Features

- 80C52 Compatible
  - 8051 Instruction Compatible
  - Four 8-bit I/O Ports (44 Pins Version)
  - Three 16-bit Timer/Counters
  - 256 bytes Scratch Pad RAM
  - 11 Interrupt Sources With 4 Priority Levels
- ISP (In-System Programming) Using Standard V_{CC} Power Supply
- Integrated Power Monitor (POR/PFD) to Supervise Internal Power Supply
- Boot ROM Contains Serial Loader for In-System Programming
- High-speed Architecture
  - In Standard Mode:
    - 40 MHz (Vcc 2.7V 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 2.7V to 5.5V, Both Internal and External Code Execution)
    - 30 MHz (Vcc 4.5V to 5.5V and Internal Code Execution Only)
- 128K bytes On-chip Flash Program/Data Memory
  - 128 bytes Page Write with auto-erase
  - 100k Write Cycles
- On-chip 8192 bytes Expanded RAM (XRAM)
  - Software Selectable Size (0, 256, 512, 768, 1024, 1792, 2048, 4096, 8192 bytes)
- Dual Data Pointer
- Extended stack pointer to 512 bytes
- Variable Length MOVX for Slow RAM/Peripherals
- Improved X2 Mode with Independant Selection for CPU and Each Peripheral
- Keyboard Interrupt Interface on Port 1
- SPI Interface (Master/Slave Mode)
- 8-bit Clock Prescaler
- Programmable Counter Array with:
  - High Speed Output
  - Compare/Capture
  - Pulse Width Modulator
  - Watchdog Timer Capabilities
- Asynchronous Port Reset
- Two Full Duplex Enhanced UART with Dedicated Internal Baud Rate Generator
- Low EMI (inhibit ALE)
- Hardware Watchdog Timer (One-time Enabled with Reset-Out), Power-Off Flag
- Power Control Modes: Idle Mode, Power-down Mode
- Power Supply: 2.7V to 5.5V
- Temperature Ranges: Industrial (-40 to +85°C)
- Packages: PLCC44, VQFP44

8-bit Flash Microcontroller

AT89C51RE2

Description

AT89C51RE2 is a high performance CMOS Flash version of the 80C51 CMOS single chip 8-bit microcontroller. It contains a 128 Kbytes Flash memory block for program.

The 128 Kbytes Flash memory can be programmed either in parallel mode or in serial mode with the ISP capability or with software. The programming voltage is internally generated from the standard V_{CC} pin.

The AT89C51RE2 retains all features of the Atmel 80C52 with 256 bytes of internal RAM, a 10-source 4-level interrupt controller and three timer/counters.

In addition, the AT89C51RE2 has a Programmable Counter Array, an XRAM of 8192 bytes, a Hardware Watchdog Timer, SPI and Keyboard, two serial channels that facilitates multiprocessor communication (EUART), a speed improvement mechanism (X2 mode) and an extended stack mode that allows the stack to be extended in the lower 256 bytes of XRAM.

The fully static design of the AT89C51RE2 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The AT89C51RE2 has 2 software-selectable modes of reduced activity and 8-bit clock prescaler for further reduction in power consumption. In the Idle mode the CPU is frozen while the peripherals and the interrupt system are still operating. In the power-down mode the RAM is saved and all other functions are inoperative.

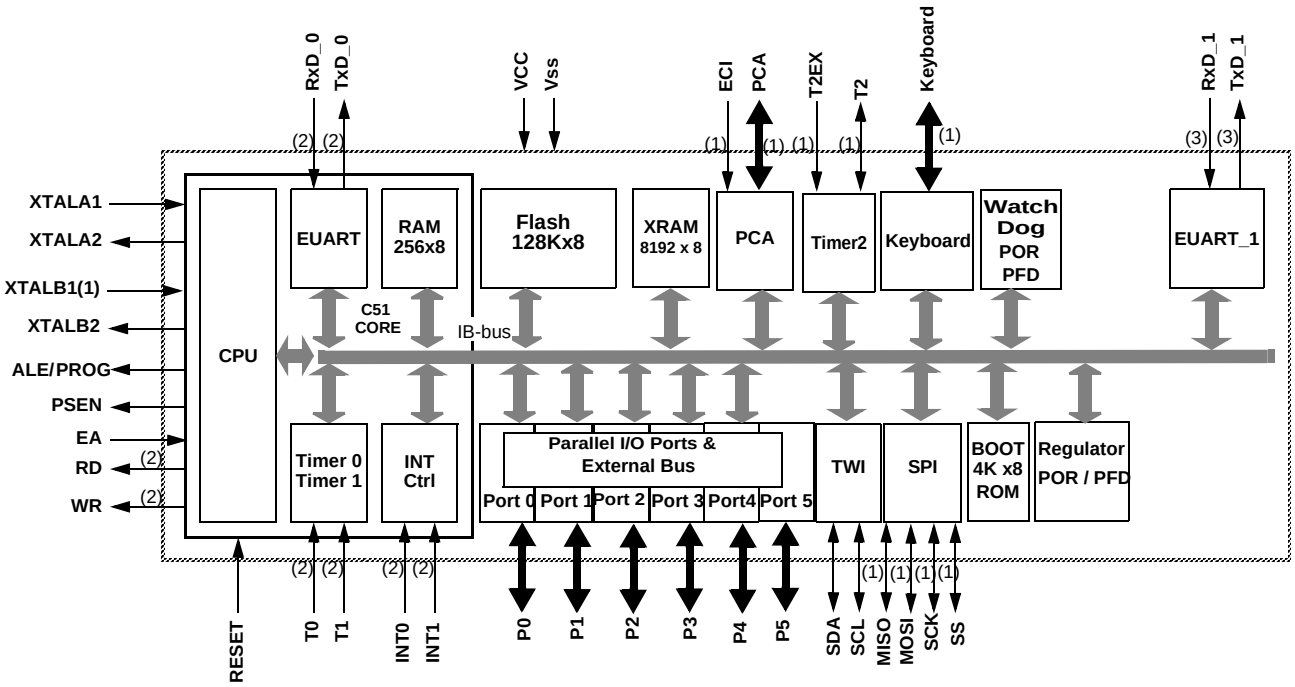
The added features of the AT89C51RE2 make it more powerful for applications that need pulse width modulation, high speed I/O and counting capabilities such as alarms, motor control, corded phones, smart card readers.

Table 1. Memory Size and I/O pins

AT89C51RE2	Flash (bytes)	XRAM (bytes)	TOTAL RAM (bytes)	I/O
PLCC44 **VQFP44**	128K	8192	8192 + 256	34

Block Diagram

Figure 1. Block Diagram

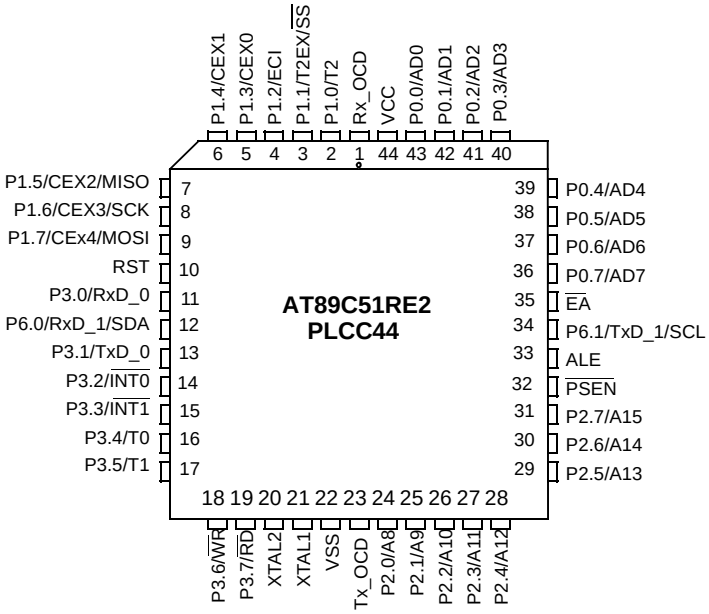


(1): Alternate function of Port 1

(2): Alternate function of Port 3

(3): Alternate function of Port 6

Pin Configurations



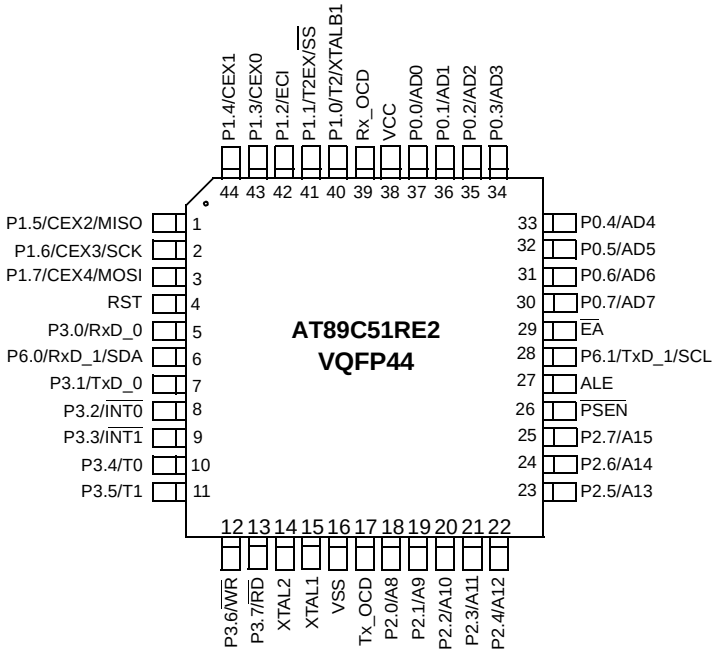


Table 2. Pin Description

Mnemonic	Pin Number LCC	Pin Number VQFP 1.4	Type	Name and Function
V_{SS}	22	16	I	**Ground:** 0V reference
Vss1		39	I	Optional Ground: **Contact the Sales Office for ground connection.**
V_{CC}	44	38	I	**Power Supply:** This is the power supply voltage for normal, idle and power-down operation
P0.0-P0.7	43-36	37-30	I/O	**Port 0**: Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high impedance inputs. Port 0 must be polarized to V_{CC} or V_{SS} in order to prevent any parasitic current consumption. Port 0 is also the multiplexed low-order address and data bus during access to external program and data memory. In this application, it uses strong internal pull-up when emitting 1s. Port 0 also inputs the code bytes during EPROM programming. External pull-ups are required during program verification during which P0 outputs the code bytes.
P1.0-P1.7	2-9	40-44 1-3	I/O	Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current because of the internal pull-ups. Port 1 also receives the low-order address byte during memory programming and verification. Alternate functions for TSC8x54/58 Port 1 include:
	2	40	I/O	**T2 (P1.0):** Timer/Counter 2 external count input/Clockout
	3	41	I	**T2EX (P1.1):** Timer/Counter 2 Reload/Capture/Direction Control
	4	42	I	**ECI (P1.2):** External Clock for the PCA
	5	43	I/O	**CEX0 (P1.3):** Capture/Compare External I/O for PCA module 0
	6	44	I/O	**CEX1 (P1.4):** Capture/Compare External I/O for PCA module 1
	7	1	I/O	**CEX2 (P1.5):** Capture/Compare External I/O for PCA module 2
	8	2	I/O	**CEX3 (P1.6):** Capture/Compare External I/O for PCA module 3
	9	3	I/O	**CEX4 (P1.7):** Capture/Compare External I/O for PCA module 4
P2.0-P2.7	24-31	18-25	I/O	**Port 2**: Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR).In this application, it uses strong internal pull-ups emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 emits the contents of the P2 SFR. Some Port 2 pins receive the high order address bits during EPROM programming and verification: P2.0 to P2.5 for RB devices P2.0 to P2.6 for RC devices P2.0 to P2.7 for RD devices.
P3.0-P3.7	11, 13-19	5, 7-13	I/O	Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally pulled low will source current because of the internal pull-ups. Port 3 also serves the special features of the 80C51 family, as listed below.
	11	5	I	**RXD_0 (P3.0):** Serial input port
	13	7	O	**TXD_0 (P3.1):** Serial output port
	14	8	I	**$\overline{\text{INT0}}$ (P3.2):** External interrupt 0

Mnemonic	Pin Number		Type	Name and Function
	LCC	VQFP 1.4		
	15	9	I	**$\overline{\text{INT1}}$ (P3.3):** External interrupt 1
	16	10	I	**T0 (P3.4):** Timer 0 external input
	17	11	I	**T1 (P3.5):** Timer 1 external input
	18	12	O	**$\overline{\text{WR}}$ (P3.6):** External data memory write strobe
	19	13	O	**$\overline{\text{RD}}$ (P3.7):** External data memory read strobe
P6.0-P6.1	12,34	6, 28		Port 6: Port 6 is an 2-bit bidirectional I/O port with internal pull-ups. Port 6 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 6 pins that are externally pulled low will source current because of the internal pull-ups. Port 6 also serves some special features as listed below.
	12	6	I	**RXD_1 (P6.0):** Serial input port
	12	6	I/O	**SDA (P6.0) : TWI Serial Data** SDA is the bidirectional TWI data line.
	34	28	O	**TXD_1 (P6.1) :** Serial output port
	34	28	I/O	**SCL (P6.1) : TWI Serial Clock** SCL output the serial clock to slave peripherals. SCL input the serial clock from master.
Reset	10	4	I/O	**Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V_{SS} permits a power-on reset using only an external capacitor to V_{CC}. This pin is an output when the hardware watchdog forces a system reset.
ALE/$\overline{\text{PROG}}$	33	27	O (I)	**Address Latch Enable/Program Pulse:** Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 (1/3 in X2 mode) the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during Flash programming. ALE can be disabled by setting SFR's AUXR.0 bit. With this bit set, ALE will be inactive during internal fetches.
PSEN	32	26	O	**Program Store ENable:** The read strobe to external program memory. When executing code from the external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory. $\overline{\text{PSEN}}$ is not activated during fetches from internal program memory.
EA	35	29	I	**External Access Enable:** $\overline{\text{EA}}$ must be externally held low to enable the device to fetch code from external program memory locations 0000H to FFFFH (RD). If security level 1 is programmed, $\overline{\text{EA}}$ will be internally latched on Reset.
XTAL1	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier
Tx_OCD	23	17	O	**Tx_OCD:** On chip debug Serial output port
Rx_OCD	1	39	I	**Rx_OCD:** On chip debug Serial input port

SFR Mapping

The Special Function Registers (SFRs) of the AT89C51RE2 fall into the following categories:

- C51 core registers: ACC, B, DPH, DPL, PSW, SP
- I/O port registers: P0, P1, P2, P3, P4, P5, P6
- Timer registers: T2CON, T2MOD, TCON, TH0, TH1, TH2, TMOD, TL0, TL1, TL2, RCAP2L, RCAP2H
- Serial I/O port registers: SADDR_0, SADEN_0, SBUF_0, SCON_0, SADDR_1, SADEN_1, SBUF_1, SCON_1,
- PCA (Programmable Counter Array) registers: CCON, CCAPMx, CL, CH, CCAPxH, CCAPxL (x: 0 to 4)
- Power and clock control registers: PCON, CKAL, CKCON0_1
- Hardware Watchdog Timer registers: WDTRST, WDTPRG
- Interrupt system registers: IE0, IPL0, IPH0, IE1, IPL1, IPH1
- Keyboard Interface registers: KBE, KBF, KBLS
- 2-wire Interface registers: SSCON, SSCS, SSDAT, SSADR
- SPI registers: SPCON, SPSTR, SPDAT
- BRG (Baud Rate Generator) registers: BRL_0, BRL_1, BDRCON_0, BDRCON_1
- Memory register: FCON, FSTA
- Clock Prescaler register: CKRL
- Others: AUXR, AUXR1, CKCON0, CKCON1, BMSEL

Table 3. C51 Core SFRs

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								
DPH	83h	Data Pointer High byte								

Table 4. System Management SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
PCON	87h	Power Control	SMOD1_0	SMOD0_0	-	POF	GF1	GF0	PD	IDL
AUXR	8Eh	Auxiliary Register 0	-	-	M0	XRS2	XRS1	XRS0	EXTRAM	AO
AUXR1	A2h	Auxiliary Register 1	EES	SP9	U2	-	GF2	0	-	DPS
CKRL	97h	Clock Reload Register	-	-	-	-	-	-	-	-
BMSEL	92h	Bank Memory Select	MBO2	MBO1	MBO0		-	FBS2	FBS1	FBS0
CKCON0	8Fh	Clock Control Register 0	TWIX2	WDX2	PCAX2	SIX2_0	T2X2	T1X2	T0X2	X2
CKCON1	AFh	Clock Control Register 1	-	-	-	-	-	-	SIX2_1	SPIX2

Table 5. Interrupt SFRs

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	B1h	Interrupt Enable Control 1	-	-	-	-	ES_1	ESPI	ETWI	EKBD
IPH0	B7h	Interrupt Priority Control High 0	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
IPL0	B8h	Interrupt Priority Control Low 0	-	PPCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
IPH1	B3h	Interrupt Priority Control High 1	-	-	-	-	PSH_1	SPIH	IE2CH	KBDH
IPL1	B2h	Interrupt Priority Control Low 1	-	-	-	-	PSL_1	SPIL	IE2CL	KBDL

Table 6. Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P0	80h	8-bit Port 0								
P1	90h	8-bit Port 1								
P2	A0h	8-bit Port 2								
P3	B0h	8-bit Port 3								
P4	C0h	8-bit Port 4								

Table 6. Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P5	E8h	8-bit Port 5								
P6	F8h	2-bit Port 5	-	-	-	-	-	-		

Table 7. Flash and EEPROM Data Memory SFR

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
FCON	D1h	Flash Controller Control	FPL3	FPL2	FPL1	FPL0	FPS	FMOD2	FMOD1	FMOD0
FSTA	D3h	Flash Controller Status	FMR					FSE	FLOAD	FBUSY

Table 8. Timer SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
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
TL0	8Ah	Timer/Counter 0 Low Byte								
TH0	8Ch	Timer/Counter 0 High Byte								
TL1	8Bh	Timer/Counter 1 Low Byte								
TH1	8Dh	Timer/Counter 1 High Byte								
WDTRST	A6h	WatchDog Timer Reset								
WDTPRG	A7h	WatchDog Timer Program	-	-	-	-	-	WTO2	WTO1	WTO0
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								
TH2	CDh	Timer/Counter 2 High Byte								
TL2	CCh	Timer/Counter 2 Low Byte								

Table 9. PCA SFRs

Mnemo-nic	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								

Table 9. PCA SFRs (Continued)

Mnemo-nic	Add	Name	7	6	5	4	3	2	1	0
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

Table 10. Serial I/O Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SCON_0	98h	Serial Control 0	FE/SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0
SBUF_0	99h	Serial Data Buffer 0								
SADEN_0	B9h	Slave Address Mask 0								
SADDR_0	A9h	Slave Address 0								
BDRCON_0	9Bh	Baud Rate Control 0				BRR_0	TBCK_0	RBCK_0	SPD_0	SRC_0
BRL_0	9Ah	Baud Rate Reload 0								
SCON_1	C0h	Serial Control 1	FE_1/SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
SBUF_1	C1h	Serial Data Buffer 1								
SADEN_1	BAh	Slave Address Mask 1								
SADDR_1	AAh	Slave Address 1								
BDRCON_1	BCh	Baud Rate Control 1	SMOD1_1	SMOD0_1		BRR_1	TBCK_1	RBCK_1	SPD_1	SRC_1
BRL_1	BBh	Baud Rate Reload 1								

Table 11. SPI Controller SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SPCON	C3h	SPI Control	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
SPSCR	C4h	SPI Status	SPIF		OVR	MODF	SPTE	UARTM	SPTEIE	MODFIE
SPDAT	C5h	SPI Data	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0

Table 12. Two-Wire Interface Controller SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SSCON	93h	Synchronous Serial control	SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0
SSCS	94h	Synchronous Serial Status	SSC4	SSC3	SSC2	SSC1	SSC0	0	0	0
SSDAT	95h	Synchronous Serial Data	SSD7	SSD6	SSD5	SSD4	SSD3	SSD2	SSD1	SSD0
SSADR	96h	Synchronous Serial Address	SSA7	SSA6	SSA5	SSA4	SSA3	SSA2	SSA1	SSGC

Table 13. Keyboard Interface SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
KBLS	9Ch	Keyboard Level Selector	KBLS7	KBLS6	KBLS5	KBLS4	KBLS3	KBLS2	KBLS1	KBLS0
KBE	9Dh	Keyboard Input Enable	KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0
KBF	9Eh	Keyboard Flag Register	KBF7	KBF6	KBF5	KBF4	KBF3	KBF2	KBF1	KBF0

Table below shows all SFRs with their address and their reset value.

