



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 2048 Bytes of On-chip ERAM
- 64K Bytes of On-chip Flash Memory
 - Data Retention: 10 Years at 85°C
 - Read/Write Cycle: 100K
- 2K Bytes of On-chip Flash for Bootloader
- 2K Bytes of On-chip EEPROM
 - Read/Write Cycle: 100K
- Integrated Power Monitor (POR: PFD) To Supervise Internal Power Supply
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- High-speed Architecture
 - In Standard Mode:
 - 40 MHz (Vcc 3V to 5.5V, both Internal and external code execution)
 - 60 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
 - In X2 mode (6 Clocks/machine cycle)
 - 20 MHz (Vcc 3V to 5.5V, both Internal and external code execution)
 - 30 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
- Five Ports: 32 + 4 Digital I/O Lines
- Five-channel 16-bit PCA with
 - PWM (8-bit)
 - High-speed Output
 - Timer and Edge Capture
- Double Data Pointer
- 21-bit WatchDog Timer (7 Programmable Bits)
- A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- SPI Interface, (PLCC52 and VFP64 packages only)
- Full CAN Controller
 - Fully Compliant with CAN Rev 2.0A and 2.0B
 - Optimized Structure for Communication Management (Via SFR)
 - 15 Independent Message Objects
 - Each Message Object Programmable on Transmission or Reception
 - Individual Tag and Mask Filters up to 29-bit Identifier/Channel
 - 8-byte Cyclic Data Register (FIFO)/Message Object
 - 16-bit Status and Control Register/Message Object
 - 16-bit Time-Stamping Register/Message Object
 - CAN Specification 2.0 Part A or 2.0 Part B Programmable for Each Message Object
 - Access to Message Object Control and Data Registers Via SFR
 - Programmable Reception Buffer Length Up To 15 Message Objects
 - Priority Management of Reception of Hits on Several Message Objects at the Same Time (Basic CAN Feature)
 - Priority Management for Transmission
 - Message Object Overrun Interrupt
 - Supports
 - Time Triggered Communication
 - Autobaud and Listening Mode
 - Programmable Automatic Reply Mode
 - 1-Mbit/s Maximum Transfer Rate at 8 MHz⁽¹⁾ Crystal Frequency in X2 Mode
 - Readable Error Counters
 - Programmable Link to On-chip Timer for Time Stamping and Network Synchronization
 - Independent Baud Rate Prescaler
 - Data, Remote, Error and Overload Frame Handling

1. At BRP = 1 sampling point will be fixed.



Enhanced 8-bit MCU with CAN Controller and Flash Memory

AT89C51CC03



- On-chip Emulation Logic (Enhanced Hook System)
- Power Saving Modes
 - Idle Mode
 - Power-down Mode
- Power Supply: 3 volts to 5.5 volts
- Temperature Range: Industrial (-40° to +85°C), Automotive (-40°C to +125°C)
- Packages: VQFP44, PLCC44, VQFP64, PLCC52

Description

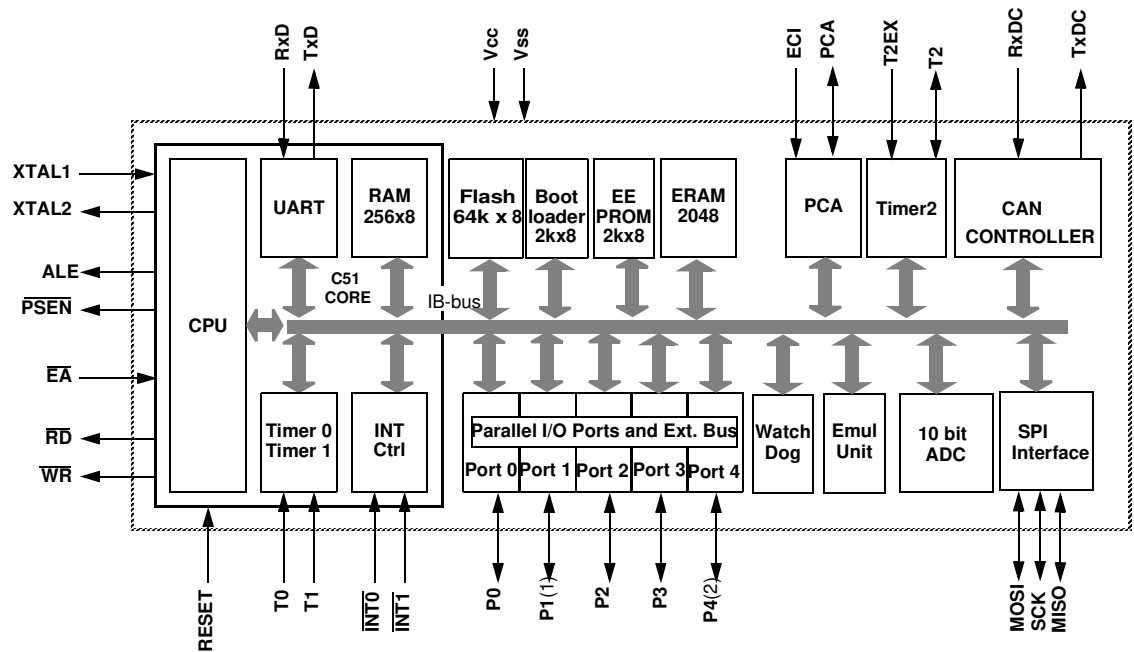
The AT89C51CC03 is a member of the family of 8-bit microcontrollers dedicated to CAN network applications.

In X2 mode a maximum external clock rate of 20 MHz reaches a 300 ns cycle time.

Besides the full CAN controller AT89C51CC03 provides 64K Bytes of Flash memory including In-System Programming (ISP), 2K Bytes Boot Flash Memory, 2K Bytes EEPROM and 2048 byte ERAM.

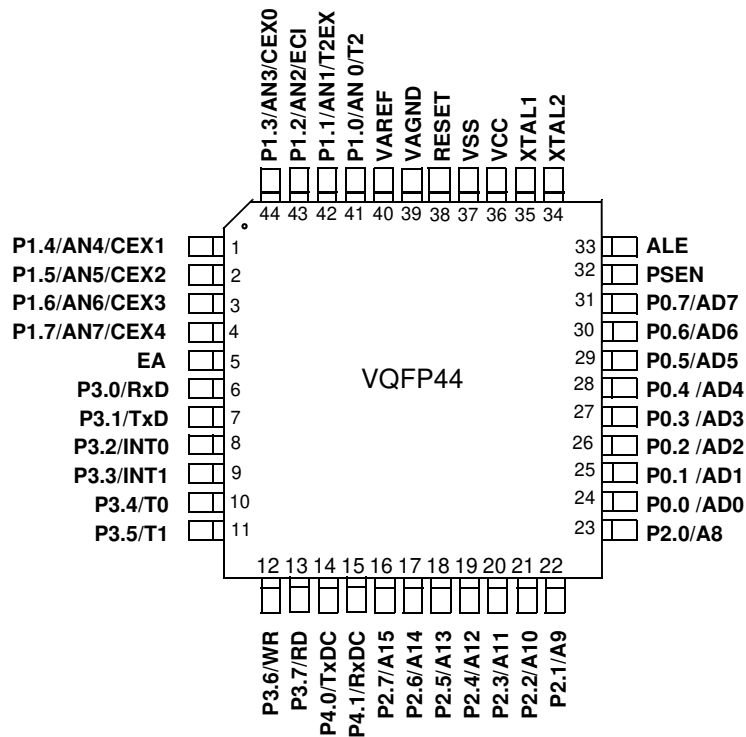
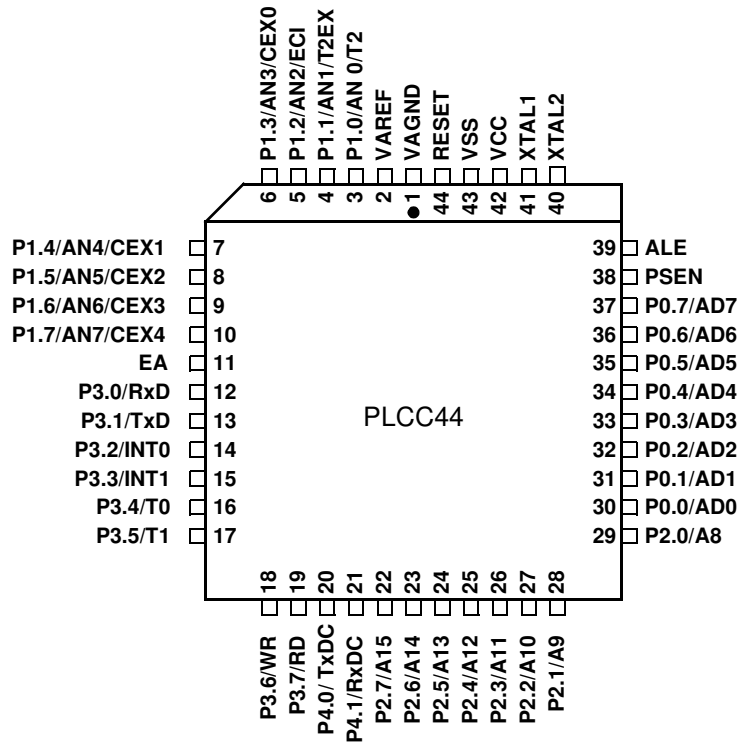
Primary attention is paid to the reduction of the electro-magnetic emission of AT89C51CC03.

