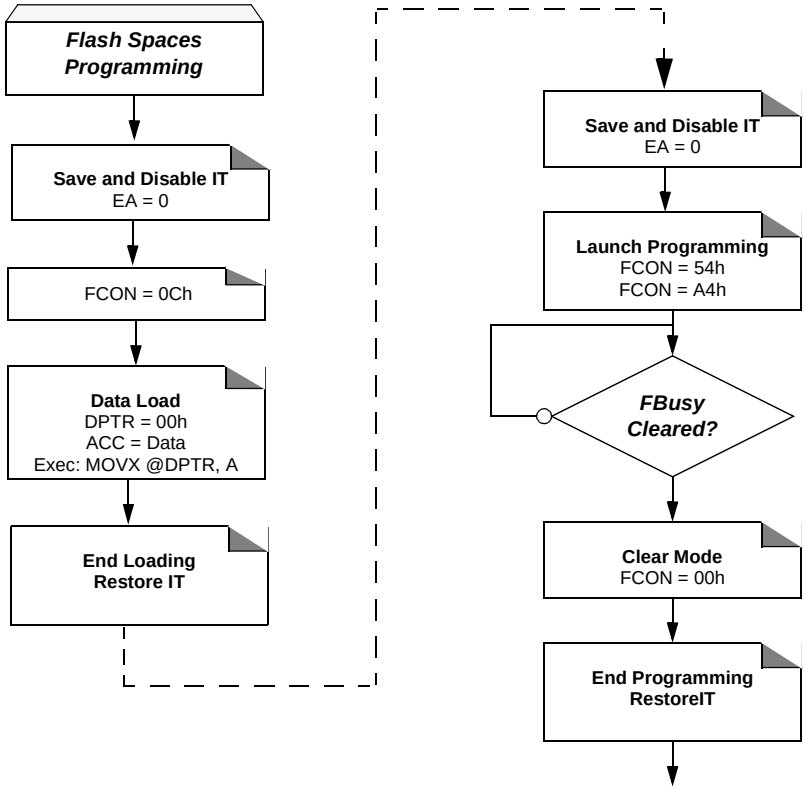


Hardware Security Byte

The following procedure is used to program the Hardware Security Byte space and is summarized in Figure 27:

- Set FPS and map Hardware byte (FCON = 0x0C)
- Save and disable the interrupts.
- Load DPTR at address 0000h.
- Load Accumulator register with the data to load.
- Execute the MOVX @DPTR, A instruction.
- Launch the programming by writing the data sequence 54h followed by A4h in FCON register (only from FM1).
  The end of the programming indicated by the FBusy flag cleared.
- Restore the interrupts.

Figure 27. Hardware Programming Procedure



Reset the Column Latches

An automatic reset of the column latches is performed after a successful Flash write sequence. User can also reset the column latches manually, for instance to reload the column latches before writing the Flash. The following procedure is summarized below.

- Save and disable the interrupts.
- Launch the reset by writing the data sequence 56h followed by A6h in FCON register (only from FM1).
- Restore the interrupts.

Error Reports

Flash Programming Sequence Errors

When a wrong sequence is detected, the SEQERR bit in FSTA register is set. Possible wrong sequence are :

- MOV FCON, 5xh instruction not immediately followed by a MOV FCON, Ax instruction.
- A write Flash sequence is launched while no data were loaded in the column latches

The SEQERR bit can be cleared

- By software
- By hardware when a correct programming sequence is completed

When multiple pages are written into the Flash, the user should check FSTA for errors after each write page sequences, not only at the end of the multiple write pages.