Table 14. SFR Mapping

		Bit addressable	Non Bit addressable						
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h		P6 XXXX XX11	CH 0000 0000	CCAP0H XXXX XXXX	CCAP1H XXXX XXXX	CCAP2H XXXX XXXX	CCAP3H XXXX XXXX	CCAP4H XXXX XXXX		FFh
F0h		B 0000 0000								F7h
E8h		P5 1111 1111	CL 0000 0000	CCAP0L XXXX XXXX	CCAP1L XXXX XXXX	CCAP2L XXXX XXXX	CCAP3L XXXX XXXX	CCAP4L XXXX XXXX		EFh
E0h		ACC 0000 0000								E7h
D8h		CCON 00X0 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		FSTA xxxx x000					D7h
C8h		T2CON 0000 0000	T2MOD XXXX XX00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000			CFh
C0h	U2(AUXR1.5) =0	SCON_1 0000 0000	SBUF_1 0000 0000		SPCON 0001 0100	SPSCR 0000 0000	SPDAT XXXX XXXX			C7h
	U2(AUXR1.5) =1	P4 1111 1111								
B8h		IPL0 X000 000	SADEN_0 0000 0000	SADEN1 0000 0000	BRL_1 0000 0000	BDRCON_1 XXX0 0000				BFh
B0h		P3 1111 1111	IEN1 XXXX 0000	IPL1 XXXX 0000	IPH1 XXXX 0111				IPH0 X000 0000	B7h
A8h		IEN0 0000 0000	SADDR_0 0000 0000	SADDR_1 0000 0000					CKCON1 XXXX XX00	AFh
A0h		P2 1111 1111		AUXR1 000x 11x0				WDTRST XXXX XXXX	WDTPRG XXXX X000	A7h
98h		SCON_0 0000 0000	SBUF_0 XXXX XXXX	BRL_0 0000 0000	BDRCON_0 XXX0 0000	KBLS 0000 0000	KBE 0000 0000	KBF 0000 0000		9Fh
90h		P1 1111 1111		BMSEL 0000 0YYY	SSCON 0000 0000	SSCS 1111 1000	SSDAT 1111 1111	SSADR 1111 1110	CKRL 1111 1111	97h
88h		TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR XX00 1000	CKCON0 0000 0000	8Fh
80h		P0 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 00X1 0000	87h
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Reserved

Enhanced Features

In comparison to the original 80C52, the AT89C51RE2 implements some new features, which are:

- X2 option
- Dual Data Pointer
- Extended RAM
- Extended stack
- Programmable Counter Array (PCA)
- Hardware Watchdog
- SPI interface
- 4-level interrupt priority system
- power-off flag
- ONCE mode
- ALE disabling
- Enhanced features on the UART and the timer 2

X2 Feature

The AT89C51RE2 core needs only 6 clock periods per machine cycle. This feature called ‘X2’ provides the following advantages:

- Divide frequency crystals by 2 (cheaper crystals) while keeping same CPU power.
- Save power consumption while keeping same CPU power (oscillator power saving).
- Save power consumption by dividing dynamically the operating frequency by 2 in operating and idle modes.
- Increase CPU power by 2 while keeping 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 software.

Description

The clock for the whole circuit and peripherals is first divided by two before being used by the CPU core and the peripherals.

This allows any cyclic ratio to be accepted on XTAL1 input. In X2 mode, as this divider is bypassed, the signals on XTAL1 must have a cyclic ratio between 40 to 60%.

Figure 2 shows the clock generation block diagram. X2 bit is validated on the rising edge of the XTAL1÷2 to avoid glitches when switching from X2 to STD mode. Figure 3 shows the switching mode waveforms.

Figure 2. Clock Generation Diagram

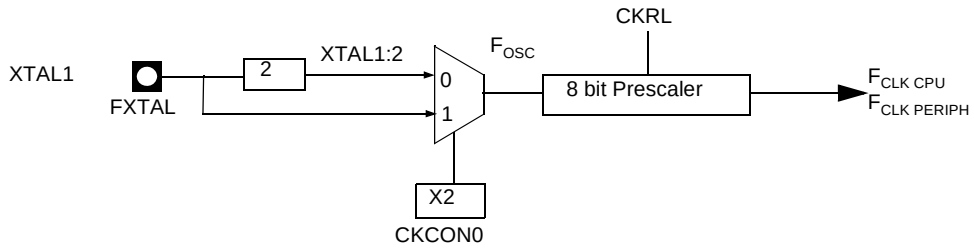
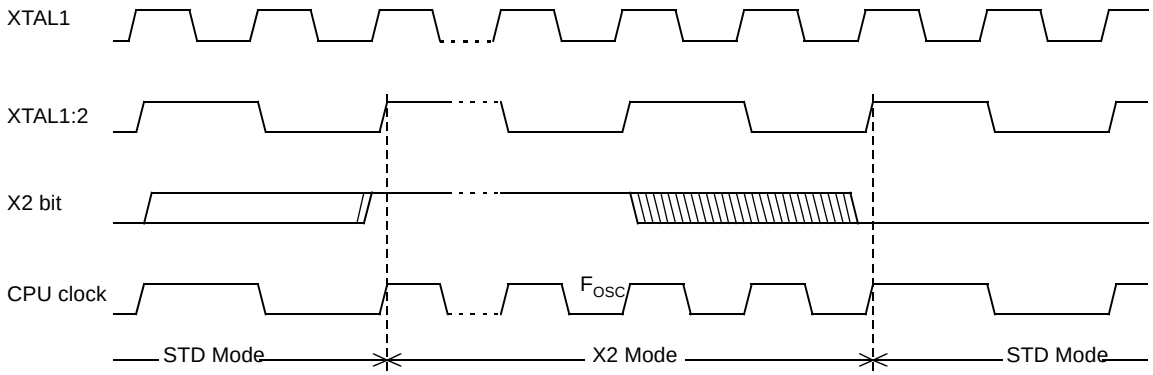


Figure 3. Mode Switching Waveforms



The X2 bit in the CKCON0 register (see Table 15) allows a switch from 12 clock periods per instruction to 6 clock periods and vice versa. At reset, the speed is set according to X2 bit of the Fuse Configuration Byte (FCB). By default, Standard mode is active. Setting the X2 bit activates the X2 feature (X2 mode).

The T0X2, T1X2, T2X2, UartX2, PcaX2, and WdX2 bits in the CKCON0 register (See Table 15.) and SPIX2 bit in the CKCON1 register (see Table 16) allows a switch from standard peripheral speed (12 clock periods per peripheral clock cycle) to fast peripheral speed (6 clock periods per peripheral clock cycle). These bits are active only in X2 mode.

Table 15. CKCON0 Register

CKCON0 - Clock Control Register (8Fh)

7	6	5	4	3	2	1	0
TWIX2	WDX2	PCAX2	SIX2_0	T2X2	T1X2	T0X2	X2

Bit Number	Bit Mnemonic	Description
7	TWIX2	**2-wire cloTBck** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect) Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
6	WDX2	**Watchdog Clock** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared 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** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
4	SIX2_0	**Enhanced UART0 Clock (Mode 0 and 2)** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
3	T2X2	**Timer2 Clock** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
2	T1X2	**Timer1 Clock** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
1	T0X2	**Timer0 Clock** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
0	X2	**CPU Clock** Cleared to select 12 clock periods per machine cycle (STD mode) for CPU and all the peripherals. Set to select 6clock periods per machine cycle (X2 mode) and to enable the individual peripherals'X2' bits. Programmed by hardware after Power-up regarding Hardware Security Byte (HSB), Default setting, X2 is cleared.

Reset Value = X000 000'HSB. X2'b (See "Fuse Configuration Byte: FCB")
Not bit addressable

Table 16. CKCON1 Register

CKCON1 - Clock Control Register (AFh)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SIX2_1	SPIX2

Bit Number	Bit Mnemonic	Description
7	-	**Reserved**
6	-	**Reserved**
5	-	**Reserved**
4	-	**Reserved**
3	-	**Reserved**
2	-	**Reserved**
1	SIX2_1	**Enhanced UART1 Clock (Mode 0 and 2)** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Cleared to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
0	SPIX2	**SPI** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect). Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.

Reset Value = XXXX XX00b
Not bit addressable

Dual Data Pointer Register DPTR

The additional data pointer can be used to speed up code execution and reduce code size.

The dual DPTR structure is a way by which the chip will specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS = AUXR1.0 (see Table 17) that allows the program code to switch between them (Refer to Figure 4).

Figure 4. Use of Dual Pointer

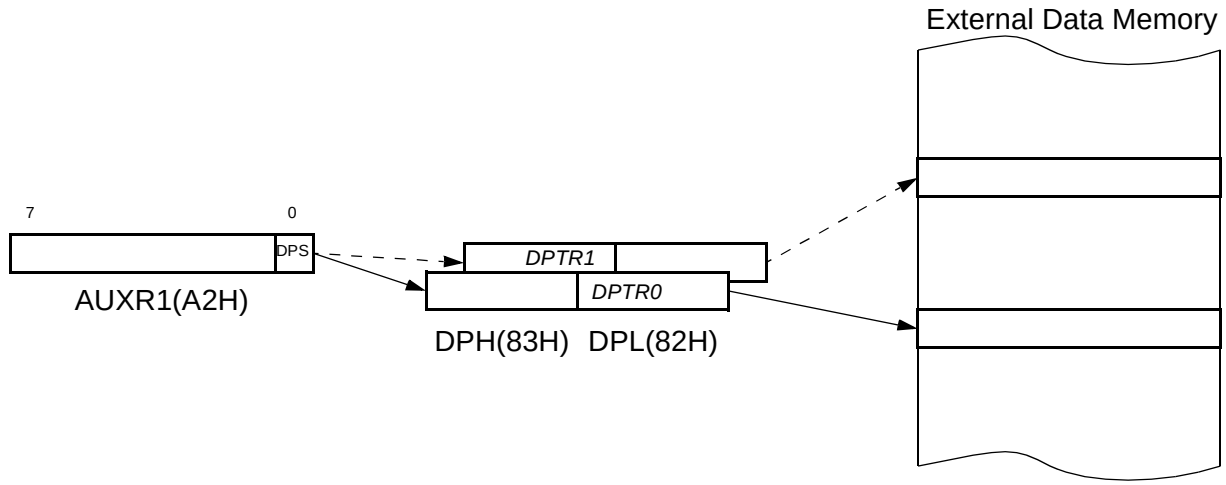


Table 17. AUXR1 register

AUXR1- Auxiliary Register 1(0A2h)

7	6	5	4	3	2	1	0
EES	**SP9**	**U2**	**-**	**GF2**	**0**	**-**	**DPS**

Bit Number	Bit Mnemonic	Description
7	EES	**Enable Extended Stack** This bit allows the selection of the stack extended mode. Set to enable the extended stack Clear to disable the extended stack (default value)
6	SP9	**Stack Pointer 9th Bit** This bit has no effect when the EES bit is cleared. Set when the stack pointer belongs to the XRAM memory space Cleared when the stack pointer belongs to the 256bytes of internal RAM.
5	U2	**P4 bit addressable** Clear to map SCON_1 register at C0h sfr address Set to map P4 port register at C0h address.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	GF2	**This bit is a general purpose user flag. ***
2	0	**Always cleared.**
1	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	DPS	**Data Pointer Selection** Cleared to select DPTR0. Set to select DPTR1.

Reset Value: XX0X XX0X0b

Not bit addressable

Note: *Bit 2 stuck at 0; this allows to use INC AUXR1 to toggle DPS without changing GF3.

ASSEMBLY LANGUAGE

```
; Block move using dual data pointers
; Modifies DPTR0, DPTR1, A and PSW
; note: DPS exits opposite of entry state
; unless an extra INC AUXR1 is added
;
00A2    AUXR1 EQU 0A2H
;
0000 909000MOV DPTR,#SOURCE ; address of SOURCE
0003 05A2 INC AUXR1 ; switch data pointers
0005 90A000 MOV DPTR,#DEST ; address of DEST
0008    LOOP:
0008 05A2 INC AUXR1 ; switch data pointers
000A E0 MOVX A,@DPTR ; get a byte from SOURCE
000B A3 INC DPTR ; increment SOURCE address
000C 05A2 INC AUXR1 ; switch data pointers
000E F0 MOVX @DPTR,A ; write the byte to DEST
000F A3 INC DPTR ; increment DEST address
```

```
0010 70F6JNZ LOOP ; check for 0 terminator
0012 05A2 INC AUXR1 ; (optional) restore DPS
```

INC is a short (2 bytes) and fast (12 clocks) way to manipulate the DPS bit in the AUXR1 SFR. However, note that the INC instruction does not directly force the DPS bit to a particular state, but simply toggles it. In simple routines, such as the block move example, only the fact that DPS is toggled in the proper sequence matters, not its actual value. In other words, the block move routine works the same whether DPS is '0' or '1' on entry. Observe that without the last instruction (INC AUXR1), the routine will exit with DPS in the opposite state.

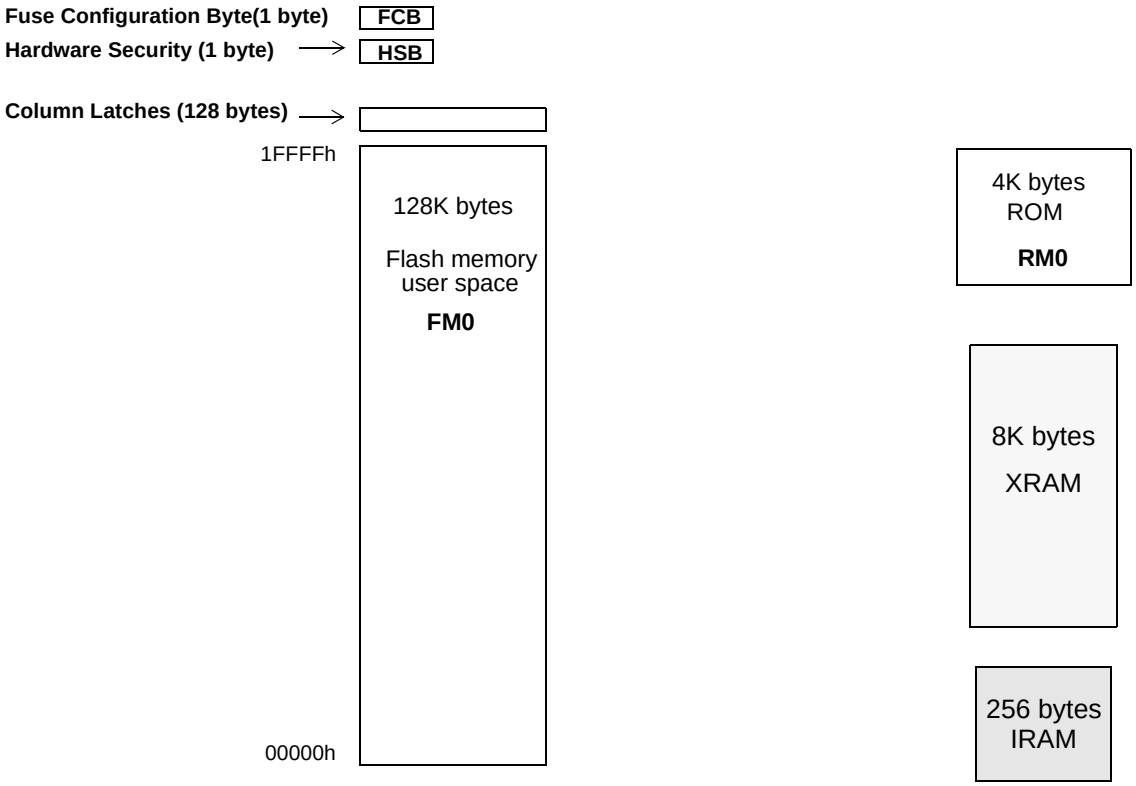
Memory Architecture

AT89C51RE2 features several on-chip memories:

- Flash memory:
  containing 128 Kbytes of program memory (user space) organized into 128 bytes pages.
- Boot ROM:
  4K bytes for boot loader.
- 8K bytes internal XRAM

Physical memory organisation

Figure 5. Physical memory organisation



Expanded RAM (XRAM)

The AT89C51RE2 provides additional Bytes of random access memory (RAM) space for increased data parameter handling and high level language usage.