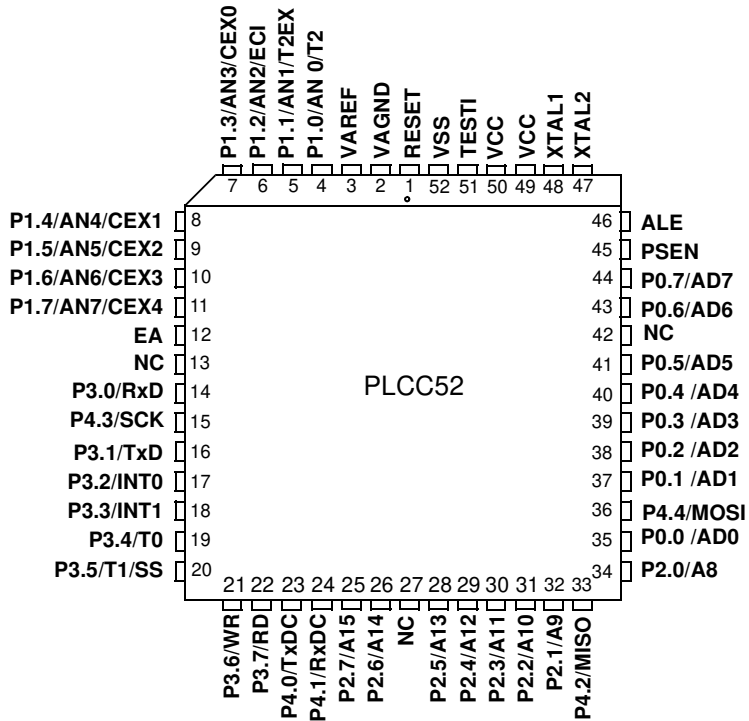
Block Diagram



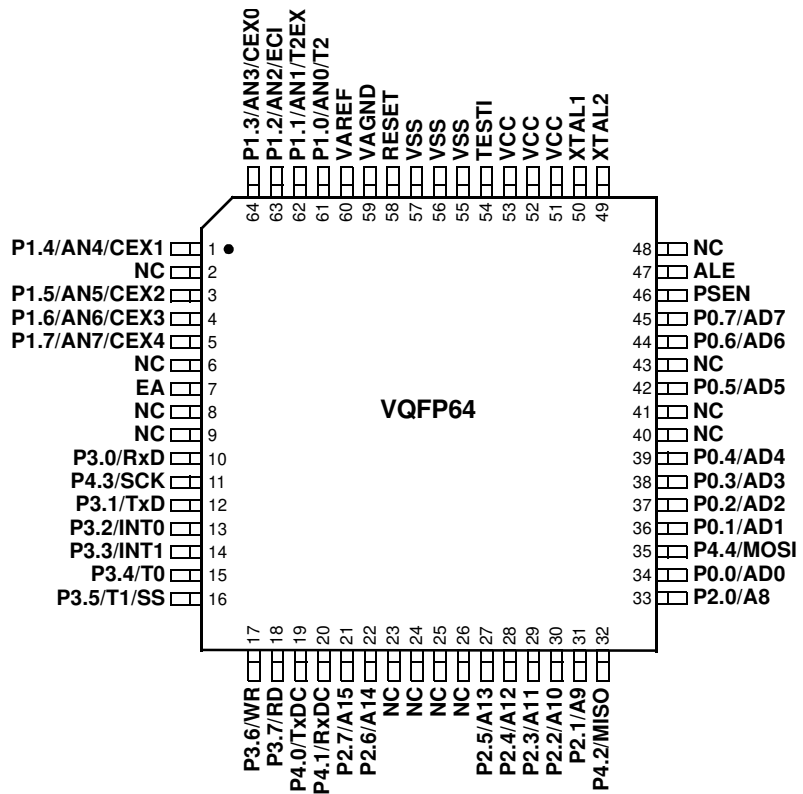
- Notes: 1. 8 analog Inputs/8 Digital I/O
2. 5-Bit I/O Port

Pin Configuration





TESTI must be connected to VSS



TESTI must be connected to VSS

Pin Name	Type	Description
VSS	GND	Circuit ground
TEST1	I	Must be connected to VSS
VCC		Supply Voltage
VAREF		Reference Voltage for ADC
VAGND		Reference Ground for ADC
P0.0:7	I/O	<p>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.</p>
P1.0:7	I/O	<p>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.</p> <p>P1.0/AN0/T2 Analog input channel 0, External clock input for Timer/counter2.</p> <p>P1.1/AN1/T2EX Analog input channel 1, Trigger input for Timer/counter2.</p> <p>P1.2/AN2/ECI Analog input channel 2, PCA external clock input.</p> <p>P1.3/AN3/CEX0 Analog input channel 3, PCA module 0 Entry of input/PWM output.</p> <p>P1.4/AN4/CEX1 Analog input channel 4, PCA module 1 Entry of input/PWM output.</p> <p>P1.5/AN5/CEX2 Analog input channel 5, PCA module 2 Entry of input/PWM output.</p> <p>P1.6/AN6/CEX3 Analog input channel 6, PCA module 3 Entry of input/PWM output.</p> <p>P1.7/AN7/CEX4 Analog input channel 7, PCA module 4 Entry of input/PWM output.</p> <p>Port 1 receives the low-order address byte during EPROM programming and program verification. It can drive CMOS inputs without external pull-ups.</p>
P2.0:7	I/O	<p>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.</p>

Pin Name	Type	Description
P3.0:7	I/O	<p>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 WR). The secondary functions are assigned to the pins of port 3 as follows:</p> <p>P3.0/RxD: Receiver data input (asynchronous) or data input/output (synchronous) of the serial interface</p> <p>P3.1/TxD: Transmitter data output (asynchronous) or clock output (synchronous) of the serial interface</p> <p>P3.2/INT0: External interrupt 0 input/timer 0 gate control input</p> <p>P3.3/INT1: External interrupt 1 input/timer 1 gate control input</p> <p>P3.4/T0: Timer 0 counter input</p> <p>P3.5/T1/SS: Timer 1 counter input SPI Slave Select</p> <p>P3.6/WR: External Data Memory write strobe; latches the data byte from port 0 into the external data memory</p> <p>P3.7/RD: External Data Memory read strobe; Enables the external data memory. It can drive CMOS inputs without external pull-ups.</p>
P4.0:4	I/O	<p>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 (I_{IL}, 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:</p> <p>P4.0/TxDC: Transmitter output of CAN controller</p> <p>P4.1/RxDC: Receiver input of CAN controller.</p> <p>P4.2/MISO: Master Input Slave Output of SPI controller</p> <p>P4.3/SCK: Serial Clock of SPI controller</p> <p>P4.4/MOSI: Master Output Slave Input of SPI controller</p> <p>It can drive CMOS inputs without external pull-ups.</p>