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

Timer 1

Timer 1 is identical to Timer 0 excepted for Mode 3 which is a hold-count mode. The following comments help to understand the differences:

- Timer 1 functions as either a Timer or event Counter in three modes of operation. Figure 35 to Figure 37 show the logical configuration for modes 0, 1, and 2. Timer 1's mode 3 is a hold-count mode.
- Timer 1 is controlled by the four high-order bits of TMOD register (see Figure 31) and bits 2, 3, 6 and 7 of TCON register (see Figure 30). TMOD register selects the method of Timer gating (GATE1), Timer or Counter operation (C/T1#) and mode of operation (M11 and M01). TCON register provides Timer 1 control functions: overflow flag (TF1), run control bit (TR1), interrupt flag (IE1) and interrupt type control bit (IT1).
- Timer 1 can serve as the Baud Rate Generator for the Serial Port. Mode 2 is best suited for this purpose.
- For normal Timer operation (GATE1 = 0), setting TR1 allows TL1 to be incremented by the selected input. Setting GATE1 and TR1 allows external pin INT1# to control Timer operation.
- Timer 1 overflow (count rolls over from all 1s to all 0s) sets the TF1 flag generating an interrupt request.
- When Timer 0 is in mode 3, it uses Timer 1's overflow flag (TF1) and run control bit (TR1). For this situation, use Timer 1 only for applications that do not require an interrupt (such as a Baud Rate Generator for the Serial Port) and switch Timer 1 in and out of mode 3 to turn it off and on.
- It is important to stop Timer/Counter before changing mode.

Mode 0 (13-bit Timer)

Mode 0 configures Timer 1 as a 13-bit Timer, which is set up as an 8-bit Timer (TH1 register) with a modulo-32 prescaler implemented with the lower 5 bits of the TL1 register (see Figure 35). The upper 3 bits of TL1 register are ignored. Prescaler overflow increments TH1 register.

Mode 1 (16-bit Timer)

Mode 1 configures Timer 1 as a 16-bit Timer with TH1 and TL1 registers connected in cascade (see Figure 36). The selected input increments TL1 register.

Mode 2 (8-bit Timer with Auto-Reload)

Mode 2 configures Timer 1 as an 8-bit Timer (TL1 register) with automatic reload from TH1 register on overflow (see Figure 37). TL1 overflow sets TF1 flag in TCON register and reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Mode 3 (Halt)

Placing Timer 1 in mode 3 causes it to halt and hold its count. This can be used to halt Timer 1 when TR1 run control bit is not available i.e. when Timer 0 is in mode 3.

Watchdog Timer

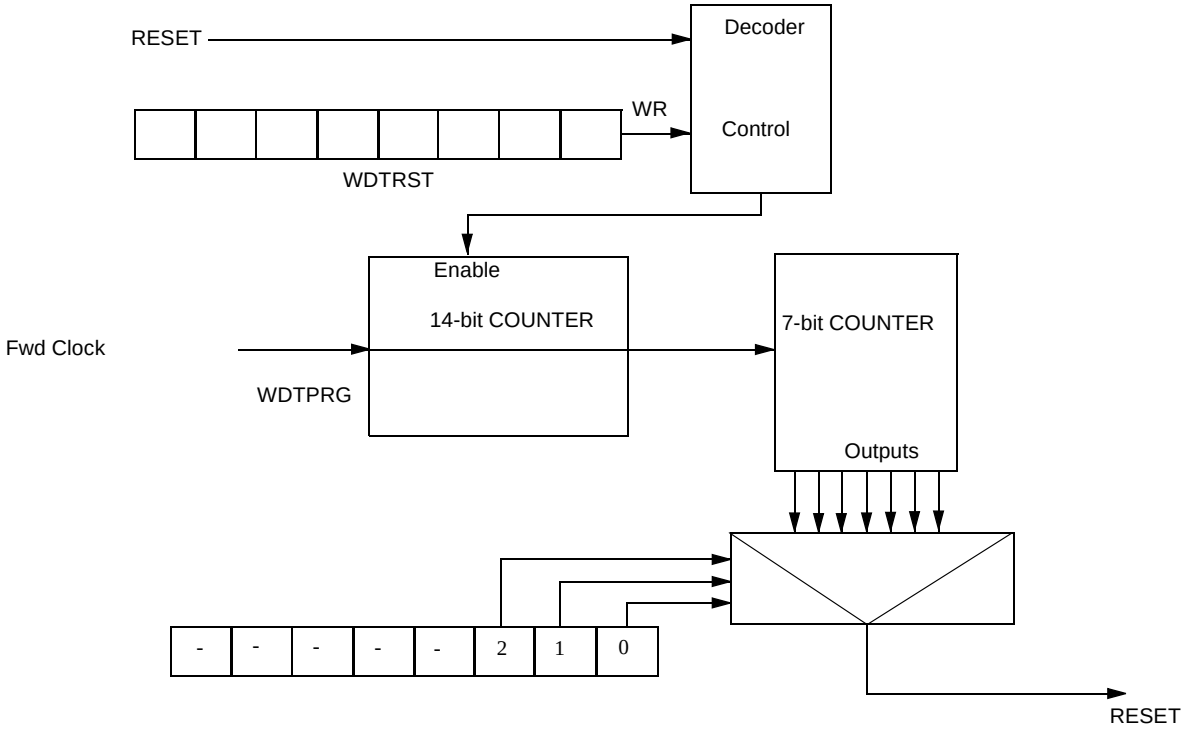
AT89C51CC03 contains a powerful programmable hardware Watchdog Timer (WDT) that automatically resets the chip if it software fails to reset the WDT before the selected time interval has elapsed. It permits large Time-Out ranking from 16ms to 2s @Fosc = 12MHz in X1 mode.