AT89C51RE2 devices have expanded RAM in external data space configurable up to 8192bytes (see Table 18.).

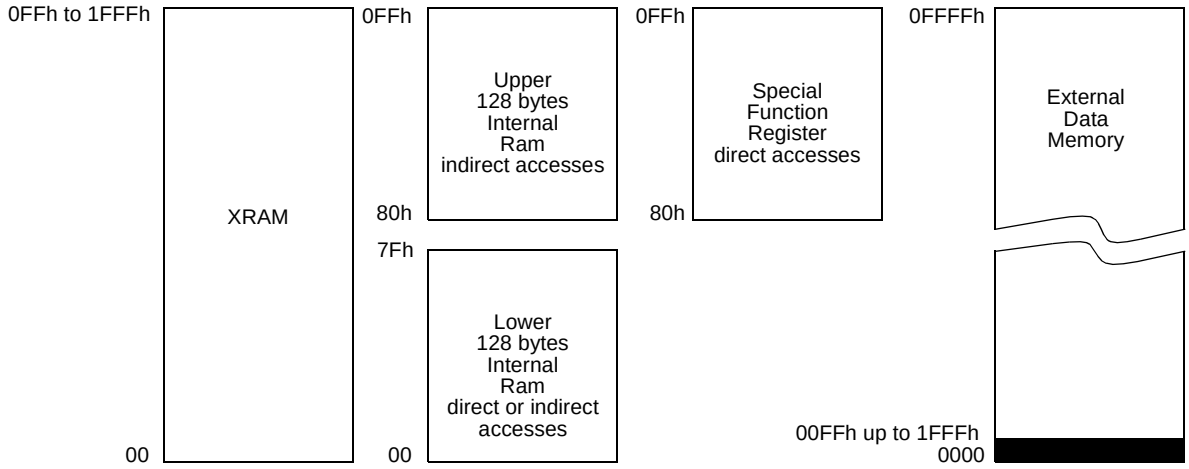
The AT89C51RE2 has internal data memory that is mapped into four separate segments.

The four segments are:

1. The Lower 128 bytes of RAM (addresses 00h to 7Fh) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80h to FFh) are indirectly addressable only.
3. The Special Function Registers, SFRs, (addresses 80h to FFh) are directly addressable only.
4. The expanded RAM bytes are indirectly accessed by MOVX instructions, and with the EXTRAM bit cleared in the AUXR register (see Table 18).

The lower 128 bytes can be accessed by either direct or indirect addressing. The Upper 128 bytes can be accessed by indirect addressing only. The Upper 128 bytes occupy the same address space as the SFR. That means they have the same address, but are physically separate from SFR space.

Figure 6. Internal and External Data Memory Address



When an instruction accesses an internal location above address 7Fh, the CPU knows whether the access is to the upper 128 bytes of data RAM or to SFR space by the addressing mode used in the instruction.

- Instructions that use direct addressing access SFR space. For example: MOV 0A0H, # data, accesses the SFR at location 0A0h (which is P2).
- Instructions that use indirect addressing access the Upper 128 bytes of data RAM. For example: MOV @R0, # data where R0 contains 0A0h, accesses the data byte at address 0A0h, rather than P2 (whose address is 0A0h).
- The XRAM bytes can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory which is physically located on-chip, logically occupies the first bytes of external data memory. The bits XRS0 and XRS1 are used to hide a part of the available XRAM as explained in Table 18. This can be useful if external peripherals are mapped at addresses already used by the internal XRAM.

- With EXTRAM = 0, the XRAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to XRAM will not affect ports P0, P2, P3.6 (WR) and P3.7 (RD). For example, with EXTRAM = 0, MOVX @R0, # data where R0 contains 0A0H, accesses the XRAM at address 0A0H rather than external memory. An access to external data memory locations higher than the accessible size of the XRAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51, with P0 and P2 as data/address busses, and P3.6 and P3.7 as write and read timing signals. Accesses to XRAM above 0FFH can only be done by the use of DPTR.
- With EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard 80C51.MOVX @ Ri will provide an eight-bit address multiplexed with data on Port0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a sixteen-bit address. Port2 outputs the high-order eight address bits (the contents of DPH) while Port0 multiplexes the low-order eight address bits (DPL) with data. MOVX @ Ri and MOVX @DPTR will generate either read or write signals on P3.6 ($\overline{WR}$) and P3.7 ($\overline{RD}$).

The stack pointer (SP) may be located anywhere in the 256 bytes RAM (lower and upper RAM) internal data memory. The stack may be located in the 256 lower bytes of the XRAM by activating the extended stack mode (see EES bit in AUXR1).

The M0 bit allows to stretch the XRAM timings; if M0 is set, the read and write pulses are extended from 6 to 30 clock periods. This is useful to access external slow peripherals.

Registers

Table 18. AUXR Register

AUXR - Auxiliary Register (8Eh)

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	M0	**Pulse length** Cleared to stretch MOVX control: the RD/ and the WR/ pulse length is 6 clock periods (default). Set to stretch MOVX control: the RD/ and the WR/ pulse length is 30 clock periods.
4-2	XRS2:0	**XRAM Size** XRS2 XRS1 XRS0 XRAM size 0 0 0 256 bytes 0 0 1 512 bytes 0 1 0 768 bytes 0 1 1 1024 bytes 1 0 0 1792 bytes 1 0 1 2048 bytes 1 1 0 4096 bytes 1 1 1 8192 bytes (default)
1	EXTRAM	**EXTRAM bit** Cleared to access internal XRAM using movx @ Ri/ @ DPTR. Set to access external memory. Programmed by hardware after Power-up regarding Hardware Security Byte (HSB), default setting, XRAM selected.
0	AO	**ALE Output bit** Cleared, ALE is emitted at a constant rate of 1/6 the oscillator frequency (or 1/3 if X2 mode is used). (default) Set, ALE is active only during a MOVX or MOVC instruction is used.

Reset Value = XX01 1100b
Not bit addressable

Extended Stack

The lowest bytes of the XRAM may be used to allow extension of the stack pointer.

The extended stack allows to extend the standard C51 stack over the 256 bytes of internal RAM. When the extended stack mode is activated (EES bit in AUXR1), the stack pointer (SP) can grow in the lower 256 bytes of the XRAM area.

The stack extension consists in a 9 bits stack pointer where the ninth bit is located in SP9 (bit 6 of AUXR1). The SP9 then indicates if the stack pointer belongs to the internal RAM (SP9 cleared) or to the XRAM memory (SP9 set).

To ensure backward compatibility with standard C51 architecture, the extended mode is disable at chip reset.

Figure 7. Stack modes

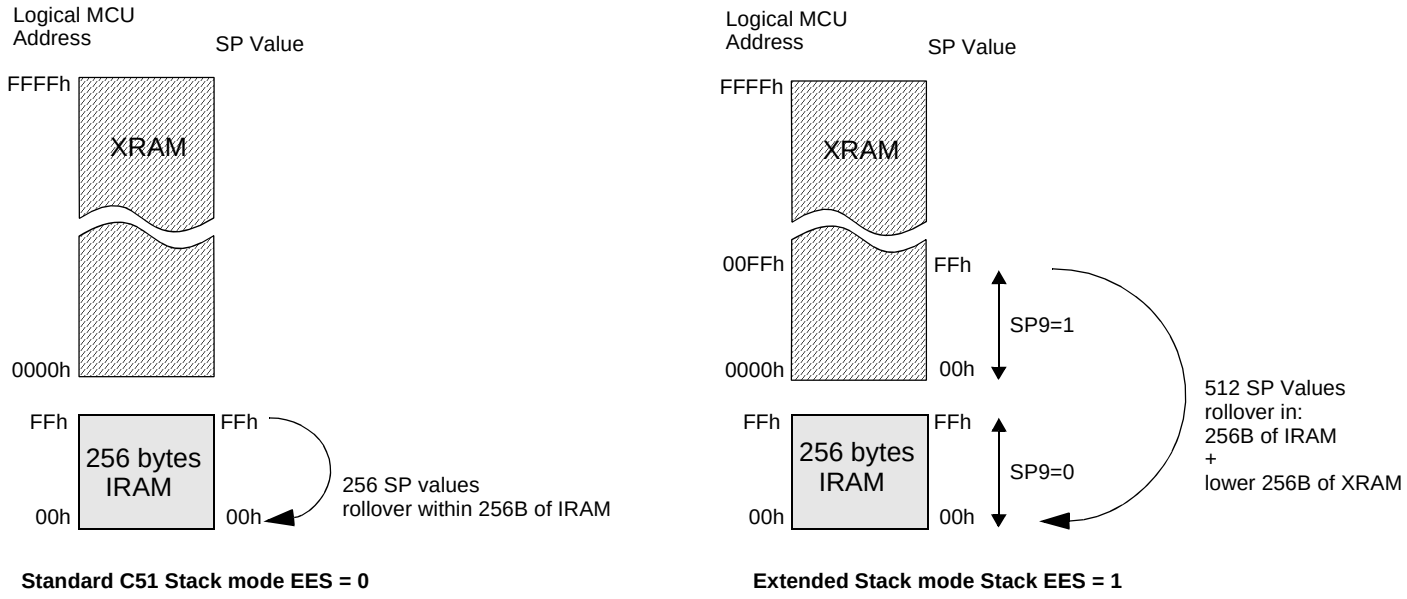


Figure 8. AUXR1 register

AUXR1- Auxiliary Register 1(0A2h)

7	6	5	4	3	2	1	0
EES	**SP9**	**U2**	**-**	**GF2**	**0**	**-**	**DPS**

Bit Number	Bit Mnemonic	Description
7	EES	**Enable Extended Stack** This bit allows the selection of the stack extended mode. Set to enable the extended stack Clear to disable the extended stack (default value)

Bit Number	Bit Mnemonic	Description
6	SP9	**Stack Pointer 9th Bit** This bit has no effect when the EES bit is cleared. Set when the stack pointer belongs to the XRAM memory space Cleared when the stack pointer belongs to the 256bytes of internal RAM. Set and cleared by hardware. Can only be read.
5	U2	**P4 bit addressable** Clear to map SCON_1 register at C0h sfr address Set to map P4 port register at C0h address.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	GF2	**This bit is a general purpose user flag. ***
2	0	**Always cleared.**
1	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	DPS	**Data Pointer Selection** Cleared to select DPTR0. Set to select DPTR1.

Reset Value = 00XX 00X0b
Not bit addressable

Flash Memory

General Description

The Flash memory increases EPROM and ROM functionality with in-circuit electrical erasure and programming. It contains 128K bytes of program memory organized in 1024 pages of 128 bytes. This memory is both parallel and serial In-System Programmable (ISP). ISP allows devices to alter their own program memory in the actual end product under software control. A default serial loader (bootloader) program allows ISP of the Flash.

The programming does not require external high programming voltage. The necessary high programming voltage is generated on-chip using the standard V_{CC} pins of the microcontroller.

Features

- Flash internal program memory.
- Boot vector allows user provided Flash loader code to reside anywhere in the Flash memory space. This configuration provides flexibility to the user.
- Default loader in Boot Flash allows programming via the serial port without the need of a user provided loader.
- Up to 64K byte external program memory if the internal program memory is disabled (EA = 0).
- Programming and erase voltage with standard 5V or 3V V_{CC} supply.

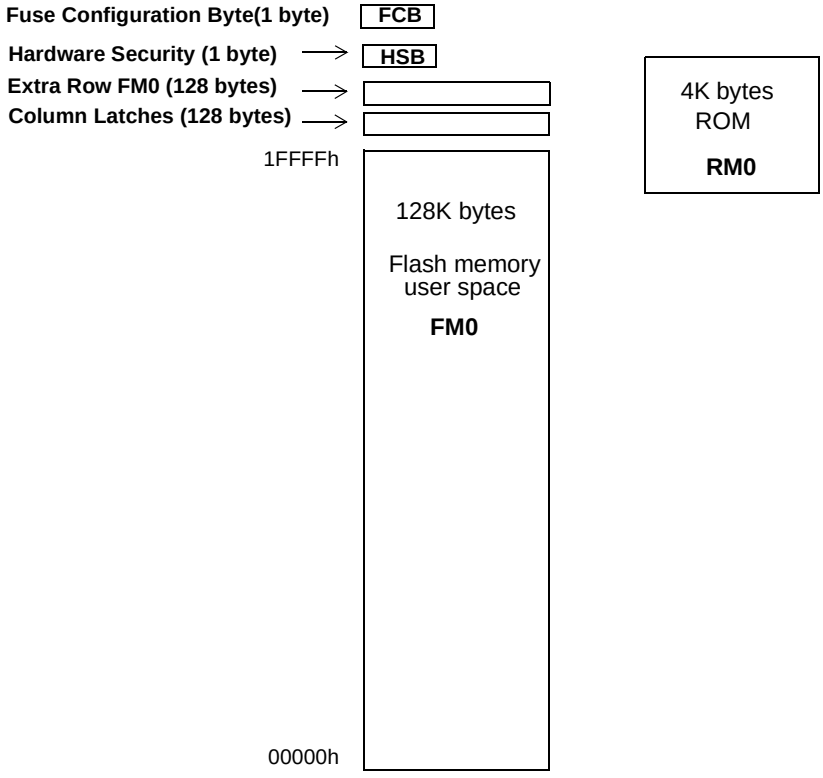
Flash memory organization

AT89C51RE2 features several on-chip memories:

- Flash memory FM0:
  containing 128 Kbytes of program memory (user space) organized into 128 bytes pages.
- Boot ROM RM0:
  4K bytes for boot loader.
- 8K bytes internal XRAM

Physical memory organisation

Figure 9. Physical memory organisation



On-Chip Flash memory

The AT89C51RE2 implements up to 128K bytes of on-chip program/code memory. Figure 9 and Figure 10 shows the partitioning of internal and external program/code memory spaces according to EA value.

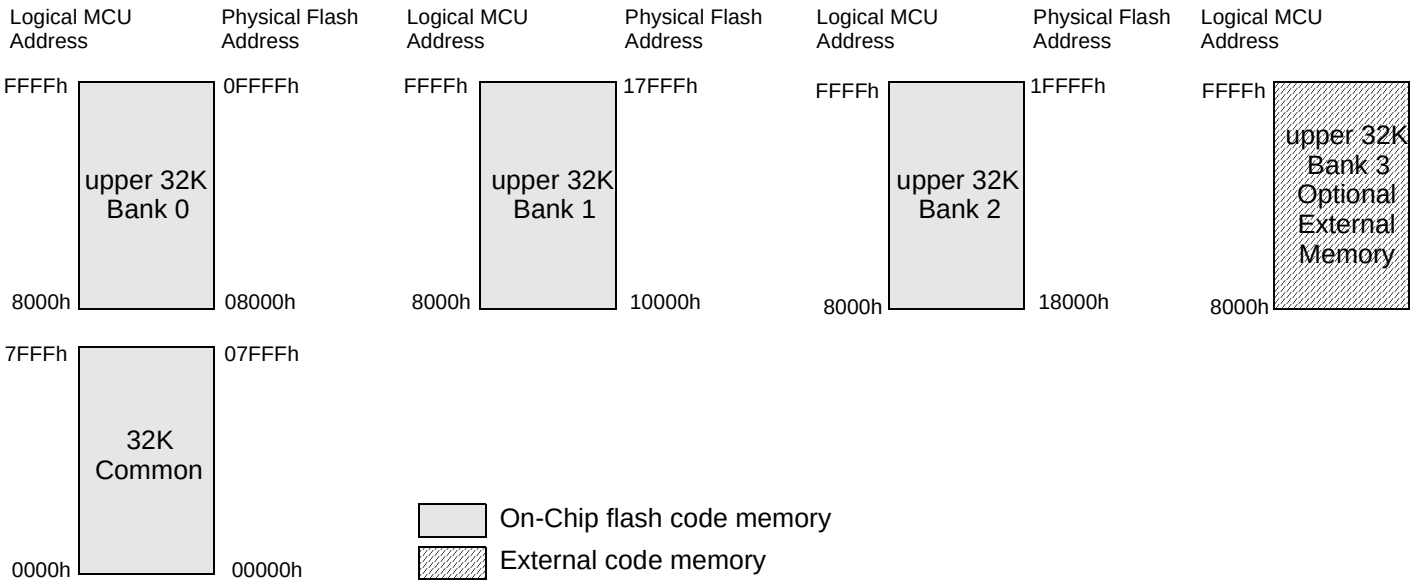
The memory partitioning of the 8051 core microcontroller is typical a Harvard architecture where program and data areas are held in separate memory areas. The program and data memory areas use the same physical address range from 0000H-FFFFH and a 8 bit instruction code/data format.

To access more than 64kBytes of code memory, without modifications of the MCU core, and development tools, the bank switching method is used.

The internal program memory is expanded to 128kByte in the ´Expanded Configuration', the data memory remains in the ´Normal Configuration´. The program memory is split into four 32 kByte banks (named Bank 0-2). The MCU core still addresses up to 64kBytes where the upper 32Kbytes can be selected between 3 32K bytes bank of on-chip flash memory. The lower 32K bank is used as common area for interrupt subroutines, bank switching and functions calls between banks.