Pin Name	Type	Description
RESET	I/O	Reset: A high level on this pin during two machine cycles while the oscillator is running resets the device. An internal pull-down resistor to VSS permits power-on reset using only an external capacitor to VCC.
ALE	O	ALE: An Address Latch Enable output for latching the low byte of the address during accesses to the external memory. The ALE is activated every 1/6 oscillator periods (1/3 in X2 mode) except during an external data memory access. When instructions are executed from an internal Flash ($\overline{EA} = 1$), ALE generation can be disabled by the software.
PSEN	O	PSEN: The Program Store Enable output is a control signal that enables the external program memory of the bus during external fetch operations. It is activated twice each machine cycle during fetches from the external program memory. However, when executing from the external program memory two activations of PSEN are skipped during each access to the external Data memory. The PSEN is not activated for internal fetches.
EA	I	\overline{EA}: When External Access is held at the high level, instructions are fetched from the internal Flash. When held at the low level, AT89C51CC03 fetches all instructions from the external program memory.
XTAL1	I	XTAL1: Input of the inverting oscillator amplifier and input of the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. To operate above a frequency of 16 MHz, a duty cycle of 50% should be maintained.
XTAL2	O	XTAL2: Output from the inverting oscillator amplifier.

I/O Configurations

Each Port SFR operates via type-D latches, as illustrated in Figure 1 for Ports 3 and 4. A CPU "write to latch" signal initiates transfer of internal bus data into the type-D latch. A CPU "read latch" signal transfers the latched Q output onto the internal bus. Similarly, a "read pin" signal transfers the logical level of the Port pin. Some Port data instructions activate the "read latch" signal while others activate the "read pin" signal. Latch instructions are referred to as Read-Modify-Write instructions. Each I/O line may be independently programmed as input or output.

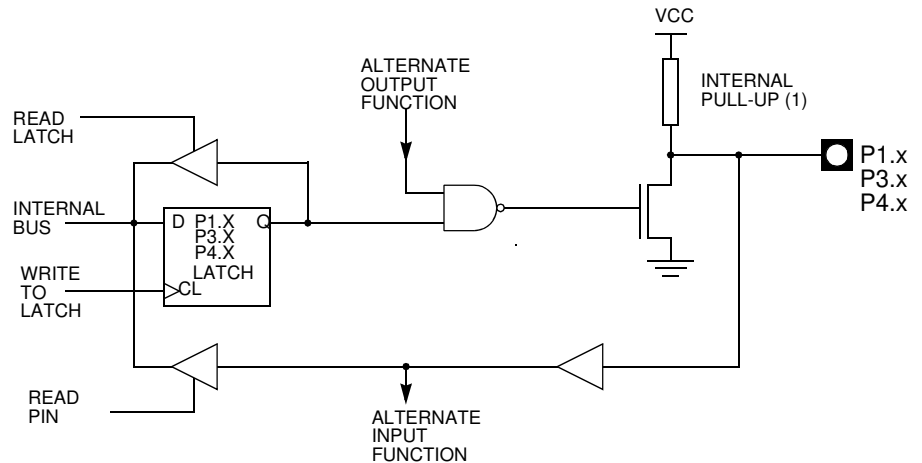
Port 1, Port 3 and Port 4

Figure 1 shows the structure of Ports 1 and 3, which have internal pull-ups. An external source can pull the pin low. Each Port pin can be configured either for general-purpose I/O or for its alternate input output function.

To use a pin for general-purpose output, set or clear the corresponding bit in the Px register (x = 1,3 or 4). To use a pin for general-purpose input, set the bit in the Px register. This turns off the output FET drive.

To configure a pin for its alternate function, set the bit in the Px register. When the latch is set, the "alternate output function" signal controls the output level (see Figure 1). The operation of Ports 1, 3 and 4 is discussed further in the "quasi-Bidirectional Port Operation" section.

Figure 1. Port 1, Port 3 and Port 4 Structure



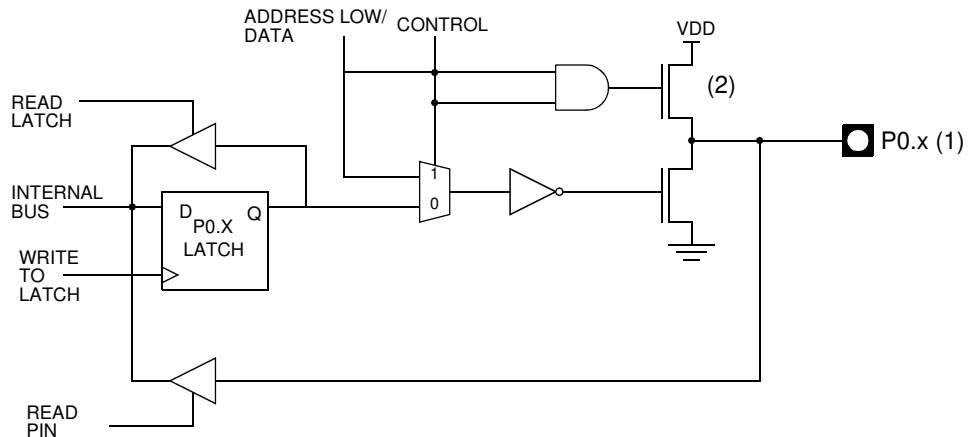
Note: The internal pull-up can be disabled on P1 when analog function is selected.

Port 0 and Port 2

Ports 0 and 2 are used for general-purpose I/O or as the external address/data bus. Port 0, shown in Figure 3, differs from the other Ports in not having internal pull-ups. Figure 3 shows the structure of Port 2. An external source can pull a Port 2 pin low.

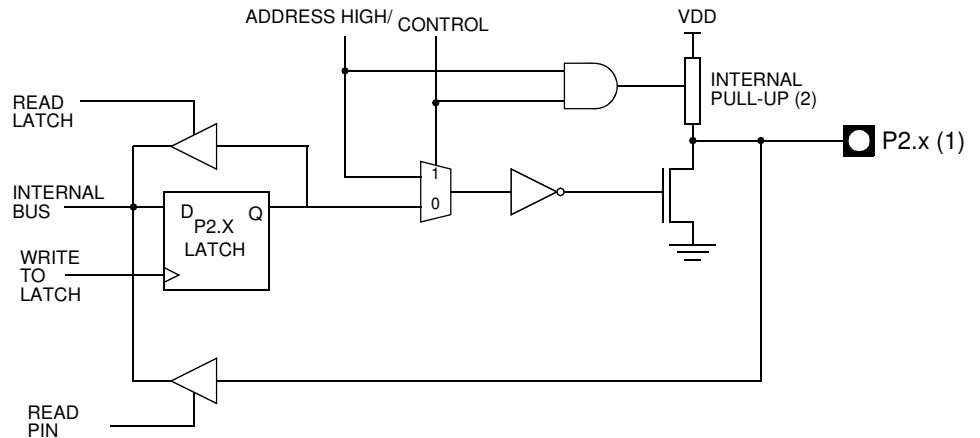
To use a pin for general-purpose output, set or clear the corresponding bit in the Px register (x = 0 or 2). To use a pin for general-purpose input, set the bit in the Px register to turn off the output driver FET.

Figure 2. Port 0 Structure



- Notes:
1. Port 0 is precluded from use as general-purpose I/O Ports when used as address/data bus drivers.
 2. Port 0 internal strong pull-ups assist the logic-one output for memory bus cycles only. Except for these bus cycles, the pull-up FET is off, Port 0 outputs are open-drain.