This WDT consists of a 14-bit counter plus a 7-bit programmable counter, a Watchdog Timer reset register (WDTRST) and a Watchdog Timer programming (WDTPRG) register. When exiting reset, the WDT is -by default- disable.

To enable the WDT, the user has to write the sequence 1EH and E1H into WDTRST register no instruction in between. When the Watchdog Timer is enabled, it will increment every machine cycle while the oscillator is running and there is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is $96xT_{OSC}$, where $T_{OSC}=1/F_{OSC}$. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset

Note: When the Watchdog is enable it is impossible to change its period.

Figure 42. Watchdog Timer



Bit Timing and Baud Rate

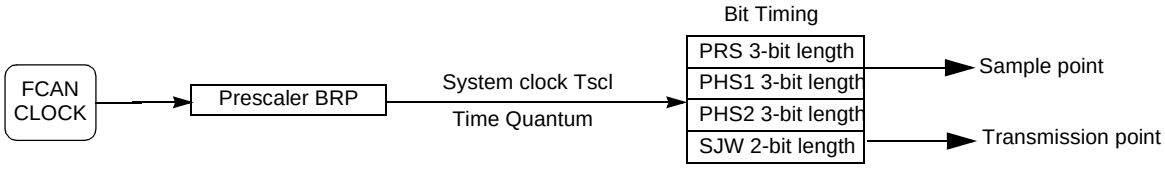
FSM's (Finite State Machine) of the CAN channel need to be synchronous to the time quantum. So, the input clock for bit timing is the clock used into CAN channel FSM's.

Field and segment abbreviations:

- BRP: Baud Rate Prescaler.
- TQ: Time Quantum (output of Baud Rate Prescaler).
- SYNS: SYNchronization Segment is 1 TQ long.
- PRS: PRopagation time Segment is programmable to be 1, 2, ..., 8 TQ long.
- PHS1: PHase Segment 1 is programmable to be 1, 2, ..., 8 TQ long.
- PHS2: PHase Segment 2 is programmable to be superior or equal to the INFORMATION PROCESSING TIME and inferior or equal to TPSH1.
- INFORMATION PROCESSING TIME is 2 TQ.
- SJW: (Re) Synchronization Jump Width is programmable to be minimum of PHS1 and 4.

The total number of TQ in a bit time has to be programmed at least from 8 to 25.