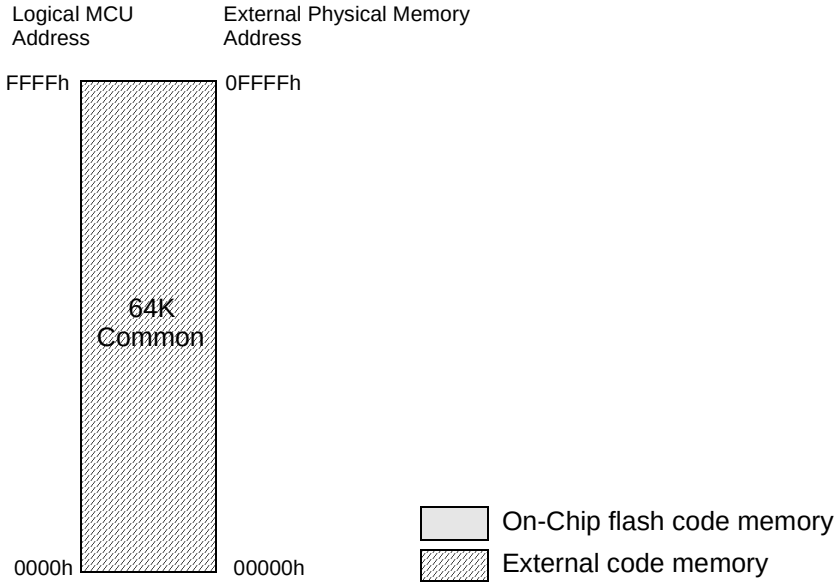
The AT89C51RE2 also implements an extra upper 32K bank (Bank3) that allows external code execution.

Figure 10. Program/Code Memory Organization EA=1



When EA=0, the on-chip flash memory is disabled and the MCU core can address only up to 64kByte of external memory (none of the on-chip flash memory FM0 banks or RM0 can be mapped and executed).

Figure 11. Program/Code Memory Organization EA=0

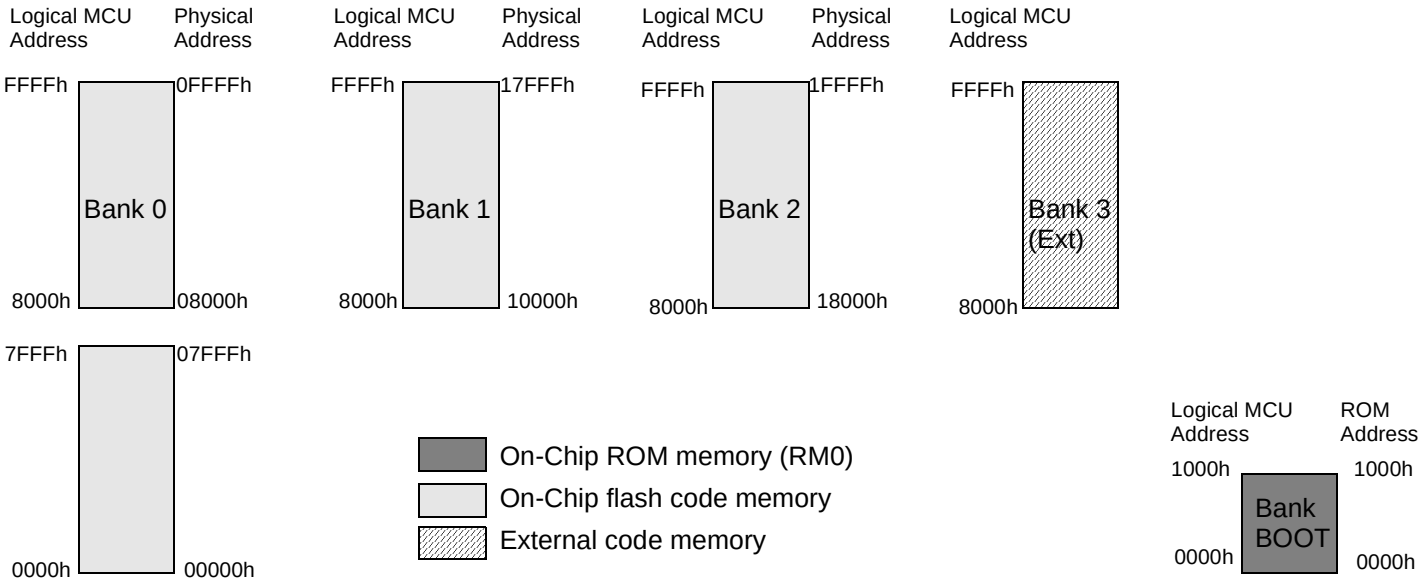


On-Chip ROM bootloader

The On-chip ROM bootloader (RM0) is enable only for ISP operations after reset (bootloader execution). The RM0 memory area belongs to a logical addressable memory space called ‘Bank Boot’.

RM0 cannot be activated from the On-chip flash memory. It means that it is not possible activate the Bank Boot area by software (it prevents any RM0 execution and flash corruption from the user application).

RM0 logical area consists in an independent code execution memory area of 4K bytes starting at logical 0x0000 address (it allows the use of the interrupts in the bootloader execution).



Boot process

The BRV2-0 bits of the FCB (see Table 20 on page 34), the EA pin value upon reset and the presence of the external hardware conditions, allow to modify the default reset vector of the AT89C51RE2.

The Hardware conditions (EA = 1, PSEN = 0) during the Reset falling edge force the on-chip bootloader execution. This allows an application to be built that will normally execute the end user's code but can be manually forced into default ISP operation. The hardware conditions allows to force the enter in ISP mode whatever the configurations bits.

Figure 12. Boot Reset vector configuration

EA pin	Hardware conditions	BRV2-0	MCU reset vector
0	X	X	External Code at address 0x0000
1	YES	X	RM0 at address 0x0000 (ATMEL Bootloader)
	NO	1 1 1	FM0 at address 0x0000 with bank0 mapped
		1 1 0	FM0 at address 0xFFFC in Bank 0
		1 0 1	FM0 at address 0xFFFC in Bank 1
		1 0 0	FM0 at address 0xFFFC in Bank 2
		0 1 1	RM0 at address 0x0000 (ATMEL Bootloader)
		0 1 0	Reserved (FM0 at address 0x0000 with bank 0 mapped)
		0 0 1	
		0 0 0	

FM0 Memory Architecture

The FM0 flash memory is made up of 5 blocks:

1. The memory array (user space) 128K bytes
2. The Extra Row also called FM0 XAF
3. The Hardware security bits (HSB)
4. The Fuse Configuration Byte (FCB)
5. The column latch

User Space

This space is composed of a 128K bytes Flash memory organized in 1024 pages of 128 bytes. It contains the user's application code. This block can be access in Read/write mode from FM0 and boot memory area. (When access in write mode from FM0, the CPU core enter pseudo idle mode).

Extra Row (XRow or XAF)

This row is a part of FM0 and has a size of 128 bytes. The extra row (XAF) may contain information for boot loader usage.This block can be access in Read/write mode from FM0 and boot memory area. (When access in write mode from FM0, the CPU core enter pseudo idle mode).

Hardware security Byte (HSB)

The Hardware security Byte is a part of FM0 and has a size of 1 byte.
The 8 bits can be read/written by software (from FM0 or RM0) and written by hardware in parallel mode.

The HSB bits can be written to '0' without any restriction (increase the security level of the chip), but can be written to '1' only when the corresponding memory area of the lock bits was full chip erased.

Table 19. Hardware Security Byte (HSB)

7	6	5	4	3	2	1	0
-	-	-	-	-	FLB2	FLB1	FLB0

Bit Number	Bit Mnemonic	Description
7	-	**Unused**
6-4	-	**Reserved**
3	-	**Unused**
2-0	FLB2-0	**FM0 Memory Lock Bits** See Table 32 on page 52

Fuse Configuration Byte (FCB)

The Fuse configuration byte is a part of FM0.
The 8 bits read/written by software (from FM0 or RM0) and written by hardware in parallel mode.

Table 20. Fuse Configuration Byte (FCB)

7	6	5	4	3	2	1	0
X2	**-**	**-**	**-**	**-**	**BRV2**	**BRV1**	**BRV0**

Bit Number	Bit Mnemonic	Description
7	X2	**X2 Mode** Programmed ('0' value) to force X2 mode (6 clocks per instruction) after reset Unprogrammed ('1' value) to force X1 mode, Standard Mode, after reset (Default)
6-3	-	**Unused**
2-0	BRV2-0-	**Boot Reset Vector** These bits allow to configure the reset vector of the product according to the following values: 1 1 1: Reset at address 0x0000 of FM0 with Bank0 mapped 1 1 0: Reset at address 0xFFFC of Bank 0 1 0 1: Reset at address 0xFFFC of Bank 1 1 0 0: Reset at address 0xFFFC of Bank 2 0 1 1: Reset at address 0x0000 of RM0 (Internal ROM bootloader execution) 0 1 0: Reserved for further extension but same as 1 1 1 0 0 1: Reserved for further extension but same as 1 1 1 0 0 0: Reserved for further extension but same as 1 1 1

Column latches

The column latches, also part of FM0, has a size of one page (128 bytes).
The column latches are the entrance buffers of the three previous memory locations (user array, XROW, Hardware security byte and Fuse Configuration Byte).

This block is write only from FM0, RM0.

Cross Memory Access Description overview

The FM0 memory can be programmed from RM0 without entering idle mode.

Programming FM0 from FM0 makes the CPU core entering “pseudo idle” mode.

In the pseudo idle mode, the code execution is halted, the peripherals are still running (like standard idle mode) but all interrupt are delayed to the end of this mode. There are fours ways of exiting pseudo idle mode:

- At the end of the regular flash programming operation
- Reset the chip by external reset
- Reset the chip by hardware watchdog
- Reset the chip by PCA watchdog

Programming FM0 from external memory code (EA=0 or EA=1,with Bank3 active) is impossible.

If a reset occurs during flash programming the target page could be incompletely erased or programmed, but any other memory location (FM0, RAM, XRAM) remain unchanged.

The Table 21 shows all software flash access allowed.

Table 21. Cross Memory Access

Code executing from	Action	**FM0 (user Flash)**	RM0 (boot ROM)
FM0 (user Flash)	Read	ok	Denied
	Load column latch	ok	N.A.
	Write	ok (pseudo idle mode)	N.A.
RM0 (boot ROM)	Read	ok	ok
	Load column latch	ok	N.A.
	Write	ok	N.A.
External memory EA = 0 or EA=1, Bank3	Read	(1)	Denied
	Load column latch	Denied	N.A.
	Write	Denied	N.A.

1. Depends of general lock bits configuration

N.A. Not applicable

Access and Operations Descriptions

FM0 FLASH Registers

The CPU interfaces to the flash memory through the FCON register, AUXR1 register and FSTA register.

These registers are used to map the columns latch, HSB, FCB and extra row in the working data or code space.

BMSEL Register

Table 22. BMSEL Register

BMSEL Register (S:92h)
Bank Memory Select

7	6	5	4	3	2	1	0
MBO2	**MBO1**	**MBO0**			**FBS2**	**FBS1**	**FBS0**

Bit Number	Bit Mnemonic	Description
7-5	MBO2:0	**Memory Bank Operation** These bits select the target memory bank for flash write or read operation. These bits allows to read or write the on-chip flash memory from one upper 32K bytes to another one. 0 X X: The on-chip flash operation target banked is the same as FBS2:0 1 0 0: The target memory bank is forced to Bank0 1 0 1: The target memory bank is forced to Bank1 1 1 0: The target memory bank is forced to Bank2 1 1 1: The target memory bank is forced to Bank3 (optional External bank)
4-3		**Reserved**
2-0	FBS2:0	**Fetch Bank Selection** These bits select the upper 32K bytes execution bank: FBS1:0 can be read/write by software. FBS2 is read-only by software (the Boot bank can not be mapped from FM0) 0 0 0 Bank0 0 0 1 Bank1 0 1 0 Bank2 0 1 1 Bank3 (optionnal external bank) **1 X X Boot Bank (Read only)** Upon reset FBS2:0 is initialized according to BRV2:0 configuration bits in FCB.

Reset Value= 0000 0YYYb (where YYY depends on BRV2:0 value in Fuse Configuration Byte)

FCON Register

Table 23. FCON Register

FCON Register (S:D1h)
Flash Control Register

7	6	5	4	3	2	1	0
FPL3	**FPL2**	**FPL1**	**FPL0**	**FPS**	**FMOD2**	**FMOD1**	**FMOD0**

Bit Number	Bit Mnemonic	Description
7-4	FPL3:0	**Programming Launch Command Bits** Write 5Xh followed by AXh to launch the programming according to FMOD2:0. (see Table 26.)
3	FPS	**Flash Map Program Space** When this bit is set: The MOVX @DPTR, A instruction writes in the columns latches space When this bit is cleared: The MOVX @DPTR, A instruction writes in the regular XDATA memory space
2-0	FMOD2:0	**Flash Mode** These bits allow to select the target memory area and operation on FM0 See Table 25.

Reset Value= 0000 0000b

Table 24. FSTA Register

FSTA Register (S:D3h)
Flash Status Register

7	6	5	4	3	2	1	0
FMR	**-**	**-**	**-**	**-**	**FSE**	**FLOAD**	**FBUSY**

Bit Number	Bit Mnemonic	Description
7	FMR	**Flash Movc Redirection** When code is executed from RM0 (and only RM0), this bit allow the MOVC instruction to be redirected to FM0. Clear this bit to allow MOVC instruction to read FM0 Set this bit to allow MOVC instruction to read RM0 This bit can be written only from RM0 (on-chip ROM bootloader execution).
6-3	-	**unused**
2	FSE	**Flash sequence error** Set by hardware when the flash activation sequencers FCON 5X and MOV FCON AX) is not correct (See Error Report Section) Clear by software or clear by hardware if the last activation sequence was correct (previous error is canceled)
1	FLOAD	**Flash Columns latch loaded** Set by hardware when the first data is loaded in the column latches. Clear by hardware when the activation sequence succeeds (flash write success, or reset column latch success)
0	FBUSY	**Flash Busy** Set by hardware when programming is in progress. Clear by hardware when programming is done. Can not be changed by software.

Reset Value= ‘R’xxx x000b

Where ‘R’ depends on the reset conditions: If RM0 is executed after Reset R=1, if FM0 is executed after reset R=0

Mapping of the Memory Space

By default, the user space is accessed by MOVC A, @A+DPTR instruction for read only. Setting FPS bit in FCON register takes precedence on the EXTRAM bit in AUXR register.

The other memory spaces (user, extra row, hardware security) are made accessible in the code segment by programming bits FMOD2:0 in FCON register in accordance with Table 25. A MOVC instruction is then used for reading these spaces.

Thanks to the columns latches access, it is possible to write FM0 array, HSB and extra row blocks. The column latches space is made accessible by setting the FPS bit in FCON register. Writing is possible from 0000h to FFFFh, address bits 6 to 0 are used to select an address within a page while bits 14 to 7 are used to select the programming address of the page.

Table 25. .FM0 blocks select bits

FMOD2	FMOD1	FMOD0	Adressable Space
0	0	0	FM0 array(0000h-FFFFh)
0	0	1	Extra Row(00h-80h)
0	1	0	Erase FM0
0	1	1	Column latches reset
1	0	0	HSB
1	0	1	FCB
1	1	0	Reserved
1	1	1	

Launching flash commands (activation sequence)

FPL3:0 bits in FCON register are used to secure the launch of programming. A specific sequence must be written in these bits to unlock the write protection and to launch the operation. This sequence is 5xh followed by Axh. Table 26 summarizes the memory spaces to program according to FMOD2:0 bits.

Table 26. FM0 Programming Sequences

	Write to FCON					
	FPL3:0	**FPS**	**FMOD2**	**FMOD1**	**FMOD0**	**Operation**
FM0	5	X	0	0	0	No action
	A	X	0	0	0	Write the column latches in FM0
XAF FM0	5	X	0	0	1	No action
	A	X	0	0	1	Write the column latches in FM0 extra row space
Erase FM0	5	X	0	1	0	No action
	A	X	0	1	0	Full erase FM0 memory area
Reset FM0 Column Latches	5	X	0	1	1	No action
	A	X	0	1	1	Reset the FM0 column latches
HSB	5	X	1	0	0	No action
	A	X	1	0	0	Write the hardware Security byte (HSB) See (4)
FCB	5	X	1	0	1	No action
	A	X	1	0	1	Write the Fuse Configuration Byte (FCB)
Reserved	5	X	1	1	0	No action
	A	X	1	1	0	
Reserved	5	X	1	1	1	
	A	X	1	1	1	

Note:
1. The sequence 5xh and Axh must be executed without instructions between them otherwise the programming is not executed (see flash status register).
2. The sequence 5xh and Axh can be executed with the different FMOD0, FMOD1 values, the last FMOD1:0 value latches the destination target.
3. When the FMOD2 bit is set (corresponding to the serial number field code) no write operation can be performed.
4. Only the bits corresponding to the previously “full erase” memory space can be written to one.

Loading the Column Latches

Any number of data from 0 byte to 128 bytes can be loaded in the column latches. The data written in the column latches can be written in a none consecutive order. The DPTR allows to select the address of the byte to load in the column latches.

The page address to be written (target page in FM0) is given by the last address loaded in the column latches and when this page belongs to the upper 32K bytes of the logical addressable MCU space, the target memory bank selection is performed upon the MBO2:0 value during the last address loaded.

When 0 byte is loaded in the column latches the activation sequence (5xh, Axh in FCON) does not launch any operations. The FSE bit in FSTA register is set.

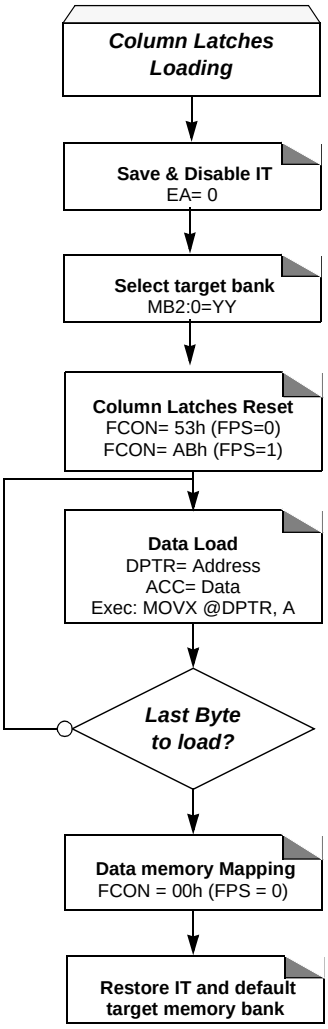
When a current flash write operation is on-going (FBUSY is set), it is impossible to load the columns latches before the end of flash programming process (the write operation in the columns latches is not performed, and the previous columns latches content is not overwritten).