Figure 3. Port 2 Structure



- Notes:
1. Port 2 is precluded from use as general-purpose I/O Ports when as address/data bus drivers.
 2. Port 2 internal strong pull-ups FET (P1 in FiGURE) assist the logic-one output for memory bus cycle.

When Port 0 and Port 2 are used for an external memory cycle, an internal control signal switches the output-driver input from the latch output to the internal address/data line.

Read-Modify-Write Instructions

Some instructions read the latch data rather than the pin data. The latch based instructions read the data, modify the data and then rewrite the latch. These are called "Read-Modify-Write" instructions. Below is a complete list of these special instructions (see Table). When the destination operand is a Port or a Port bit, these instructions read the latch rather than the pin:

Instruction	Description	Example
ANL	logical AND	ANL P1, A
ORL	logical OR	ORL P2, A
XRL	logical EX-OR	XRL P3, A
JBC	jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	complement bit	CPL P3.0
INC	increment	INC P2
DEC	decrement	DEC P2
DJNZ	decrement and jump if not zero	DJNZ P3, LABEL
MOV Px.y, C	move carry bit to bit y of Port x	MOV P1.5, C
CLR Px.y	clear bit y of Port x	CLR P2.4
SET Px.y	set bit y of Port x	SET P3.3

It is not obvious the last three instructions in this list are Read-Modify-Write instructions. These instructions read the port (all 8 bits), modify the specifically addressed bit and

write the new byte back to the latch. These Read-Modify-Write instructions are directed to the latch rather than the pin in order to avoid possible misinterpretation of voltage (and therefore, logic) levels at the pin. For example, a Port bit used to drive the base of an external bipolar transistor can not rise above the transistor's base-emitter junction voltage (a value lower than VIL). With a logic one written to the bit, attempts by the CPU to read the Port at the pin are misinterpreted as logic zero. A read of the latch rather than the pins returns the correct logic-one value.

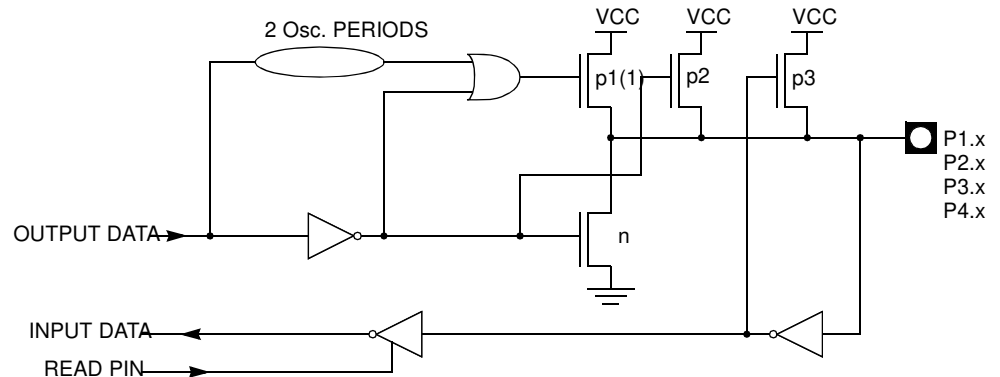
Quasi-Bidirectional Port Operation

Port 1, Port 2, Port 3 and Port 4 have fixed internal pull-ups and are referred to as "quasi-bidirectional" Ports. When configured as an input, the pin impedance appears as logic one and sources current in response to an external logic zero condition. Port 0 is a "true bidirectional" pin. The pins float when configured as input. Resets write logic one to all Port latches. If logical zero is subsequently written to a Port latch, it can be returned to input conditions by a logical one written to the latch.

Note: Port latch values change near the end of Read-Modify-Write instruction cycles. Output buffers (and therefore the pin state) update early in the instruction after Read-Modify-Write instruction cycle.

Logical zero-to-one transitions in Port 1, Port 2, Port 3 and Port 4 use an additional pull-up (p1) to aid this logic transition (see Figure 4.). This increases switch speed. This extra pull-up sources 100 times normal internal circuit current during 2 oscillator clock periods. The internal pull-ups are field-effect transistors rather than linear resistors. Pull-ups consist of three p-channel FET (pFET) devices. A pFET is on when the gate senses logical zero and off when the gate senses logical one. pFET #1 is turned on for two oscillator periods immediately after a zero-to-one transition in the Port latch. A logical one at the Port pin turns on pFET #3 (a weak pull-up) through the inverter. This inverter and pFET pair form a latch to drive logical one. pFET #2 is a very weak pull-up switched on whenever the associated nFET is switched off. This is traditional CMOS switch convention. Current strengths are 1/10 that of pFET #3.

Figure 4. Internal Pull-Up Configurations



Note: Port 2 p1 assists the logic-one output for memory bus cycles.

SFR Mapping

The Special Function Registers (SFRs) of the AT89C51CC03 fall into the following categories:

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
ACC	E0h	Accumulator	-	-	-	-	-	-	-	-
B	F0h	B Register	-	-	-	-	-	-	-	-
PSW	D0h	Program Status Word	CY	AC	F0	RS1	RS0	OV	F1	P
SP	81h	Stack Pointer	-	-	-	-	-	-	-	-
DPL	82h	Data Pointer Low byte LSB of DPTR	-	-	-	-	-	-	-	-
DPH	83h	Data Pointer High byte MSB of DPTR	-	-	-	-	-	-	-	-

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P0	80h	Port 0	-	-	-	-	-	-	-	-
P1	90h	Port 1	-	-	-	-	-	-	-	-
P2	A0h	Port 2	-	-	-	-	-	-	-	-
P3	B0h	Port 3	-	-	-	-	-	-	-	-
P4	C0h	Port 4 (x5)	-	-	-	P4.4 / MOSI	P4.3 / SCK	P4.2 / MISO	P4.1 / RxD	P4.0 / TxD

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
TH0	8Ch	Timer/Counter 0 High byte	-	-	-	-	-	-	-	-
TL0	8Ah	Timer/Counter 0 Low byte	-	-	-	-	-	-	-	-
TH1	8Dh	Timer/Counter 1 High byte	-	-	-	-	-	-	-	-
TL1	8Bh	Timer/Counter 1 Low byte	-	-	-	-	-	-	-	-
TH2	CDh	Timer/Counter 2 High byte	-	-	-	-	-	-	-	-
TL2	CCh	Timer/Counter 2 Low byte	-	-	-	-	-	-	-	-
TCON	88h	Timer/Counter 0 and 1 control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89h	Timer/Counter 0 and 1 Modes	GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
T2CON	C8h	Timer/Counter 2 control	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#
T2MOD	C9h	Timer/Counter 2 Mode	–	–	–	–	–	–	T2OE	DCEN
RCAP2H	CBh	Timer/Counter 2 Reload/Capture High byte	–	–	–	–	–	–	–	–
RCAP2L	CAh	Timer/Counter 2 Reload/Capture Low byte	–	–	–	–	–	–	–	–
WDTRST	A6h	WatchDog Timer Reset	–	–	–	–	–	–	–	–
WDTPRG	A7h	WatchDog Timer Program	–	–	–	–	–	S2	S1	S0

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SCON	98h	Serial Control	FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF	99h	Serial Data Buffer	–	–	–	–	–	–	–	–
SADEN	B9h	Slave Address Mask	–	–	–	–	–	–	–	–
SADDR	A9h	Slave Address	–	–	–	–	–	–	–	–