Figure 51. Sample And Transmission Point



The baud rate selection is made by Tbit calculation:

Tbit = Tsyns + Tprs + Tphs1 + Tphs2

1. Tsyns = Tscl = (BRP[5..0]+ 1)/Fcan = 1TQ.
2. Tprs = (1 to 8) * Tscl = (PRS[2..0]+ 1) * Tscl
3. Tphs1 = (1 to 8) * Tscl = (PHS1[2..0]+ 1) * Tscl
4. Tphs2 = (1 to 8) * Tscl = (PHS2[2..0]+ 1) * Tscl
   Tphs2 = Max of (Tphs1 and 2TQ)
5. Tsjw = (1 to 4) * Tscl = (SJW[1..0]+ 1) * Tscl

The total number of Tscl (Time Quanta) in a bit time must be comprised between **8 to 25.**

```
// Enable the CAN macro
 CANGCON = 02h
```

2. Configure message object 3 in reception to receive only standard (11-bit identifier) message 100h

```
// Select the message object 3
 CANPAGE = 30h
// Enable the interrupt on this message object
 CANIE2 = 08h
// Clear the status and control register
 CANSTCH = 00h
 CANCONCH = 00h
// Init the acceptance filter to accept only message 100h in standard mode
 CANIDT1 = 20h
 CANIDT2 = 00h
 CANIDT3 = 00h
 CANIDT4 = 00h
 CANIDM1 = FFh
 CANIDM2 = FFh
 CANIDM3 = FFh
 CANIDM4 = FFh
// Enable channel in reception
 CANCONCH = 88h // enable reception
```

Note: To enable the CAN interrupt in reception:

```
EA = 1
ECAN = 1
CANGIE = 20h
```

3. Send a message on the message object 12

```
// Select the message object 12
 CANPAGE = C0h
// Enable the interrupt on this message object
 CANIE1 = 01h
// Clear the Status register
 CANSTCH = 00h;
// load the identifier to send (ex: 555h)
 CANIDT1 = AAh;
 CANIDT2 = A0h;
// load data to send
 CANMSG = 00h
 CANMSG = 01h
 CANMSG = 02h
 CANMSG = 03h
 CANMSG = 04h
 CANMSG = 05h
 CANMSG = 06h
 CANMSG = 07h
// configure the control register
 CANCONCH = 18h
```

Bit Number	Bit Mnemonic	Description
3	CPOL	**Clock Polarity** Cleared to have the SCK set to '0' in idle state. Set to have the SCK set to '1' in idle state.
2	CPHA	**Clock Phase** Cleared to have the data sampled when the SCK leaves the idle state (see CPOL). Set to have the data sampled when the SCK returns to idle state (see CPOL).
1	SPR1	**SPR2 SPR1 SPR0 Serial Peripheral Rate** 0 0 0 Invalid 0 0 1 $F_{CLK\ PERIPH}$/4 0 1 0 $F_{CLK\ PERIPH}$/8 0 1 1 $F_{CLK\ PERIPH}$/16 1 0 0 $F_{CLK\ PERIPH}$/32 1 0 1 $F_{CLK\ PERIPH}$/64 1 1 0 $F_{CLK\ PERIPH}$/128 1 1 1 Invalid
0	SPR0	

Reset Value = 0001 0100b

Not bit addressable

Serial Peripheral Status Register and Control (SPSCR)

The Serial Peripheral Status Register contains flags to signal the following conditions:

- Data transfer complete
- Write collision
- Inconsistent logic level on $\overline{SS}$ pin (mode fault error)

Table 93. SPSCR Register

SPSCR - Serial Peripheral Status and Control register (0D5H)

7	6	5	4	3	2	1	0
SPIF	**-**	**OVR**	**MODF**	**SPTE**	**UARTM**	**SPTEIE**	**MODFIE**

Bit Number	Bit Mnemonic	Description
7	SPIF	**Serial Peripheral Data Transfer Flag** Cleared by hardware to indicate data transfer is in progress or has been approved by a clearing sequence. Set by hardware to indicate that the data transfer has been completed. This bit is cleared when reading or writing SPDATA after reading SPSCR.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	OVR	**Overrun Error Flag** - Set by hardware when a byte is received whereas SPIF is set (the previous received data is not overwritten). - Cleared by hardware when reading SPSCR