When programming is launched, an **automatic erase** of the entire memory page is first performed, then programming is effectively done. Thus no page or block erase is needed and only the loaded data are programmed in the corresponding page. The unloaded data of the target memory page are programmed at 0xFF value (automatic page erase value).

The following procedure is used to load the column latches and is summarized in Figure 13:

- Disable interrupt and map the column latch space by setting FPS bit.
- Select the target memory bank (for page address larger than 32K)
- Map the column latch
- Reset the column latch
- Load the DPTR with the address to write.
- Load Accumulator register with the data to write.
- Execute the MOVX @DPTR, A instruction, and only this one (no MOVX @Ri, A).
- If needed loop the last three instructions until the page is completely loaded.
- Unmap the column latch if needed (it can be left mapped) and Enable Interrupt

Figure 13. Column Latches Loading Procedure



Note: The last page address used when loading the column latch is the one used to select the page programming address.

Note: The value of MB02:0 during the last load gives the upper 32K bytes bank target selection.

Note: The execution of this sequence when BUSY flag is set leads to the no-execution of the write in the column latches (the previous loaded data remains unchanged).

Writing the Flash Spaces

User

The following procedure is used to program the User space and is summarized in Figure 14:

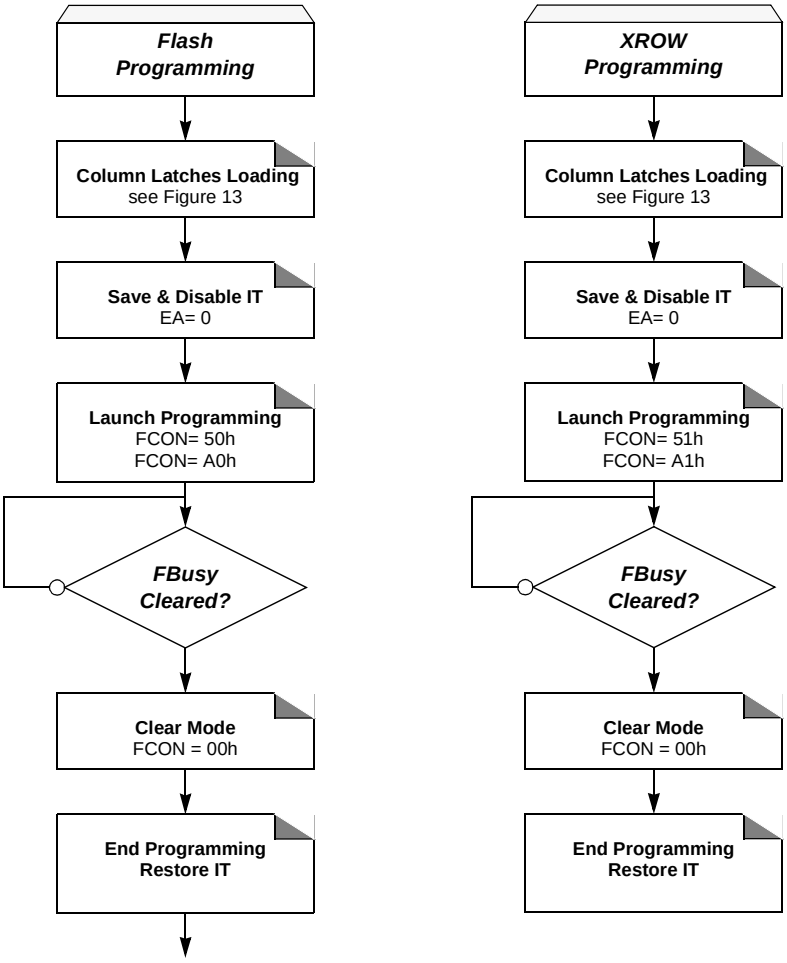
- Load up to one page of data in the column latches from address 0000h to FFFFh (see Figure 13.).
- Disable the interrupts.
- Launch the programming by writing the data sequence 50h followed by A0h in FCON register.
  The end of the programming indicated by the FBUSY flag cleared.
- Enable the interrupts.

Extra Row

The following procedure is used to program the Extra Row space and is summarized in Figure 14:

- Load data in the column latches from address FF80h to FFFFh.
- Disable the interrupts.
- Launch the programming by writing the data sequence 51h followed by A1h in FCON register.
  The end of the programming indicated by the FBUSY flag cleared.
- Enable the interrupts.

Figure 14. Flash and Extra row Programming Procedure



Hardware Security Byte (HSB)

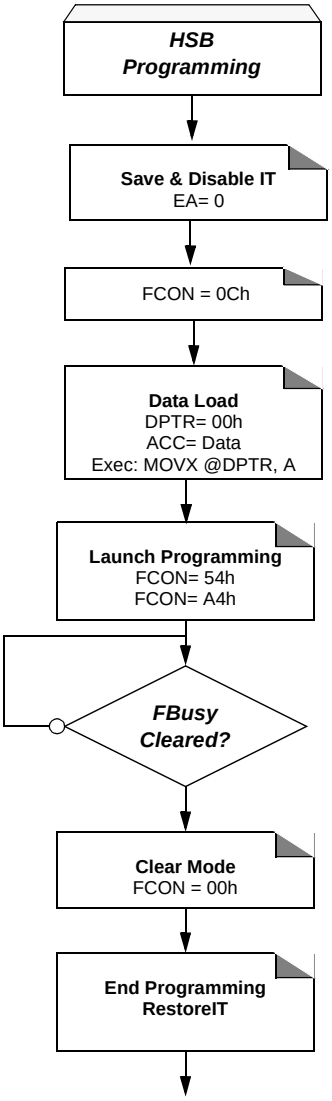
The following procedure is used to program the Hardware Security Byte space and is summarized in Figure 15:

- 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.
  The end of the programming indicated by the FBusy flag cleared.
- Restore the interrupts

.

Figure 15. Hardware Security Byte Programming Procedure



Fuse Configuration Byte (FCB)

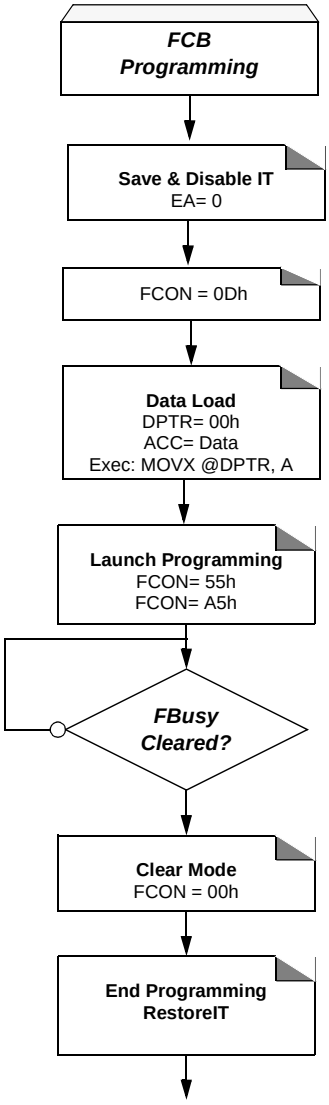
The following procedure is used to program the Fuse Configuration Byte space and is summarized in Figure 16:

- Set FPS and map FCB (FCON = 0x0D)
- 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 55h followed by A5h in FCON register.
  The end of the programming indicated by the FBusy flag cleared.
- Restore the interrupts

.

Figure 16. Fuse Configuration Byte Programming Procedure



Reset of columns latches space

No automatic reset of the columns latches is performed after a successful flash write process. Resetting the columns latches during a flash write process is mandatory. User shall implement a reset of the column latch before each column latch load sequence.

In addition, the user application can reset the columns latches space manually. The following procedure is used to reset the columns latches space

Launch the programming by writing the data sequence 53h followed by A3h in FCON register (from FM0 and RM0).

Errors Report / Miscellaneous states

Flash Busy flag

The FBUSY flag indicates on-going flash write operation.

The busy flag is set by hardware, the hardware clears this flag after the end of the programming operation.

Flash Programming Sequence Error

When a wrong sequence is detected the FSE in FSTA is set.

The following events are considered as not correct activation sequence:

- The two “MOV FCON,5x and MOV FCON, Ax” were not consecutive, or the second instruction differs from “MOV FCON Ax” (for example, an interrupt occurs during the sequence).

- The sequence (write flash or reset column latches) occurred with no data loaded in the column latches

The FSE bit can be cleared:

- By software

- By hardware when a correct programming sequence occurs.

Note: When a good sequence occurs just after an incorrect sequence, the previous error is lost. The user software application should take care to check the FSE bit before initiating a new sequence.

Power Down Mode Request

In Power Down mode, the on-chip flash memory is deselected (to reduce power consumption), this leads to the lost of the columns latches content.

In this case, if columns latches were previously loaded they are reset: FLOAD bit in FSTA register should be reset after power down mode.

If a power down mode is requested during flash programming (FBUSY=1), all power down sequence instructions should be ignored until the end of flash process.

Reading the Flash Spaces

User

The following procedure is used to read the User space:

- Read one byte in Accumulator by executing MOVC A,@A+DPTR

Note: FCON is supposed to be reset when not needed.

Depending of the MBO2:0 bits, the MOVC A,@A+DPTR can address a specific upper 32K bytes bank. It allows to read the 32K bytes upper On-chip flash memory from one bank to another one.

When read from the bootloader area, the user memory shall be mapped before any read access by setting the FMR bit of the FSTA register.

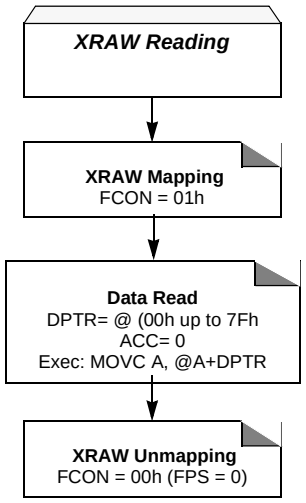
By default, when the bootloader is entered by hardware conditions, the ROM area is mapped for MOVC A,@A+DPTR operations. It is necessary to remap the user memory before each read access.

Extra Row (XAF)

The following procedure is used to read the Extra Row space and is summarized in Figure 17:

- Map the Extra Row space by writing 01h in FCON register.
- Read one byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= 0000h to 007Fh.
- Clear FCON to unmap the Extra Row.

Figure 17. XAF Reading Procedure

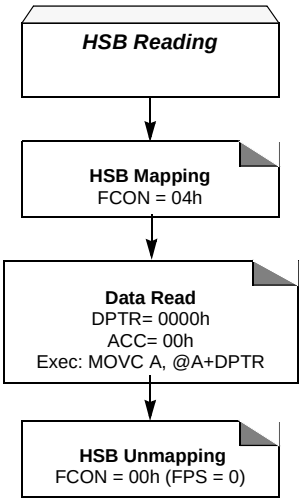


Hardware Security Byte

The following procedure is used to read the Hardware Security space and is summarized in Figure 18:

- Map the Hardware Security space by writing 04h in FCON register.
- Read the byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= 0000h.
- Clear FCON to unmap the Hardware Security Byte.

Figure 18. HSB Reading Procedure

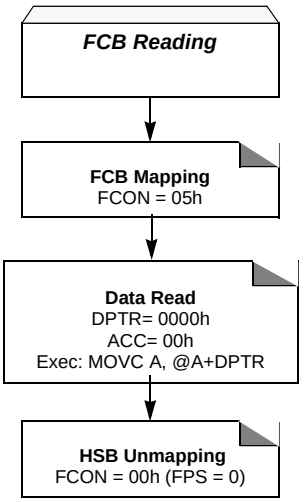


Fuse ConfigurationByte

The following procedure is used to read the Fuse Configuration byte and is summarized in Figure 18:

- Map the FCB by writing 05h in FCON register.
- Read the byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= 0000h.
- Clear FCON to unmap the Hardware Security Byte.

HSB Reading Procedure



Operation Cross Memory Access

Space addressable in read and write are:

- RAM
- ERAM (Expanded RAM access by movx)
- XRAM (eXternal RAM)
- FM0 (user flash)
- Hardware byte
- XROW FM0
- Boot RM0
- Flash Column latch

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

Table 27. Cross Memory Access

	Action	RAM	XRAM ERAM	boot RM0	FM0	HSB	FCB	XAF FM0
boot RM0	Read	ok	ok	ok	ok	ok	ok	ok
	Write	ok	ok	-	ok (RWW)	ok (RWW)	ok (RWW)	ok (RWW)
FM0	Read	ok	ok	-	ok	ok	ok	ok
	Write	ok	ok	-	ok (idle)	ok	ok	ok
External memory EA = 0 or BANK3	Read	ok	ok	-	-	-	-	-
	Write	ok	ok	-	-	-	-	-

Sharing Instructions

Table 28. Instructions shared

Action	RAM	XRAM	RM0	CL FM0	FM0	HSB	XAF FM0
Read	MOV	MOVX A,@DPTR	MOVC A, @A+DPTR	-	MOVC A, @A+DPTR	MOVC A, @A+DPTR	MOVC A, @A+DPTR
Write	MOV	MOVX @DPTR,A	-	MOVX @DPTR,A	by CL FM0	by CL FM0	by CL FM0

Note: by cl: using Column Latch

Table 29. Write MOVX @DPTR,A

FPS of FCCON	EA	XRAM ERAM	CL FM0
0	X	winner	
1	1		winner
	0	winner	

Table 30. MOVC A, @A+DPTR executed from External code EA=0

FMOD2:0	FBS (Fetch)	MBO (Target)	MOVC A,@A+DPTR
X	X	X	Read External Code

Table 31. MOVC A, @A+DPTR executed from External code EA=1, PC>=0x8000, FBS=Bank3

FMOD2:0	MBO (Target)	DPTR	MOVC A,@A+DPTR
X	X	< 0x8000	Depends on FLB2:0 Can Returns Random value, for secured part.
	X	>= 0x8000	External code read

Flash Protection from Parallel Programming

The three lock bits in Hardware Security Byte (see "In-System Programming" section) are programmed according to Table 32 provide different level of protection for the on-chip flash memory FM0.

They are set by default to level 4

Table 32. Program Lock Bit FLB2-0

Program Lock Bits				
Security level	FLB0	FLB1	FLB2	**Protection Description**
1	U	U	U	No program lock features enabled.
2	P	U	U	MOVC instruction executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further parallel programming of the Flash is disabled. ISP allows only flash verification (no write operations are allowed) but IAP from internal code still allowed.
3	U	P	U	Same as 2, also verify through parallel programming interface is disabled and ISP read operation not allowed.
4	U	U	P	Same as 3, also external execution is disabled (external bank not accessible)

Program Lock bits

U: unprogrammed

P: programmed

WARNING: Security level 2 and 3 should only be programmed after verification.

Bootloader Architecture

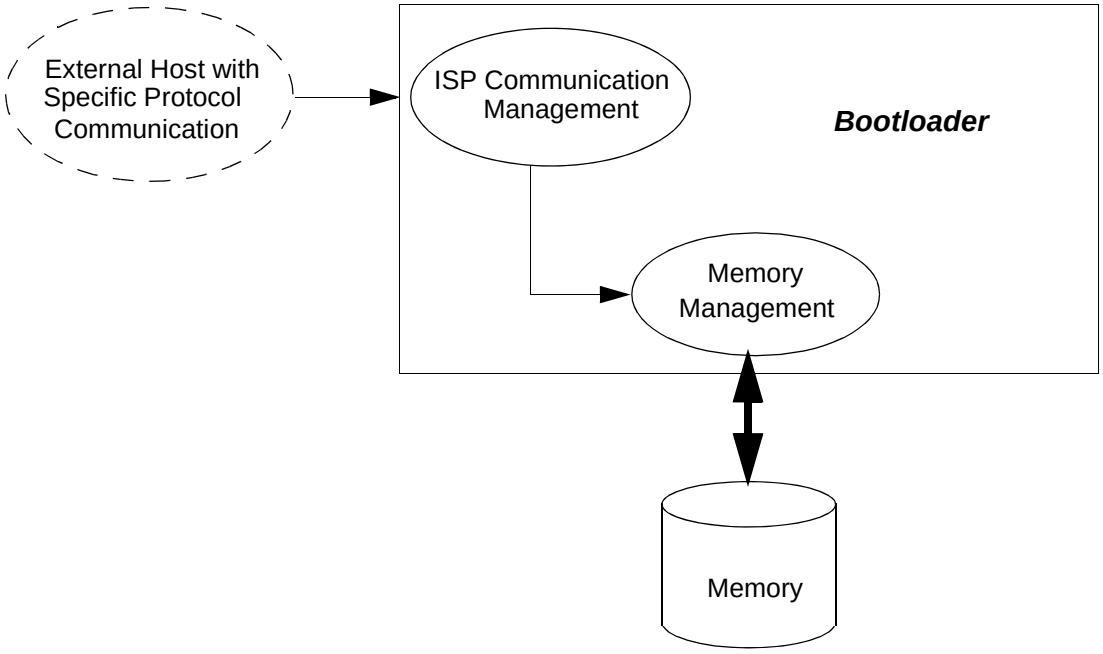
Introduction

The bootloader manages a communication between a host platform running an ISP tool and a AT89C51RE2 target.

The bootloader implemented in AT89C51RE2 is designed to reside in the dedicated ROM bank. This memory area can only be executed (fetched) when the processor enters the boot process.

The implementation of the bootloader is based on standard set of libraries including INTEL hex based protocol, standard communication links and ATMEL ISP command set.

Figure 19. Bootloader Functional Description



On the above diagram, the on-chip bootloader processes are:

- ISP Communication Management

The purpose of this process is to manage the communication and its protocol between the on-chip bootloader and a external device. The on-chip ROM implement a serial protocol (see section Bootloader Protocol). This process translate serial communication frame (UART) into Flash memory access (read, write, erase...).

- Memory Management

This process manages low level access to Flash memory (performs read and write access).

Bootloader Description

Entry points

After reset only one bootloader entry point is possible. This entry point stands at address 0x0000 of the boot ROM memory. This entry point executes the boot process of the bootloader.