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CCON	D8h	PCA Timer/Counter Control	CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0
CMOD	D9h	PCA Timer/Counter Mode	CIDL	WDTE	–	–	–	CPS1	CPS0	ECF
CL	E9h	PCA Timer/Counter Low byte	–	–	–	–	–	–	–	–
CH	F9h	PCA Timer/Counter High byte	–	–	–	–	–	–	–	–
CCAPM0	DAh	PCA Timer/Counter Mode 0	–	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0
CCAPM1	DBh	PCA Timer/Counter Mode 1	–	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1
CCAPM2	DCh	PCA Timer/Counter Mode 2	–	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2
CCAPM3	DDh	PCA Timer/Counter Mode 3	–	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3
CCAPM4	DEh	PCA Timer/Counter Mode 4	–	ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4
CCAP0H	FAh	PCA Compare Capture Module 0 H	CCAP0H7	CCAP0H6	CCAP0H5	CCAP0H4	CCAP0H3	CCAP0H2	CCAP0H1	CCAP0H0
CCAP1H	FBh	PCA Compare Capture Module 1 H	CCAP1H7	CCAP1H6	CCAP1H5	CCAP1H4	CCAP1H3	CCAP1H2	CCAP1H1	CCAP1H0
CCAP2H	FCh	PCA Compare Capture Module 2 H	CCAP2H7	CCAP2H6	CCAP2H5	CCAP2H4	CCAP2H3	CCAP2H2	CCAP2H1	CCAP2H0
CCAP3H	FDh	PCA Compare Capture Module 3 H	CCAP3H7	CCAP3H6	CCAP3H5	CCAP3H4	CCAP3H3	CCAP3H2	CCAP3H1	CCAP3H0
CCAP4H	FEh	PCA Compare Capture Module 4 H	CCAP4H7	CCAP4H6	CCAP4H5	CCAP4H4	CCAP4H3	CCAP4H2	CCAP4H1	CCAP4H0
CCAP0L	EAh	PCA Compare Capture Module 0 L	CCAP0L7	CCAP0L6	CCAP0L5	CCAP0L4	CCAP0L3	CCAP0L2	CCAP0L1	CCAP0L0
CCAP1L	EBh	PCA Compare Capture Module 1 L	CCAP1L7	CCAP1L6	CCAP1L5	CCAP1L4	CCAP1L3	CCAP1L2	CCAP1L1	CCAP1L0
CCAP2L	ECh	PCA Compare Capture Module 2 L	CCAP2L7	CCAP2L6	CCAP2L5	CCAP2L4	CCAP2L3	CCAP2L2	CCAP2L1	CCAP2L0
CCAP3L	EDh	PCA Compare Capture Module 3 L	CCAP3L7	CCAP3L6	CCAP3L5	CCAP3L4	CCAP3L3	CCAP3L2	CCAP3L1	CCAP3L0
CCAP4L	EEh	PCA Compare Capture Module 4 L	CCAP4L7	CCAP4L6	CCAP4L5	CCAP4L4	CCAP4L3	CCAP4L2	CCAP4L1	CCAP4L0

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
IEN0	A8h	Interrupt Enable Control 0	EA	EC	ET2	ES	ET1	EX1	ET0	EX0
IEN1	E8h	Interrupt Enable Control 1	–	–	–	–	ESPI	ETIM	EADC	ECAN
IPL0	B8h	Interrupt Priority Control Low 0	–	PPC	PT2	PS	PT1	PX1	PT0	PX0
IPH0	B7h	Interrupt Priority Control High 0	–	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
IPL1	F8h	Interrupt Priority Control Low 1	–	–	–	–	SPIL	POVRL	PADCL	PCANL
IPH1	F7h	Interrupt Priority Control High 1	–	–	–	–	SPIH	POVRH	PADCH	PCANH

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
ADCON	F3h	ADC Control	–	PSIDLE	ADEN	ADEOC	ADSST	SCH2	SCH1	SCH0
ADCF	F6h	ADC Configuration	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
ADCLK	F2h	ADC Clock	–	–	–	PRS4	PRS3	PRS2	PRS1	PRS0
ADDH	F5h	ADC Data High byte	ADAT9	ADAT8	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2
ADDL	F4h	ADC Data Low byte	–	–	–	–	–	–	ADAT1	ADAT0

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CANGCON	ABh	CAN General Control	ABRQ	OVRQ	TTC	SYNCTTC	AUT-BAUD	TEST	ENA	GRES
CANGSTA	AAh	CAN General Status	–	OVFG	–	TBSY	RBSY	ENFG	BOFF	ERRP
CANGIT	9Bh	CAN General Interrupt	CANIT	–	OVRTIM	OVRBUF	SERG	CERG	FERG	AERG
CANBT1	B4h	CAN Bit Timing 1	–	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	–
CANBT2	B5h	CAN Bit Timing 2	–	SJW1	SJW0	–	PRS2	PRS1	PRS0	–
CANBT3	B6h	CAN Bit Timing 3	–	PHS22	PHS21	PHS20	PHS12	PHS11	PHS10	SMP
CANEN1	CEh	CAN Enable Channel byte 1	–	ENCH14	ENCH13	ENCH12	ENCH11	ENCH10	ENCH9	ENCH8
CANEN2	CFh	CAN Enable Channel byte 2	ENCH7	ENCH6	ENCH5	ENCH4	ENCH3	ENCH2	ENCH1	ENCH0
CANGIE	C1h	CAN General Interrupt Enable	–	–	ENRX	ENTX	ENERCH	ENBUF	ENERG	–
CANIE1	C2h	CAN Interrupt Enable Channel byte 1	–	IECH14	IECH13	IECH12	IECH11	IECH10	IECH9	IECH8

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CANIE2	C3h	CAN Interrupt Enable Channel byte 2	IECH7	IECH6	IECH5	IECH4	IECH3	IECH2	IECH1	IECH0
CANSIT1	BAh	CAN Status Interrupt Channel byte1	–	SIT14	SIT13	SIT12	SIT11	SIT10	SIT9	SIT8
CANSIT2	BBh	CAN Status Interrupt Channel byte2	SIT7	SIT6	SIT5	SIT4	SIT3	SIT2	SIT1	SIT0
CANTCON	A1h	CAN Timer Control	TPRESC 7	TPRESC 6	TPRESC 5	TPRESC 4	TPRESC 3	TPRESC 2	TPRESC 1	TPRESC 0
CANTIMH	ADh	CAN Timer high	CANTIM 15	CANTIM 14	CANTIM 13	CANTIM 12	CANTIM 11	CANTIM 10	CANTIM 9	CANTIM 8
CANTIML	ACH	CAN Timer low	CANTIM 7	CANTIM 6	CANTIM 5	CANTIM 4	CANTIM 3	CANTIM 2	CANTIM 1	CANTIM 0
CANSTMP H	AFh	CAN Timer Stamp high	TIMSTMP 15	TIMSTMP 14	TIMSTMP 13	TIMSTMP 12	TIMSTMP 11	TIMSTMP 10	TIMSTMP 9	TIMSTMP 8
CANSTMP L	A Eh	CAN Timer Stamp low	TIMSTMP7	TIMSTMP 6	TIMSTMP 5	TIMSTMP 4	TIMSTMP 3	TIMSTMP 2	TIMSTMP 1	TIMSTMP 0
CANTTCH	A5h	CAN Timer TTC high	TIMTTC 15	TIMTTC 14	TIMTTC 13	TIMTTC 12	TIMTTC 11	TIMTTC 10	TIMTTC 9	TIMTTC 8
CANTTCL	A4h	CAN Timer TTC low	TIMTTC 7	TIMTTC 6	TIMTTC 5	TIMTTC 4	TIMTTC 3	TIMTTC 2	TIMTTC 1	TIMTTC 0
CANTEC	9Ch	CAN Transmit Error Counter	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
CANREC	9Dh	CAN Receive Error Counter	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
CANPAGE	B1h	CAN Page	CHNB3	CHNB2	CHNB1	CHNB0	AINC	INDX2	INDX1	INDX0
CANSTCH	B2h	CAN Status Channel	DLCW	TXOK	RXOK	BERR	SERR	CERR	FERR	AERR
CANCONC H	B3h	CAN Control Channel	CONCH1	CONCH0	RPLV	IDE	DLC3	DLC2	DLC1	DLC0
CANMSG	A3h	CAN Message Data	MSG7	MSG6	MSG5	MSG4	MSG3	MSG2	MSG1	MSG0
CANIDT1	BCh	CAN Identifier Tag byte 1(Part A)	IDT10	IDT9	IDT8	IDT7	IDT6	IDT5	IDT4	IDT3
		CAN Identifier Tag byte 1(PartB)	IDT28	IDT27	IDT26	IDT25	IDT24	IDT23	IDT22	IDT21
CANIDT2	BDh	CAN Identifier Tag byte 2 (PartA)	IDT2	IDT1	IDT0	–	–	–	–	–
		CAN Identifier Tag byte 2 (PartB)	IDT20	IDT19	IDT18	IDT17	IDT16	IDT15	IDT14	IDT13
CANIDT3	BEh	CAN Identifier Tag byte 3(PartA)	–	–	–	–	–	–	–	–
		CAN Identifier Tag byte 3(PartB)	IDT12	IDT11	IDT10	IDT9	IDT8	IDT7	IDT6	IDT5


Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CANIDT4	BFh	CAN Identifier Tag byte 4(PartA)	-	-	-	-	-	RTRTAG	-	RB0TAF
		CAN Identifier Tag byte 4(PartB)	IDT4	IDT3	IDT2	IDT1	IDT0		RB1TAG	
CANIDM1	C4h	CAN Identifier Mask byte 1(PartA)	IDMSK10	IDMSK9	IDMSK8	IDMSK7	IDMSK6	IDMSK5	IDMSK4	IDMSK3
		CAN Identifier Mask byte 1(PartB)	IDMSK28	IDMSK27	IDMSK26	IDMSK25	IDMSK24	IDMSK23	IDMSK22	IDMSK21
CANIDM2	C5h	CAN Identifier Mask byte 2(PartA)	IDMSK2	IDMSK1	IDMSK0	-	-	-	-	-
		CAN Identifier Mask byte 2(PartB)	IDMSK20	IDMSK19	IDMSK18	IDMSK17	IDMSK16	IDMSK15	IDMSK14	IDMSK13
CANIDM3	C6h	CAN Identifier Mask byte 3(PartA)	-	-	-	-	-	-	-	-
		CAN Identifier Mask byte 3(PartB)	IDMSK12	IDMSK11	IDMSK10	IDMSK9	IDMSK8	IDMSK7	IDMSK6	IDMSK5
CANIDM4	C7h	CAN Identifier Mask byte 4(PartA)	-	-	-	-	-	RTRMSK	-	IDEMSK
		CAN Identifier Mask byte 4(PartB)	IDMSK4	IDMSK3	IDMSK2	IDMSK1	IDMSK0			

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SPCON	D4h	SPI Control	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
SPSCR	D5h	SPI Status and Control	SPIF	-	OVR	MODF	SPTE	UARTM	SPTEIE	MOFIE
SPDAT	D6h	SPI Data	-	-	-	-	-	-	-	-

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
PCON	87h	Power Control	SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
AUXR	8Eh	Auxiliary Register 0	DPU	VPFDP	M0	XRS2	XRS1	XRS0	EXTRAM	A0
AUXR1	A2h	Auxiliary Register 1	-	-	ENBOOT	-	GF3	0	-	DPS
CKCON0	8Fh	Clock Control 0	CANX2	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2
CKCON1	9Fh	Clock Control 1	-	-	-	-	-	-	-	SPIX2
FCON	D1h	Flash Control	FPL3	FPL2	FPL1	FPL0	FPS	FMOD1	FMOD0	FBUSY
EECON	D2h	EEPROM Control	EEPL3	EEPL2	EEPL1	EEPL0	-	-	EEE	EEBUSY
FSTA	D3	Flash Status	-	-	-	-	-	-	SEQERR	FLOAD

Table 1. SFR Mapping

	0/8 ⁽²⁾	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	IPL1 xxxx x000	CH 0000 0000	CCAP0H 0000 0000	CCAP1H 0000 0000	CCAP2H 0000 0000	CCAP3H 0000 0000	CCAP4H 0000 0000		FFh
F0h	B 0000 0000		ADCLK xxx0 0000	ADCON x000 0000	ADDL 0000 0000	ADDH 0000 0000	ADCF 0000 0000	IPH1 xxxx x000	F7h
E8h	IEN1 xxxx x000	CL 0000 0000	CCAP0L 0000 0000	CCAP1L 0000 0000	CCAP2L 0000 0000	CCAP3L 0000 0000	CCAP4L 0000 0000		EFh
E0h	ACC 0000 0000								E7h
D8h	CCON 0000 0000	CMOD 00xx x000	CCAPM0 x000 0000	CCAPM1 x000 0000	CCAPM2 x000 0000	CCAPM3 x000 0000	CCAPM4 x000 0000		DFh
D0h	PSW 0000 0000	FCON 0000 0000	EECON xxxx xx00	FSTA xxxx xx00	SPCON 0001 0100	SPSCR 0000 0000	SPDAT xxxx xxxx		D7h
C8h	T2CON 0000 0000	T2MOD xxxx xx00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000	CANEN1 x000 0000	CANEN2 0000 0000	CFh
C0h	P4 xxx1 1111	CANGIE xx00 000x	CANIE1 x000 0000	CANIE2 0000 0000	CANIDM1 xxxx xxxx	CANIDM2 xxxx xxxx	CANIDM3 xxxx xxxx	CANIDM4 xxxx xxxx	C7h
B8h	IPL0 x000 0000	SADEN 0000 0000	CANSIT1 0000 0000	CANSIT2 0000 0000	CANIDT1 xxxx xxxx	CANIDT2 xxxx xxxx	CANIDT3 xxxx xxxx	CANIDT4 xxxx xxxx	BFh
B0h	P3 1111 1111	CANPAGE 0000 0000	CANSTCH xxxx xxxx	CANCONCH xxxx xxxx	CANBT1 xxxx xxxx	CANBT2 xxxx xxxx	CANBT3 xxxx xxxx	IPH0 x000 0000	B7h
A8h	IEN0 0000 0000	SADDR 0000 0000	CANGSTA x0x0 0000	CANGCON 0000 0x00	CANTIML 0000 0000	CANTIMH 0000 0000	CANSTMPL 0000 0000	CANSTMPH 0000 0000	AFh
A0h	P2 1111 1111	CANTCON 0000 0000	AUXR1 xxxx 00x0	CANMSG xxxx xxxx	CANTTCL 0000 0000	CANTTCH 0000 0000	WDTRST 1111 1111	WDTPRG xxxx x000	A7h
98h	SCON 0000 0000	SBUF 0000 0000		CANGIT 0x00 0000	CANTEC 0000 0000	CANREC 0000 0000		CKCON1 xxxx xxx0	9Fh
90h	P1 1111 1111								97h
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR x001 0100	CKCON0 0000 0000	8Fh
80h	P0 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 00x1 0000	87h
	0/8 ⁽²⁾	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

 Reserved 

Note: 1. Do not read or write Reserved Registers

2. These registers are bit-addressable.

Sixteen addresses in the SFR space are both byte-addressable and bit-addressable. The bit-addressable SFR's are those whose address ends in 0 and 8. The bit addresses, in this area, are 0x80 through to 0xFF.

Clock

The AT89C51CC03 core needs only 6 clock periods per machine cycle. This feature, called "X2", provides the following advantages:

- Divides frequency crystals by 2 (cheaper crystals) while keeping the same CPU power.
- Saves power consumption while keeping the same CPU power (oscillator power saving).
- Saves power consumption by dividing dynamic operating frequency by 2 in operating and idle modes.
- Increases CPU power by 2 while keeping the same crystal frequency.

In order to keep the original C51 compatibility, a divider-by-2 is inserted between the XTAL1 signal and the main clock input of the core (phase generator). This divider may be disabled by the software.