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

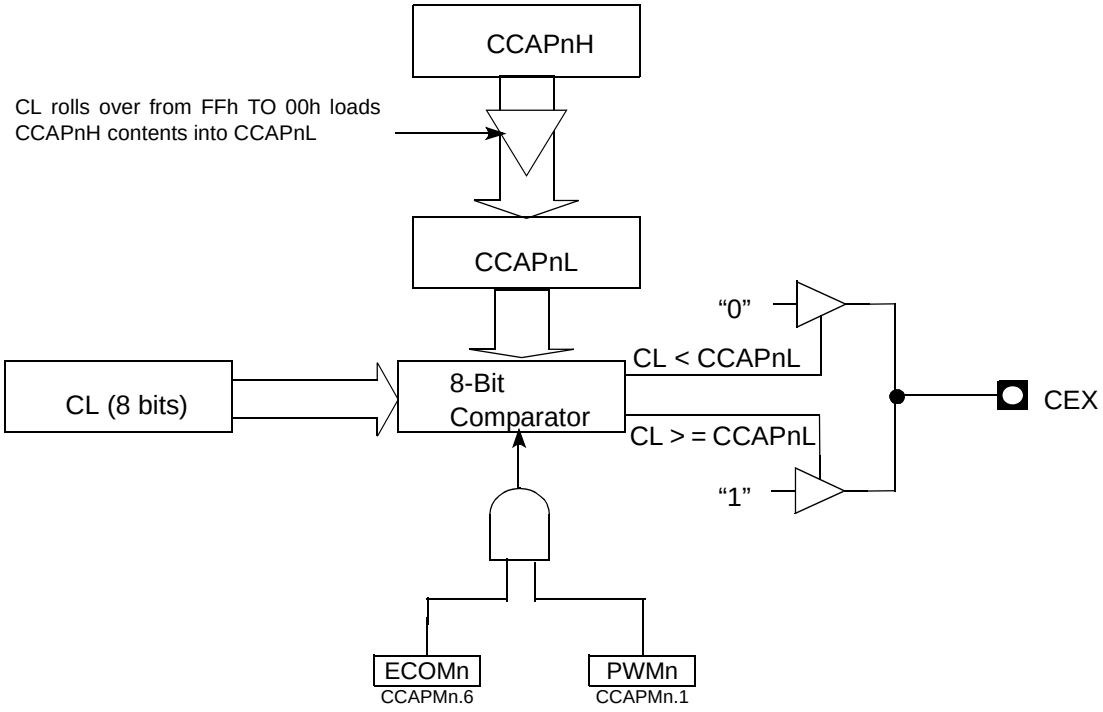
SPDAT - Serial Peripheral Data Register (0D6H)

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

Reset Value = Indeterminate

R7:R0: Receive data bits

Figure 72. PCA PWM Mode



PCA WatchDog Timer

An on-board WatchDog timer is available with the PCA to improve system reliability without increasing chip count. WatchDog timers are useful for systems that are sensitive to noise, power glitches, or electrostatic discharge. Module 4 is the only PCA module that can be programmed as a WatchDog. However, this module can still be used for other modes if the WatchDog is not needed. The user pre-loads a 16-bit value in the compare registers. Just like the other compare modes, this 16-bit value is compared to the PCA timer value. If a match is allowed to occur, an internal reset will be generated. This will not cause the RST pin to be driven high.

To hold off the reset, the user has three options:

- periodically change the compare value so it will never match the PCA timer,
- periodically change the PCA timer value so it will never match the compare values, or
- disable the WatchDog by clearing the WDTE bit before a match occurs and then re-enable it.

The first two options are more reliable because the WatchDog timer is never disabled as in the third option. If the program counter ever goes astray, a match will eventually occur and cause an internal reset. If other PCA modules are being used the second option not recommended either. Remember, the PCA timer is the time base for all modules; changing the time base for other modules would not be a good idea. Thus, in most applications the first solution is the best option.