The bootloader entry point can be selected through two processes:

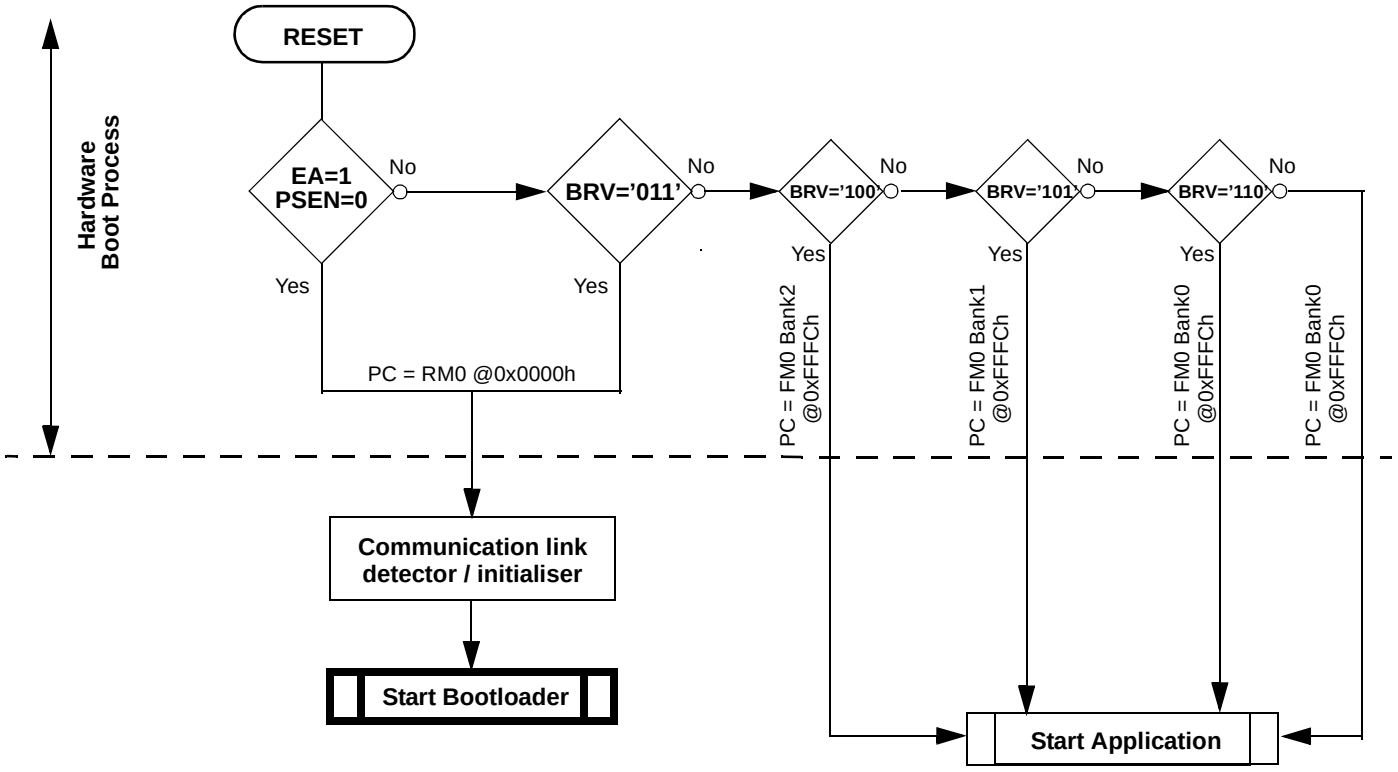
At reset, if the hardware conditions are applied, the bootloader entry point is accessed and executed.

At reset, if the hardware conditions are not set and the BRV2-0 is programmed ‘011’, the bootloader entry point is accessed and the bootprocess is started.

Boot Process Description

The boot process consists in three main operations:

- The hardware boot process request detection
- The communication link detection (Uart or OCD)
- The start-up of the bootloader



Hardware boot process request detection

The hardware boot process request is detected when the hardware conditions (under reset, EA=1 and PSEN=0) are received by the processor or when no hardware condition is applied and the BRV2:0 is configured ‘011’.

Communication link detection

Two interfaces are available for ISP:

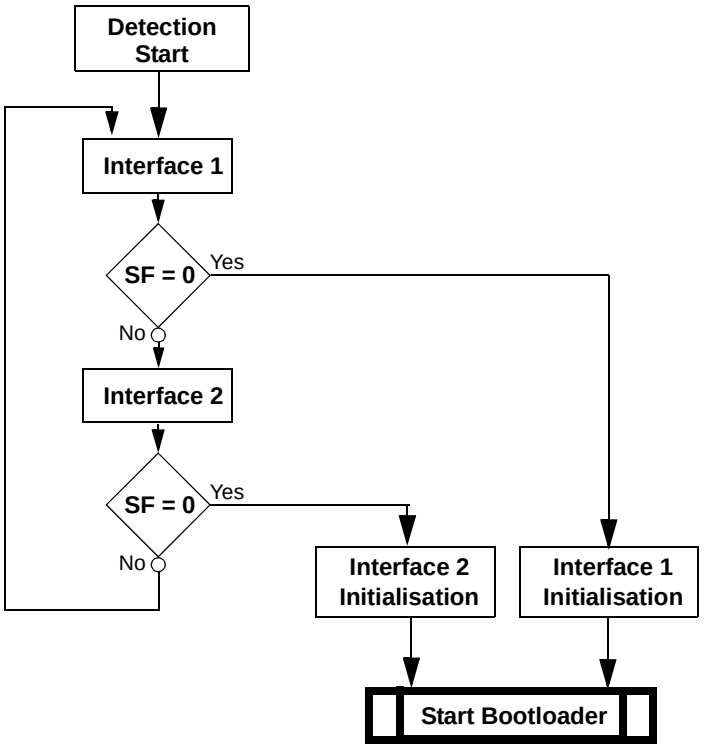
- UART0
- OCD UART

The communication link detection is done by a circular polling on all the interfaces. On AT89C51RE2, the ISP interfaces are all based on simple UART mechanisms (Rx, Tx).

The Rx line default state is ‘1’ when no communication is in progress. A transition from ‘1’ to ‘0’ on the Rx line indicates a start of frame.

Once one of the interface detects a starts of frame (‘0’) on its Rx line, the interface is selected and configuration of the communication link starts.

Figure 20. Communication link Detection



Notes:
1. SF: Start of Frame (‘0’ = detected; ‘1’ = not detected)
2. In AT89C51RE2 implementation, Interface 1 refers to UART0 and Interface 2 refers to the OCD UART interface.

ISP Protocol Description

Physical Layer

The UART used to transmit information has the following configuration:

- Character: 8-bit data
- Parity: none
- Stop: 1 bit
- Flow control: none
- Baud rate: autobaud is performed by the bootloader to compute the baud rate chosen by the host.

Frame Description

The Serial Protocol is based on the Intel Extended Hex-type records.

Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below.

Table 33. Intel Hex Type Frame

Record Mark ':'	Record length	Load Offset	Record Type	Data or Info	Checksum
1 byte	1 byte	2 bytes	1 bytes	n byte	1 byte

- Record Mark:
  - Record Mark is the start of frame. This field must contain':'.
- Record length:
  - Record length specifies the number of Bytes of information or data which follows the Record Type field of the record.
- Load Offset:
  - Load Offset specifies the 16-bit starting load offset of the data Bytes, therefore this field is used only for
  - Data Program Record.
- Record Type:
  - Record Type specifies the command type. This field is used to interpret the remaining information within the frame.
- Data/Info:
  - Data/Info is a variable length field. It consists of zero or more Bytes encoded as pairs of hexadecimal digits. The meaning of data depends on the Record Type.
- Checksum:
  - Checksum is the two's complement of the 8-bit Bytes that result from converting each pair of ASCII hexadecimal digits to one Byte of binary, thus including all field from the Record Length field to the last Byte of the Data/Info field. Therefore, the sum of all the ASCII pairs in a record after converting to binary, including all field from the Record Length field to the Checksum field, is zero.

Protocol

Overview

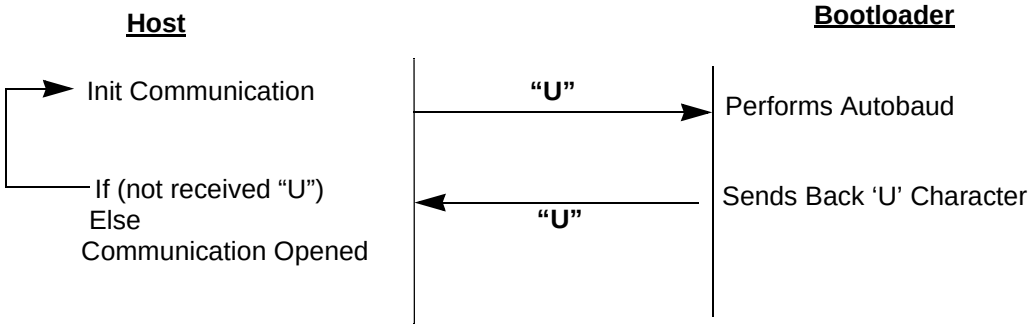
An initialization step must be performed after each Reset. After microcontroller reset, the bootloader waits for an autobaud sequence (see Section “Autobaud Performances”).

When the communication is initialized the protocol depends on the record type issued by the host.

Communication Initialization

The host initiates the communication by sending a ’U’ character to help the bootloader to compute the baudrate (autobaud).

Figure 21. Initialization



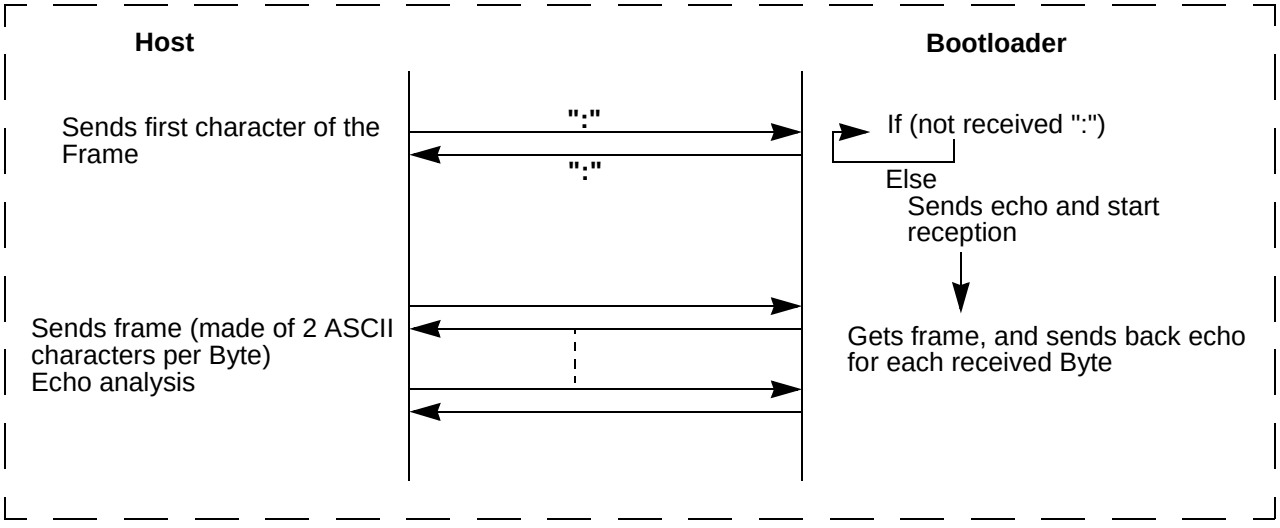
Autobaud Performances

The bootloader supports a wide range of baud rates. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts based on the oscillator frequency.

Command Data Stream Protocol

All commands are sent using the same flow. To increase performance, the echo has been removed from the bootloader response.

Figure 22. Command Flow



Each command flow may end with:

- “X”: If checksum error
- “L”: If read security is set
- “P”: If program security is set
- “.”: If command ok
- byte + “.”: read byte ok

Reading/Blank checking memory

To start the reading or blank checking operation,

Requests from Host

Command	Record Type	Record Length	Offset	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]
Read selected memory	04h	05h	0000h	Start Address		End Address		00h
Blank Check selected memory								01h

Answers from Bootloader

The boot loader can answer to a read command with:

- 'Address = data '& 'CR' &'LF' the number of data by line depends of the bootloader.
- 'X' & 'CR' & 'LF' if the checksum is wrong
- 'L' & 'CR' & 'LF' if the Security is set

The bootloader answers to blank check command:

- '.' & 'CR' &'LF' when the blank check is ok
- 'First Address wrong' 'CR' & 'LF' when the blank check is fail
- 'X' & 'CR' & 'LF' if the checksum is wrong
- 'L' & 'CR' & 'LF' if the Security is set

Changing memory/page

To change the memory selected and/or the page, the Host can send two commands.

- Select New Page to keep the same memory.
- Select Memory to change the Memory and page

Requests from Host

Command	Record Type	Record Length	Offset	Data[0]	Data[1]
Select New Page	02h	02h	start address	Page (4 bits) + 0h	00h
Select Memory	04h	02h	0000h	Memory space	Page

Answers from Bootloader

The boot loader can answer to a read command with:

- '. '& 'CR' &'LF' if the command is done
- 'X' & 'CR' & 'LF' if the checksum is wrong

Programming/Erasing memory

Requests from Host

Command	Record Type	Record Length	Offset	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]
Program selected memory	00h	nb of data	start address	x	x	x	x	x
Erase selected memory	04h	05h	0000h	00h	FFh	00h	00h	02h

Answers from Bootloader

The boot loader answers with:

- '.' & 'CR' &'LF' when the data are programmed
- 'X' & 'CR' & 'LF' if the checksum is wrong
- 'P' & 'CR' & 'LF' if the Security is set

Starting application

The application can only be started by a Watchdog reset.

No answer is returned by the bootloader.

Requests from Host

Command	Record Type	Record Length	Offset
Start application with watchdog	01h	00h	0000h

ISP Commands description

Select Memory Space

The '***Select Memory Space***' command allows to route all read, write commands to a selected area. For each area (Family) a code is defined. This code corresponds to the memory area encoded value in the INTEL HEX frame.

The area supported and there coding are listed in the table below.

Table 34. Memory Families & coding

Memory/Information Family	coding*	name
FLASH	0	MEM_FLASH
SECURITY	7	MEM_PROTECT
CONFIGURATION	8	MEM_CONF
BOOTLOADER	3	MEM_BOOT
SIGNATURE	6	MEM_SIGNATURE

The Bootloader information and the signature areas are read only. The value in the coding column is the value to report in the corresponding protocol field.

Note: * the coding number doesn't include any information on the authorized address range of the family. A summary of these addresses is available in appendix (See "Address Mapping" on page 67.)

Select Page

The '***Select Page***' command allows to define a page number in the selected area. A page is defined as a 64K linear memory space (According to the INTEL HEX format). It doesn't corresponds to a physical bank from the processor.

The following table summarizes the memory spaces for which the select page command can be applied.

Table 35. Memory space & Select page

Memory/Information Family	Comments/Restriction
FLASH	page 0 (0->64K) and 1(64k->128k) available

Write commands

The following table summarizes the memory spaces for which the write command can be applied.

Table 36. Memory space & Select page

Memory/Information Family	Comments/Restriction
FLASH	need security level check
SECURITY	only a higher level can be write
CONFIGURATION	

In case of write command to other area, nothing is done.

The bootloader returns a Write protection ('P') if the SECURITY do not allow any write operation from the bootloader.

FLASH

The program/data Flash memory area can be programmed by the bootloader by data pages of up to 128bytes.

If the Flash memory security level is at least '2' (FLB2:0 = '110'), no write operation can be performed through the bootloader.

Table 37. Flash Write Authorization Summary

Command	Security level (HSB)			
	FLB2:0			
	111	110	101	011
Write	Allowed	Forbidden	Forbidden	Forbidden

CONFIGURATION

The FCB configuration byte can always be written, whatever are the security levels.

SECURITY

The Security byte can always be written with a value that enables a protection higher than the previous one.

If attempting to write a lower security, no action is performed and the bootloader returns a protection error code ('P')

Table 38. Security Write Authorization Summary

write from FLB2:0	Security level (HSB)			
	to FLB2:0			
	111	110	101	011
111	Allowed	Allowed	Allowed	Allowed
110	Forbidden	Allowed	Allowed	Allowed
101	Forbidden	Forbidden	Allowed	Allowed
011	Forbidden	Forbidden	Forbidden	Allowed

Erasing commands

The erasing command is supported by the following areas:

Table 39. Memory space & Erase

Memory/Information Family	Comments/Restriction
FLASH	need security level check

Nothing is done on the other areas.

FLASH

The erasing command on the Flash memory:

- erases the four physical flash memory banks (from address 0000h to 1FFFFh).
- the HSB (Hardware Security Byte) is set at NO_PROTECTION:
  - FLB2.0 = ‘111’

Blank Checking commands

The blank checking command is supported by the following areas

Table 40. Memory space & Erase

Memory/Information Family	Comments/Restriction
FLASH	need security level check

Nothing is done on the other areas.

The first not erased address is returned if the blank check is failed.

FLASH

The blank checking command on the Flash memory can be done from address 0000h to 1FFFFh.

The blank check operation is only possible if the HSB (Hardware Security Byte) has a security level lower than or equal to ‘2’ (FLB2.0 = ‘110’)

Table 41. Flash Blank check Authorization Summary

Command	Security level (HSB)			
	FLB2:0			
	111	110	101	011
Blank Check	Allowed	Allowed	Forbidden	Forbidden

Reading commands

The reading command is supported by the following areas:

Table 42. Memory space & Select page

Memory/Information Family	Comments/Restriction
FLASH	need security level check
SECURITY	
CONFIGURATION	
BOOTLOADER	
SIGNATURE	

FLASH

The reading command on the Flash memory can be done from address 000h to 1FFFFh. The read operation is only possible if the HSB (Hardware Security Byte) has a security level lower than or equal to ‘2’ (FLB2.0 = ‘110’)

Table 43. Flash Read Authorization Summary

Command	Security level (HSB)			
	FLB2:0			
	111	110	101	011
Read	Allowed	Allowed	Forbidden	Forbidden

CONFIGURATION

The CONFIGURATION family can always be read.

SECURITY

The SECURITY family can always be read.