An extra feature is available to start after Reset in the X2 mode. This feature can be enabled by a bit X2B in the Hardware Security Byte. This bit is described in the section "In-System Programming".

Description

The X2 bit in the CKCON register (see Table 2) allows switching from 12 clock cycles per instruction to 6 clock cycles and vice versa. At reset, the standard speed is activated (STD mode).

Setting this bit activates the X2 feature (X2 mode) for the CPU Clock only (see Figure 5.).

The Timers 0, 1 and 2, Uart, PCA, WatchDog or CAN switch in X2 mode only if the corresponding bit is cleared in the CKCON register.

The clock for the whole circuit and peripheral is first divided by two before being used by the CPU core and peripherals. This allows any cyclic ratio to be accepted on the XTAL1 input. In X2 mode, as this divider is bypassed, the signals on XTAL1 must have a cyclic ratio between 40 to 60%. Figure 5. shows the clock generation block diagram. The X2 bit is validated on the XTAL1÷2 rising edge to avoid glitches when switching from the X2 to the STD mode. Figure 6 shows the mode switching waveforms.

Figure 5. Clock CPU Generation Diagram

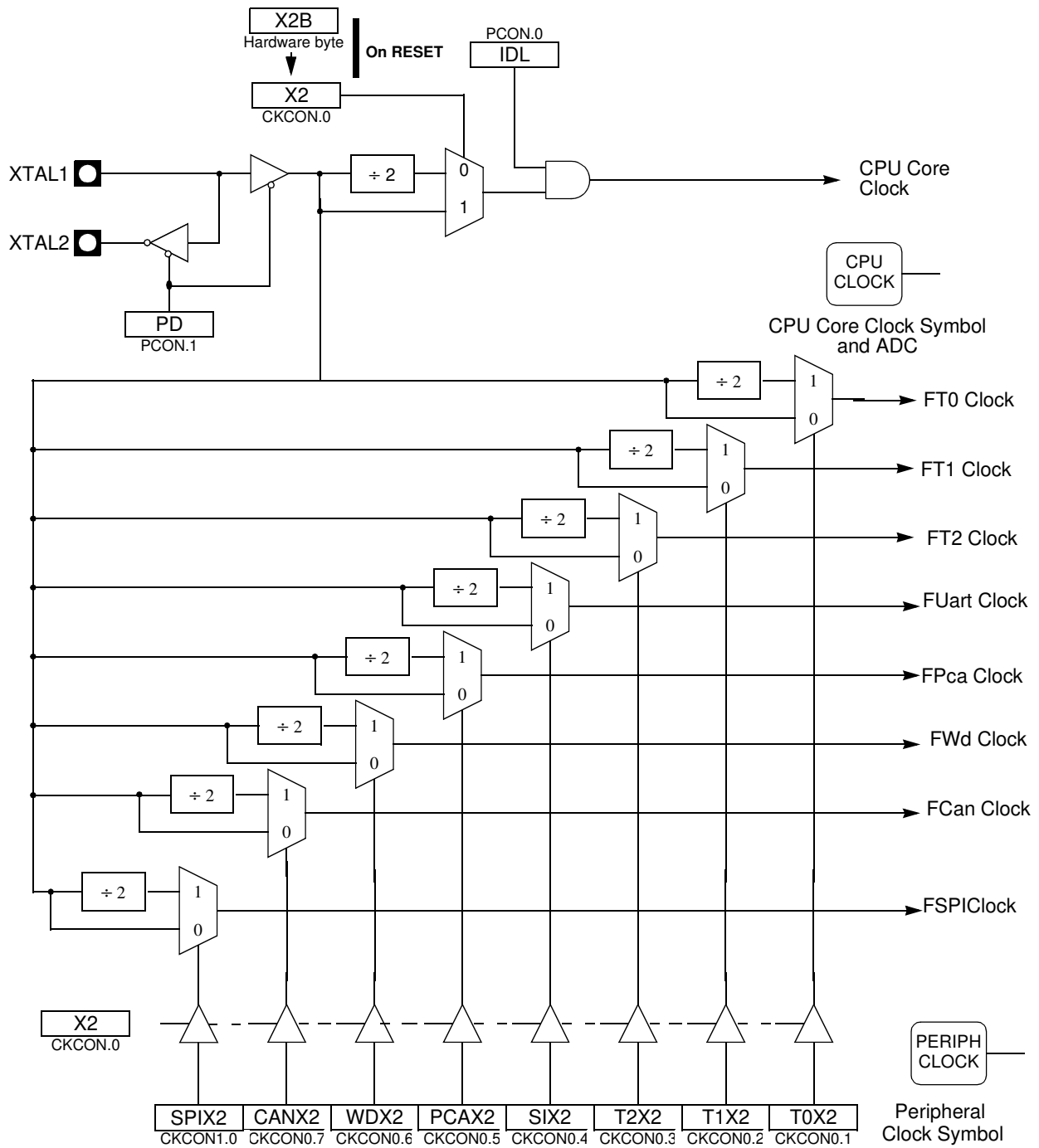
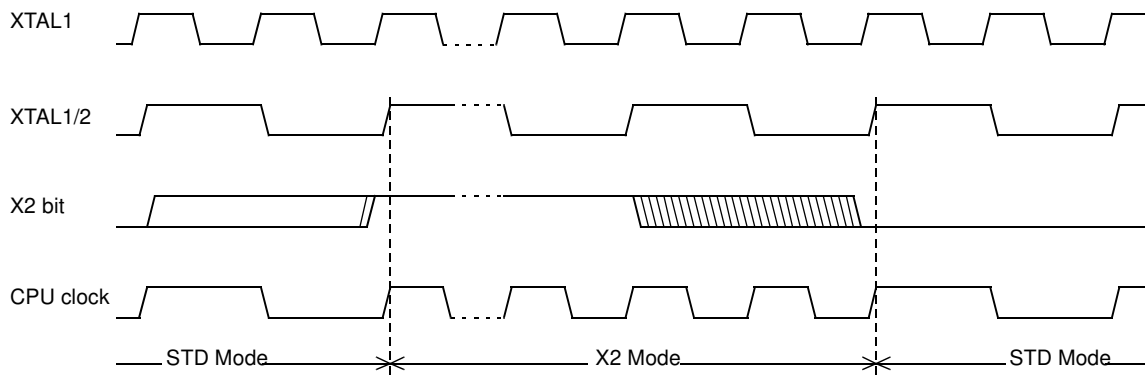


Figure 6. Mode Switching Waveforms



Note: In order to prevent any incorrect operation while operating in the X2 mode, users must be aware that all peripherals using the clock frequency as a time reference (UART, timers...) will have their time reference divided by two. For example a free running timer generating an interrupt every 20 ms will then generate an interrupt every 10 ms. A UART with a 4800 baud rate will have a 9600 baud rate.

Registers

Table 2. CKCON0 Register

CKCON0 (S:8Fh)
Clock Control Register

7	6	5	4	3	2	1	0
CANX2	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2
Bit Number	Bit Mnemonic	Description					
7	CANX2	CAN clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
6	WDX2	WatchDog clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
5	PCAX2	Programmable Counter Array clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
4	SIX2	Enhanced UART clock (MODE 0 and 2) ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
3	T2X2	Timer2 clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
2	T1X2	Timer1 clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
1	T0X2	Timer0 clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					
0	X2	CPU clock Clear to select 12 clock periods per machine cycle (STD mode) for CPU and all the peripherals. Set to select 6 clock periods per machine cycle (X2 mode) and to enable the individual peripherals "X2"bits.					

Note: 1. This control bit is validated when the CPU clock bit X2 is set; when X2 is low, this bit has no effect.

Reset Value = 0000 0000b

Table 3. CKCON1 Register

CKCON1 (S:9Fh)
Clock Control Register 1

7	6	5	4	3	2	1	0
							SPIX2
Bit Number	Bit Mnemonic	Description					
7-1	-	Reserved The value read from these bits is indeterminate. Do not set these bits.					
0	SPIX2	SPI clock ⁽¹⁾ Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.					