Table 105. ADCLK Register

ADCLK (S:F2h)
ADC Clock Prescaler

7	6	5	4	3	2	1	0
-	-	-	PRS 4	PRS 3	PRS 2	PRS 1	PRS 0

Bit Number	Bit Mnemonic	Description
7-5	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
4-0	PRS4:0	**Clock Prescaler** See Note (1)

Reset Value = XXX0 0000b

Note: 1. In X1 mode:

For PRS > 0 $F_{ADC} = \frac{FXTAL}{4xPRS}$

For PRS = 0 $F_{ADC} = \frac{FXTAL}{128}$

In X2 mode:

For PRS > 0 $F_{ADC} = \frac{FXTAL}{2xPRS}$

For PRS = 0 $F_{ADC} = \frac{FXTAL}{64}$

Table 106. ADDH Register

ADDH (S:F5h Read Only)
ADC Data High Byte Register

7	6	5	4	3	2	1	0
ADAT 9	ADAT 8	ADAT 7	ADAT 6	ADAT 5	ADAT 4	ADAT 3	ADAT 2

Bit Number	Bit Mnemonic	Description
7-0	ADAT9:2	**ADC result** bits 9-2

Reset Value = 00h

Table 107. ADDL Register

ADDL (S:F4h Read Only)
ADC Data Low Byte Register

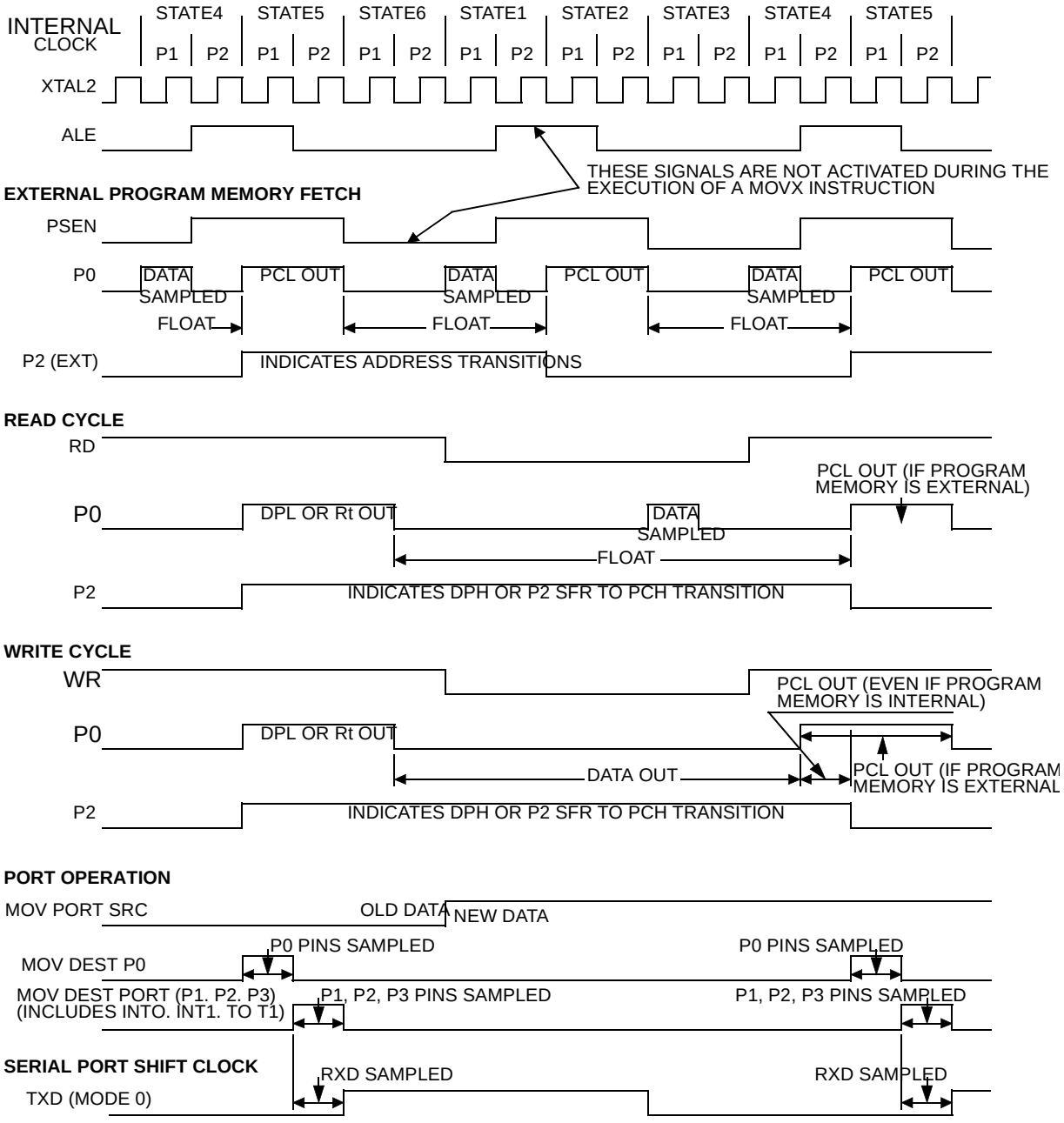
7	6	5	4	3	2	1	0
-	-	-	-	-	-	ADAT 1	ADAT 0