BOOTLOADER

All the field from the BOOTLOARED family can be read from the bootloader. Each bootloader information shall be read unitary. Accesses must be done byte per byte according to the address definition

SIGNATURE

All the field from the SIGNATURE family can be read from the bootloader. Each signature information shall be read unitary. Accesses must be done byte per byte according to the address definition

Start Application

The start application command is used to quit the bootloader and start the application loaded.

The start application is performed by a watchdog reset.

The best way to start the application from a user defined entry point is to configure the FCB (Fuse Configuration Byte) before launching the watchdog. Then, depending on the configuration of the BRV2:0 field, the hardware boots from the selected memory area.

ISP Command summary

UART Protocol frames

Table 44. Summary of frames from Host

Command	Record Type	Record Length	Offset	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]
Program selected memory	00h	nb of data	start address	x	x	x	x	x
Start application with watchdog	01h	00h	0000h	x	x	x	x	x
Select New Page	02h	02h	start address	Page (4 bits) + 0h	00h	x	x	x
Select Memory	04h	02h	0000h	Memory space	Page	x	x	x
Read selected memory		05h	0000h	Start Address		End Address		00h
Blank Check selected memory								01h
Erase Selected memory				00h	FFh	00h	00h	02h

Address Mapping

Table 45. Memory Families, Addresses & Coding

Memory/Parameter	coding	Address	Page number	Memory/Information Family
FLASH	0	0 up to 0x1FFFF	0 up to 1	FLASH
HSB	7	0	0	SECURITY
FCB	8	0	0	CONFIGURATION
Bootloader revision	3	00h	0	BOOTLOADER
Boot id1		01h		
Boot id2		02h		
Manuf. code	6	30h	0	SIGNATURE
Family code		31h		
Product name		60h		
Product rev		61h		

Attempting an access with any other ‘coding’, ‘page number’ or ‘Address’ results in no action and no answer from the bootloader.

Timers/Counters

The AT89C51RE2 implements two general-purpose, 16-bit Timers/Counters. Such are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a Timer or an event Counter. When operating as a Timer, the Timer/Counter runs for a programmed length of time, then issues an interrupt request. When operating as a Counter, the Timer/Counter counts negative transitions on an external pin. After a preset number of counts, the Counter issues an interrupt request.
The various operating modes of each Timer/Counter are described in the following sections.

Timer/Counter Operations

A basic operation is Timer registers THx and TLx (x = 0, 1) connected in cascade to form a 16-bit Timer. Setting the run control bit (TRx) in TCON register (see Figure 46) turns the Timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the Timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx Timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the Timer/Counter is unpredictable.

The C/Tx# control bit selects Timer operation or Counter operation by selecting the divided-down peripheral clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation, otherwise the behavior of the Timer/Counter is unpredictable.

For Timer operation (C/Tx# = 0), the Timer register counts the divided-down peripheral clock. The Timer register is incremented once every peripheral cycle (6 peripheral clock periods). The Timer clock rate is $F_{PER}/6$, i.e. $F_{OSC}/12$ in standard mode or $F_{OSC}/6$ in X2 mode.

For Counter operation (C/Tx# = 1), the Timer register counts the negative transitions on the Tx external input pin. The external input is sampled every peripheral cycles. When the sample is high in one cycle and low in the next one, the Counter is incremented. Since it takes 2 cycles (12 peripheral clock periods) to recognize a negative transition, the maximum count rate is $F_{PER}/12$, i.e. $F_{OSC}/24$ in standard mode or $F_{OSC}/12$ in X2 mode. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full peripheral cycle.

Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 23 to Figure 26 show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of TMOD register (see Figure 47) and bits 0, 1, 4 and 5 of TCON register (see Figure 46). TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (T/C0#) and mode of operation (M10 and M00). TCON register provides Timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).

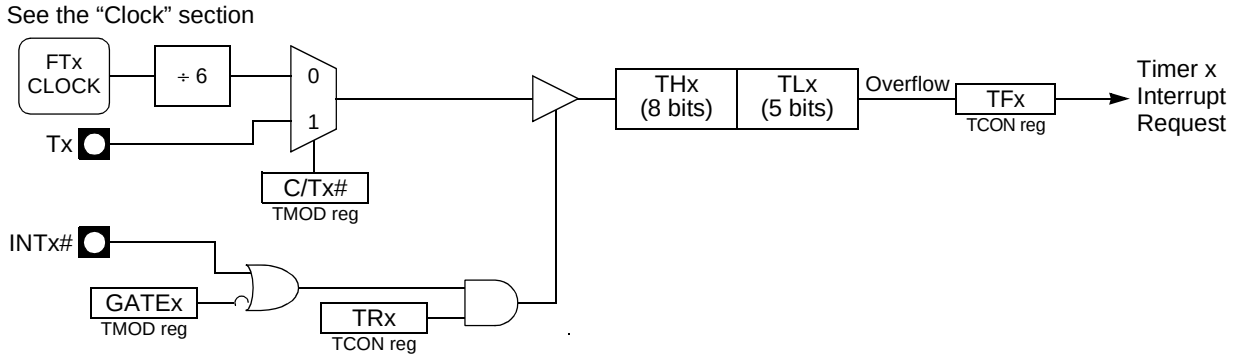
For normal Timer operation (GATE0 = 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control Timer operation.

Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag generating an interrupt request.

It is important to stop Timer/Counter before changing mode.

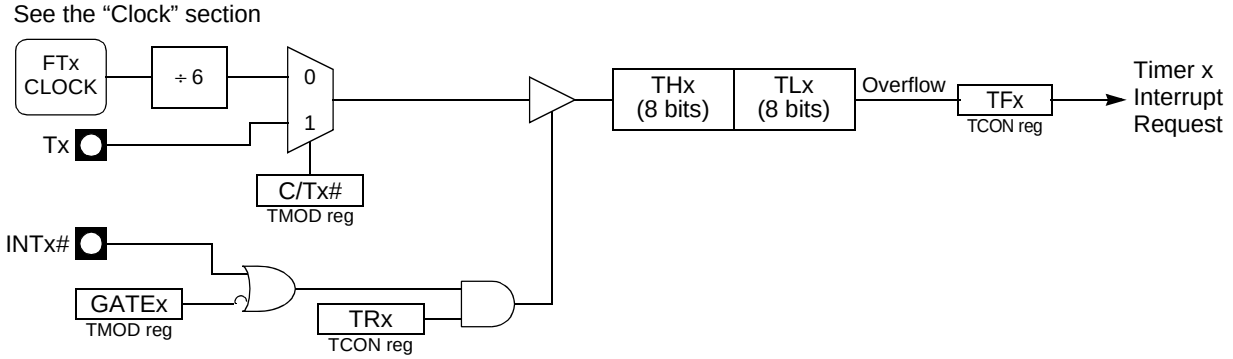
Mode 0 (13-bit Timer) Mode 0 configures Timer 0 as an 13-bit Timer which is set up as an 8-bit Timer (TH0 register) with a modulo 32 prescaler implemented with the lower five bits of TL0 register (see Figure 23). The upper three bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments TH0 register.

Figure 23. Timer/Counter x (x = 0 or 1) in Mode 0



Mode 1 (16-bit Timer) Mode 1 configures Timer 0 as a 16-bit Timer with TH0 and TL0 registers connected in cascade (see Figure 24). The selected input increments TL0 register.

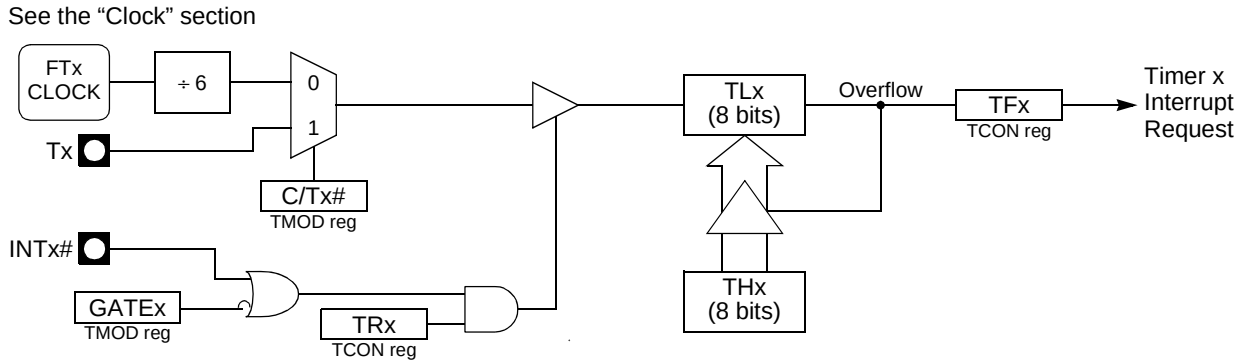
Figure 24. Timer/Counter x (x = 0 or 1) in Mode 1



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

Mode 2 configures Timer 0 as an 8-bit Timer (TL0 register) that automatically reloads from TH0 register (see Figure 25). TL0 overflow sets TF0 flag in TCON register and reloads TL0 with the contents of TH0, which is preset by software. When the interrupt request is serviced, hardware clears TF0. The reload leaves TH0 unchanged. The next reload value may be changed at any time by writing it to TH0 register.

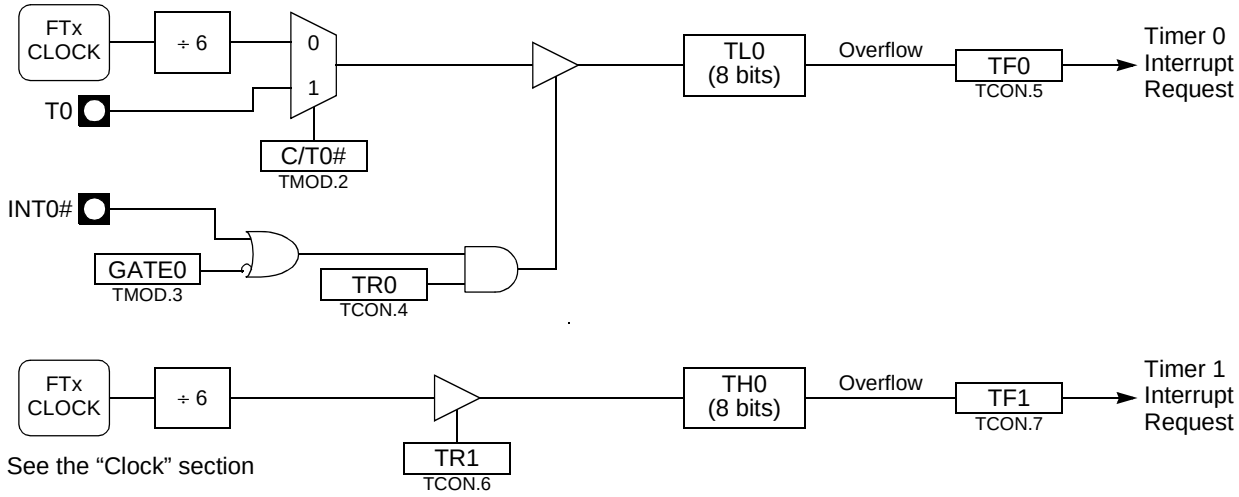
Figure 25. Timer/Counter x (x = 0 or 1) in Mode 2



Mode 3 (Two 8-bit Timers)

Mode 3 configures Timer 0 such that registers TL0 and TH0 operate as separate 8-bit Timers (see Figure 26). This mode is provided for applications requiring an additional 8-bit Timer or Counter. TL0 uses the Timer 0 control bits C/T0# and GATE0 in TMOD register, and TR0 and TF0 in TCON register in the normal manner. TH0 is locked into a Timer function (counting F_{PER} /6) and takes over use of the Timer 1 interrupt (TF1) and run control (TR1) bits. Thus, operation of Timer 1 is restricted when Timer 0 is in mode 3.

Figure 26. Timer/Counter 0 in Mode 3: Two 8-bit Counters



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 23 to Figure 25 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 47) and bits 2, 3, 6 and 7 of TCON register (see Figure 46). 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 23). 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 24). 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 25). 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.

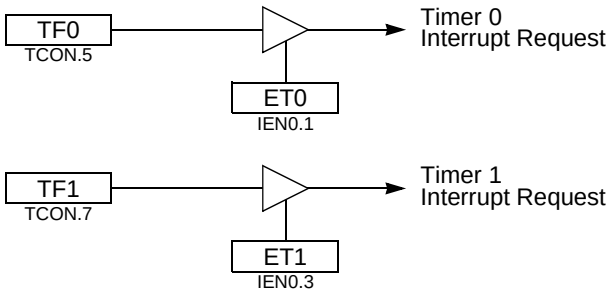
Mode 3 (Halt)

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.

Interrupt

Each Timer handles one interrupt source that is the timer overflow flag TF0 or TF1. This flag is set every time an overflow occurs. Flags are cleared when vectoring to the Timer interrupt routine. Interrupts are enabled by setting ETx bit in IEN0 register. This assumes interrupts are globally enabled by setting EA bit in IEN0 register.

Figure 27. Timer Interrupt System



Registers

Table 46. TCON Register

TCON (S:88h)
Timer/Counter Control Register

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

Bit Number	Bit Mnemonic	Description
7	TF1	**Timer 1 Overflow Flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 1 register overflows.
6	TR1	**Timer 1 Run Control Bit** Clear to turn off Timer/Counter 1. Set to turn on Timer/Counter 1.
5	TF0	**Timer 0 Overflow Flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 0 register overflows.
4	TR0	**Timer 0 Run Control Bit** Clear to turn off Timer/Counter 0. Set to turn on Timer/Counter 0.
3	IE1	**Interrupt 1 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.
2	IT1	**Interrupt 1 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	**Interrupt 0 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	**Interrupt 0 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.

Reset Value = 0000 0000b

Table 47. TMOD Register

TMOD (S:89h)
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	**C/T1#**	**M11**	**M01**	**GATE0**	**C/T0#**	**M10**	**M00**

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control Bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select Bit** Clear for Timer operation: Timer 1 counts the divided-down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select Bits** M11 M01 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL1)(1) 1 1 Mode 3: Timer 1 halted. Retains count
4	M01	
3	GATE0	**Timer 0 Gating Control Bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select Bit** Clear for Timer operation: Timer 0 counts the divided-down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select Bit** M10 M00 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH0) with 5-bit prescaler (TL0). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL0)(2) 1 1 Mode 3: TL0 is an 8-bit Timer/Counter TH0 is an 8-bit Timer using Timer 1's TR0 and TF0 bits.
0	M00	

Notes: 1. Reloaded from TH1 at overflow.
2. Reloaded from TH0 at overflow.

Reset Value = 0000 0000b

Table 48. TH0 Register

TH0 (S:8Ch)
Timer 0 High Byte Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 0.**

Reset Value = 0000 0000b

Table 49. TL0 Register

TL0 (S:8Ah)
Timer 0 Low Byte Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 0.**

Reset Value = 0000 0000b

Table 50. TH1 Register

TH1 (S:8Dh)
Timer 1 High Byte Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 1.**

Reset Value = 0000 0000b

Table 51. TL1 Register

TL1 (S:8Bh)
Timer 1 Low Byte Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 1.**

Reset Value = 0000 0000b

Timer 2

The Timer 2 in the AT89C51RE2 is the standard C52 Timer 2.
It is a 16-bit timer/counter: the count is maintained by two eight-bit timer registers, TH2 and TL2 are cascaded. It is controlled by T2CON (Table 52) and T2MOD (Table 53) registers. Timer 2 operation is similar to Timer 0 and Timer 1.C/$\overline{T2}$ selects $F_{OSC}/12$ (timer operation) or external pin T2 (counter operation) as the timer clock input. Setting TR2 allows TL2 to increment by the selected input.

Timer 2 has 3 operating modes: capture, autoreload and Baud Rate Generator. These modes are selected by the combination of RCLK, TCLK and CP/$\overline{RL2}$ (T2CON).

Refer to the Atmel 8-bit Microcontroller Hardware description for the description of Capture and Baud Rate Generator Modes.

Timer 2 includes the following enhancements:

- Auto-reload mode with up or down counter
- Programmable clock-output

Auto-Reload Mode

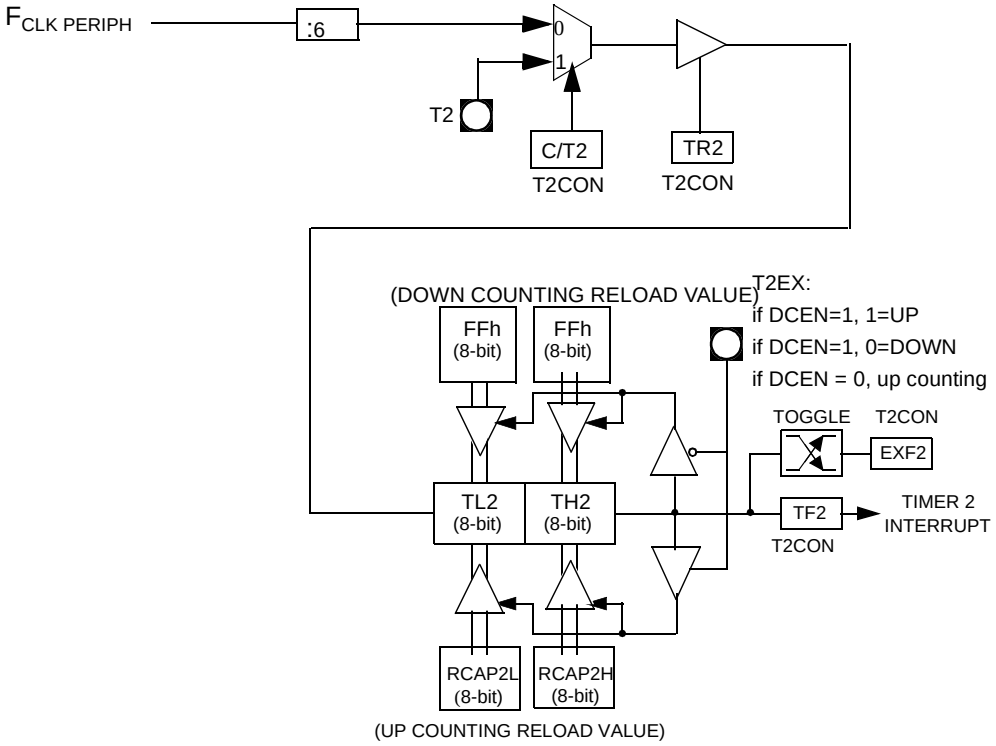
The auto-reload mode configures Timer 2 as a 16-bit timer or event counter with automatic reload. If DCEN bit in T2MOD is cleared, Timer 2 behaves as in 80C52 (refer to the Atmel C51 Microcontroller Hardware description). If DCEN bit is set, Timer 2 acts as an Up/down timer/counter as shown in Figure 28. In this mode the T2EX pin controls the direction of count.