Note: 1. This control bit is validated when the CPU clock bit X2 is set; when X2 is low, this bit has no effect.

Reset Value = 0000 0000b

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

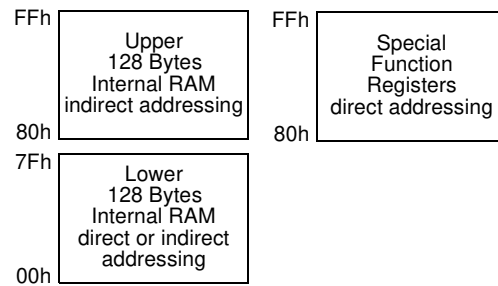
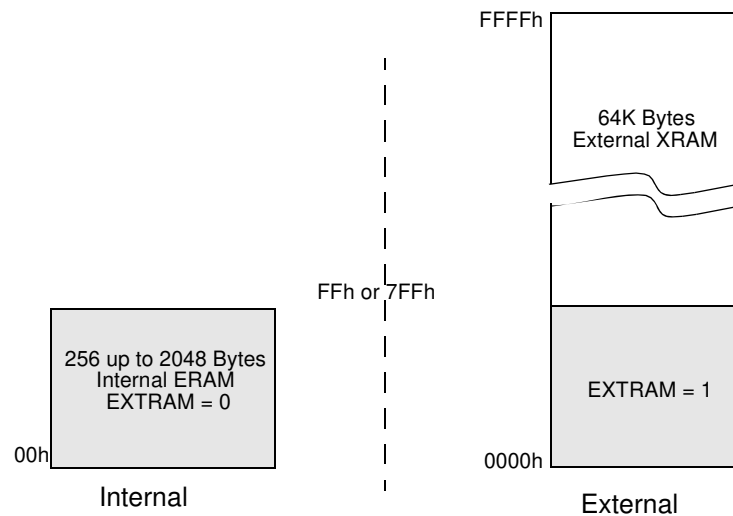


Figure 8. Internal and External Data Memory Organization ERAM-XRAM



Internal Space

Lower 128 Bytes RAM

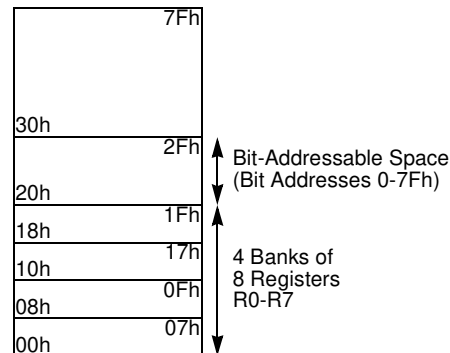
The lower 128 Bytes of RAM (see Figure 8) are accessible from address 00h to 7Fh using direct or indirect addressing modes. The lowest 32 Bytes are grouped into 4 banks of 8 registers (R0 to R7). Two bits RS0 and RS1 in PSW register (see Figure 6) select which bank is in use according to Table 4. This 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.

External Space

Memory Interface

The external memory interface comprises the external bus (port 0 and port 2) as well as the bus control signals (RD#, WR#, and ALE).

Figure 10 shows the structure of the external address bus. P0 carries address A7:0 while P2 carries address A15:8. Data D7:0 is multiplexed with A7:0 on P0. Table 5 describes the external memory interface signals.

Figure 10. External Data Memory Interface Structure

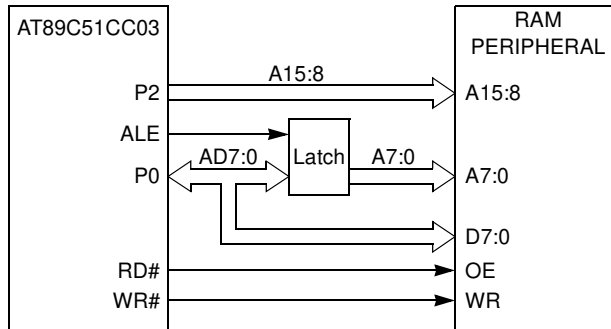


Table 5. External Data Memory Interface Signals

Signal Name	Type	Description	Alternative Function
A15:8	O	Address Lines Upper address lines for the external bus.	P2.7:0
AD7:0	I/O	Address/Data Lines Multiplexed lower address lines and data for the external memory.	P0.7:0
ALE	O	Address Latch Enable ALE signals indicates that valid address information are available on lines AD7:0.	-
RD#	O	Read Read signal output to external data memory.	P3.7
WR#	O	Write Write signal output to external memory.	P3.6

External Bus Cycles

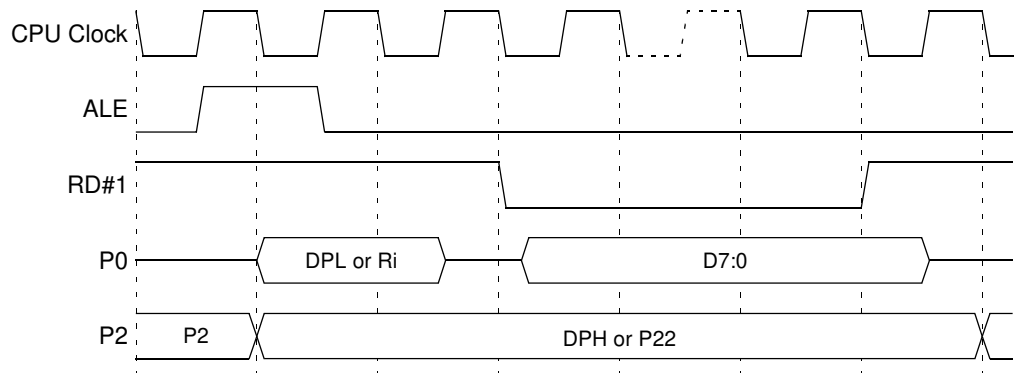
This section describes the bus cycles the AT89C51CC03 executes to read (see Figure 11), and write data (see Figure 12) in the external data memory.

External memory cycle takes 6 CPU clock periods. This is equivalent to 12 oscillator clock period in standard mode or 6 oscillator clock periods in X2 mode. For further information on X2 mode.

Slow peripherals can be accessed by stretching the read and write cycles. This is done using the M0 bit in AUXR register. Setting this bit changes the width of the RD# and WR# signals from 3 to 15 CPU clock periods.

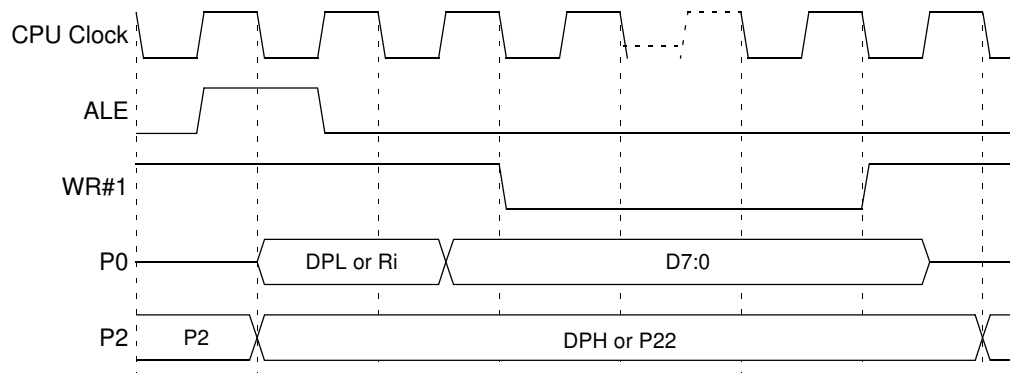
For simplicity, the accompanying figures depict the bus cycle waveforms in idealized form and do not provide precise timing information. For bus cycle timing parameters refer to the Section “AC Characteristics” of the AT89C51CC03 datasheet.

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.