Table 116. DC Parameters in Standard Voltage (Continued)

Symbol	Parameter	Min	Typ[5]	Max	Unit	Test Conditions
V_{OH}	Output High Voltage, ports 1, 2, 3, and 4	V_{CC} - 0.3 V_{CC} - 0.7 V_{CC} - 1.5			V V V	I_{OH} = -10 µA I_{OH} = -30 µA I_{OH} = -60 µA V_{CC} = 3V to 5.5V
V_{OH1}	Output High Voltage, port 0, ALE, $\overline{PSEN}$	V_{CC} - 0.3 V_{CC} - 0.7 V_{CC} - 1.5			V V V	I_{OH} = -200 µA I_{OH} = -3.2 mA I_{OH} = -7.0 mA V_{CC} = 5V ± 10%
R_{RST}	RST Pulldown Resistor	20	100	200	kΩ	
I_{IL}	Logical 0 Input Current ports 1, 2, 3 and 4			-50	µA	Vin = 0.45V
I_{LI}	Input Leakage Current			±10	µA	0.45V < Vin < V_{CC}
I_{TL}	Logical 1 to 0 Transition Current, ports 1, 2, 3 and 4			-650	µA	Vin = 2.0V
C_{IO}	Capacitance of I/O Buffer			10	pF	Fc = 1 MHz T_A = 25°C
I_{PD}	Power-down Current Industrial		75	150	µA	3V < V_{CC} < 5.5V[3]
	Power-down Current Automotive		100	350	µA	3V < V_{CC} < 5.5V[3]
I_{CC}	Power Supply Current	I_{CCOP} = 0.4 Frequency (MHz) + 8 I_{CCIDLE} = 0.2 Frequency (MHz) + 8			mA	Vcc = 5.5V[1][2]
$I_{CCWRITE}$	Power Supply Current on flash or EEdata write			0.8 x Frequency (MHz) + 15	mA	V_{CC} = 5.5V

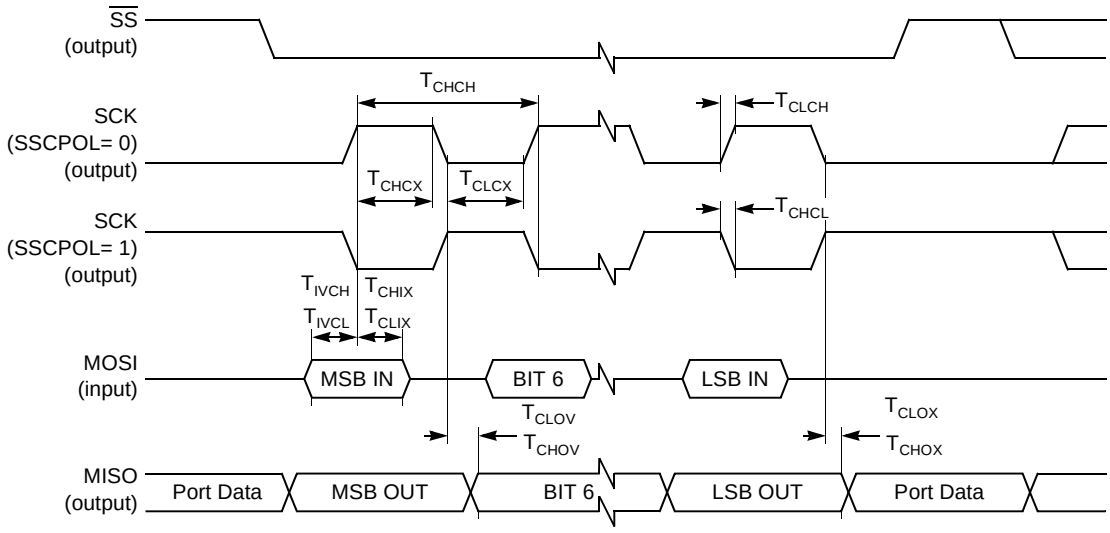
Notes:
1. Operating I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns (see Figure 81.), V_{IL} = V_{SS} + 0.5V,
   V_{IH} = V_{CC} - 0.5V; XTAL2 N.C.; $\overline{EA}$ = RST = Port 0 = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator used (see Figure 78.).
2. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} - 0.5V; XTAL2 N.C; Port 0 = V_{CC}; $\overline{EA}$ = RST = V_{SS} (see Figure 79.).
3. Power-down I_{CC} is measured with all output pins disconnected; $\overline{EA}$ = V_{CC}, PORT 0 = V_{CC}; XTAL2 NC.; RST = V_{SS} (see Figure 80.). In addition, the WDT must be inactive and the POF flag must be set.
4. Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. 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 operation. In the worst cases (capacitive loading 100pF), the noise pulse on the ALE line may exceed 0.45V with maxi V_{OL} peak 0.6V. A Schmitt Trigger use is not necessary.
5. Typicals are based on a limited number of samples and are not guaranteed. The values listed are at room temperature.
6. 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} per 8-bit port:
   Port 0: 26 mA
   Ports 1, 2, 3 and 4: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

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



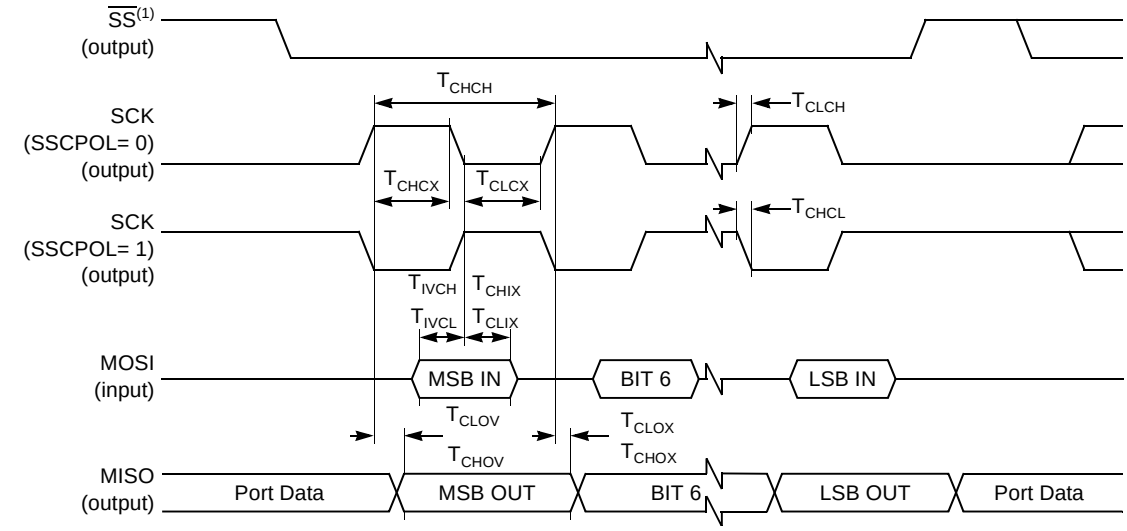
This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically 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.

Figure 86. SPI Master Waveforms (SSCPHA= 0)



Note: 1. $\overline{SS}$ handled by software using general purpose port pin.

Figure 87. SPI Master Waveforms (SSCPHA= 1)



Note: 1. $\overline{SS}$ handled by software using general purpose port pin.

Note:

Changes from 4182N 03/08 to 4182O 09/08

1. Correction to SPDT register address Table 94 on page 139.