When T2EX is high, Timer 2 counts up. Timer overflow occurs at FFFFh which sets the TF2 flag and generates an interrupt request. The overflow also causes the 16-bit value in RCAP2H and RCAP2L registers to be loaded into the timer registers TH2 and TL2.

When T2EX is low, Timer 2 counts down. Timer underflow occurs when the count in the timer registers TH2 and TL2 equals the value stored in RCAP2H and RCAP2L registers. The underflow sets TF2 flag and reloads FFFFh into the timer registers.

The EXF2 bit toggles when Timer 2 overflows or underflows according to the direction of the count. EXF2 does not generate any interrupt. This bit can be used to provide 17-bit resolution.

Figure 28. Auto-Reload Mode Up/Down Counter (DCEN = 1)



Programmable Clock-Output

In the clock-out mode, Timer 2 operates as a 50%-duty-cycle, programmable clock generator (See Figure 29). The input clock increments TL2 at frequency $F_{CLK\ PERIPH}/2$.The timer repeatedly counts to overflow from a loaded value. At overflow, the contents of RCAP2H and RCAP2L registers are loaded into TH2 and TL2.In this mode, Timer 2 overflows do not generate interrupts. The formula gives the clock-out frequency as a function of the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

$$Clock\text{–}OutFrequency = \frac{F_{CLKPERIPH}}{4 \times (65536 - RCAP2H/RCAP2L)}$$

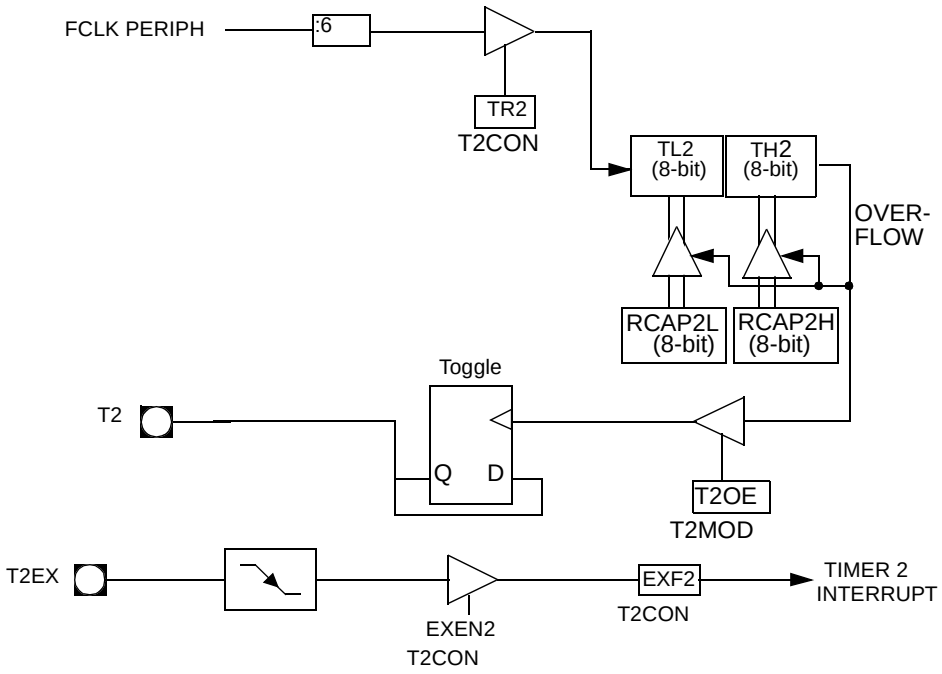
For a 16 MHz system clock, Timer 2 has a programmable frequency range of 61 Hz ($F_{CLK\ PERIPH}/2^{16}$) to 4 MHz ($F_{CLK\ PERIPH}/4$). The generated clock signal is brought out to T2 pin (P1.0).

Timer 2 is programmed for the clock-out mode as follows:

- Set T2OE bit in T2MOD register.
- Clear C/$\overline{T2}$ bit in T2CON register.
- Determine the 16-bit reload value from the formula and enter it in RCAP2H/RCAP2L registers.
- Enter a 16-bit initial value in timer registers TH2/TL2.It can be the same as the reload value or a different one depending on the application.
- To start the timer, set TR2 run control bit in T2CON register.

It is possible to use Timer 2 as a baud rate generator and a clock generator simultaneously. For this configuration, the baud rates and clock frequencies are not independent since both functions use the values in the RCAP2H and RCAP2L registers.

Figure 29. Clock-Out Mode C/$\overline{T2}$ = 0



Registers

Table 52. T2CON Register

T2CON - Timer 2 Control Register (C8h)

7	6	5	4	3	2	1	0
TF2	**EXF2**	**RCLK**	**TCLK**	**EXEN2**	**TR2**	**C/T2#**	**CP/RL2#**

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 overflow Flag** Must be cleared by software. Set by hardware on Timer 2 overflow, if RCLK = 0 and TCLK = 0.
6	EXF2	**Timer 2 External Flag** Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. When set, causes the CPU to vector to Timer 2 interrupt routine when Timer 2 interrupt is enabled. Must be cleared by software. EXF2 doesn't cause an interrupt in Up/down counter mode (DCEN = 1).
5	RCLK	**Receive Clock bit** Cleared to use timer 1 overflow as receive clock for serial port in mode 1 or 3. Set to use Timer 2 overflow as receive clock for serial port in mode 1 or 3.
4	TCLK	**Transmit Clock bit** Cleared to use timer 1 overflow as transmit clock for serial port in mode 1 or 3. Set to use Timer 2 overflow as transmit clock for serial port in mode 1 or 3.
3	EXEN2	**Timer 2 External Enable bit** Cleared to ignore events on T2EX pin for Timer 2 operation. Set to cause a capture or reload when a negative transition on T2EX pin is detected, if Timer 2 is not used to clock the serial port.
2	TR2	**Timer 2 Run control bit** Cleared to turn off Timer 2. Set to turn on Timer 2.
1	C/T2#	**Timer/Counter 2 select bit** Cleared for timer operation (input from internal clock system: $F_{CLK\ PERIPH}$). Set for counter operation (input from T2 input pin, falling edge trigger). Must be 0 for clock out mode.
0	CP/RL2#	**Timer 2 Capture/Reload bit** If RCLK=1 or TCLK=1, CP/RL2# is ignored and timer is forced to auto-reload on Timer 2 overflow. Cleared to auto-reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2=1. Set to capture on negative transitions on T2EX pin if EXEN2=1.

Reset Value = 0000 0000b
Bit addressable

Table 53. T2MOD Register

T2MOD - Timer 2 Mode Control Register (C9h)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	T2OE	DCEN

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	T2OE	**Timer 2 Output Enable bit** Cleared to program P1.0/T2 as clock input or I/O port. Set to program P1.0/T2 as clock output.
0	DCEN	**Down Counter Enable bit** Cleared to disable Timer 2 as up/down counter. Set to enable Timer 2 as up/down counter.

Reset Value = XXXX XX00b
Not bit addressable

Programmable Counter Array PCA

The PCA provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy. The PCA consists of a dedicated timer/counter which serves as the time base for an array of five compare/capture modules. Its clock input can be programmed to count any one of the following signals:

- Peripheral clock frequency ($F_{CLK\ PERIPH}$) ÷ 6
- Peripheral clock frequency ($F_{CLK\ PERIPH}$) ÷ 2
- Timer 0 overflow
- External input on ECI (P1.2)

Each compare/capture modules can be programmed in any one of the following modes:

- Rising and/or falling edge capture
- Software timer
- High-speed output
- Pulse width modulator

Module 4 can also be programmed as a watchdog timer (See Section "PCA Watchdog Timer", page 93).

When the compare/capture modules are programmed in the capture mode, software timer, or high speed output mode, an interrupt can be generated when the module executes its function. All five modules plus the PCA timer overflow share one interrupt vector.

The PCA timer/counter and compare/capture modules share Port 1 for external I/O. These pins are listed below. If the port is not used for the PCA, it can still be used for standard I/O.

PCA component	External I/O Pin
16-bit Counter	P1.2 / ECI
16-bit Module 0	P1.3 / CEX0
16-bit Module 1	P1.4 / CEX1
16-bit Module 2	P1.5 / CEX2
16-bit Module 3	P1.6 / CEX3

The PCA timer is a common time base for all five modules (See Figure 30). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD register (Table 54) and can be programmed to run at:

- 1/6 the peripheral clock frequency ($F_{CLK\ PERIPH}$)
- 1/2 the peripheral clock frequency ($F_{CLK\ PERIPH}$)
- The Timer 0 overflow
- The input on the ECI pin (P1.2)

Figure 30. PCA Timer/Counter

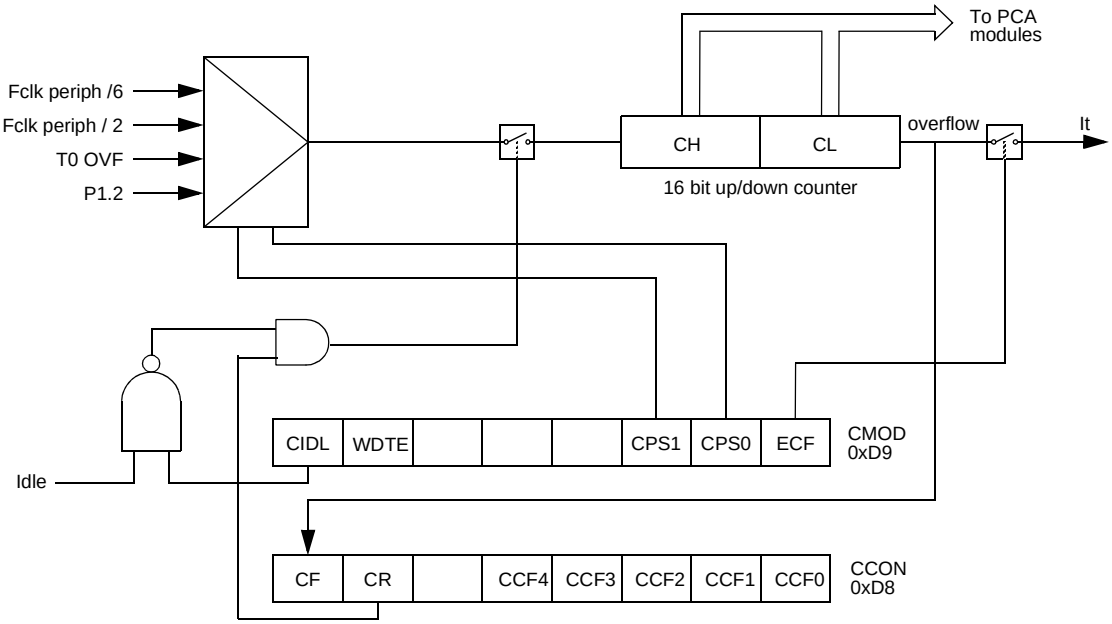


Table 54. CMOD Register

CMOD - PCA Counter Mode Register (D9h)

7	6	5	4	3	2	1	0
CIDL	WDTE	-	-	-	CPS1	CPS0	ECF

Bit Number	Bit Mnemonic	Description
7	CIDL	**Counter Idle Control** Cleared to program the PCA Counter to continue functioning during idle Mode. Set to program PCA to be gated off during idle.
6	WDTE	**Watchdog Timer Enable** Cleared to disable Watchdog Timer function on PCA Module 4. Set to enable Watchdog Timer function on PCA Module 4.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	CPS1	**PCA Count Pulse Select** CPS1 CPS0 Selected PCA input 0 0 Internal clock fCLK PERIPH/6 0 1 Internal clock fCLK PERIPH/2 1 0 Timer 0 Overflow 1 1 External clock at ECI/P1.2 pin (max rate = fCLK PERIPH/ 4)
1	CPS0	
0	ECF	**PCA Enable Counter Overflow Interrupt** Cleared to disable CF bit in CCON to inhibit an interrupt. Set to enable CF bit in CCON to generate an interrupt.

Reset Value = 00XX X000b
Not bit addressable

The CMOD register includes three additional bits associated with the PCA (See Figure 30 and Table 54).

- The CIDL bit which allows the PCA to stop during idle mode.
- The WDTE bit which enables or disables the watchdog function on module 4.
- The ECF bit which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows.

The CCON register contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (Refer to Table 55).

- Bit CR (CCON.6) must be set by software to run the PCA. The PCA is shut off by clearing this bit.
- Bit CF: The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software.
- Bits 0 through 4 are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software.

Table 55. CCON Register

CCON - PCA Counter Control Register (D8h)

7	6	5	4	3	2	1	0
CF	**CR**	**-**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**

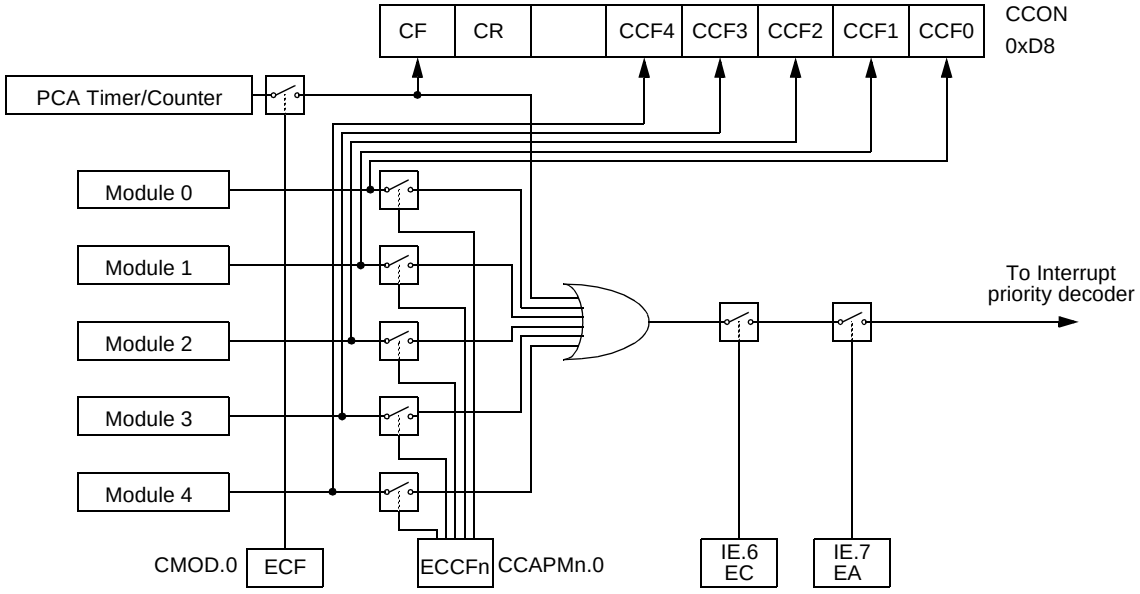
Bit Number	Bit Mnemonic	Description
7	CF	**PCA Counter Overflow flag** Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.
6	CR	**PCA Counter Run control bit** Must be cleared by software to turn the PCA counter off. Set by software to turn the PCA counter on.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	CCF4	**PCA Module 4 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
3	CCF3	**PCA Module 3 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
2	CCF2	**PCA Module 2 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
1	CCF1	**PCA Module 1 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
0	CCF0	**PCA Module 0 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.

Reset Value = 00X0 0000b
Not bit addressable

The watchdog timer function is implemented in module 4 (See Figure 33).

The PCA interrupt system is shown in Figure 31.

Figure 31. PCA Interrupt System



PCA Modules: each one of the five compare/capture modules has six possible functions. It can perform:

- 16-bit Capture, positive-edge triggered
- 16-bit Capture, negative-edge triggered
- 16-bit Capture, both positive and negative-edge triggered
- 16-bit Software Timer
- 16-bit High Speed Output
- 8-bit Pulse Width Modulator

In addition, module 4 can be used as a Watchdog Timer.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (See Table 56). The registers contain the bits that control the mode that each module will operate in.

- The ECCF bit (CCAPMn.0 where n=0, 1, 2, 3, or 4 depending on the module) enables the CCF flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module.
- PWM (CCAPMn.1) enables the pulse width modulation mode.
- The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register.
- The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.
- The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition.
- The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function.

Table 56 shows the CCAPMn settings for the various PCA functions.

Table 56. CCAPMn Registers (n = 0-4)

CCAPM0 - PCA Module 0 Compare/Capture Control Register (0DAh)

CCAPM1 - PCA Module 1 Compare/Capture Control Register (0DBh)

CCAPM2 - PCA Module 2 Compare/Capture Control Register (0DCh)

CCAPM3 - PCA Module 3 Compare/Capture Control Register (0DDh)

CCAPM4 - PCA Module 4 Compare/Capture Control Register (0DEh)

7	6	5	4	3	2	1	0
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	ECOMn	**Enable Comparator** Cleared to disable the comparator function. Set to enable the comparator function.
5	CAPPn	**Capture Positive** Cleared to disable positive edge capture. Set to enable positive edge capture.
4	CAPNn	**Capture Negative** Cleared to disable negative edge capture. Set to enable negative edge capture.
3	MATn	**Match** When MATn = 1, a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.
2	TOGn	**Toggle** When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.
1	PWMn	**Pulse Width Modulation Mode** Cleared to disable the CEXn pin to be used as a pulse width modulated output. Set to enable the CEXn pin to be used as a pulse width modulated output.
0	CCF0	**Enable CCF interrupt** Cleared to disable compare/capture flag CCFn in the CCON register to generate an interrupt. Set to enable compare/capture flag CCFn in the CCON register to generate an interrupt.

Reset Value = X000 0000b
Not bit addressable

Table 57. PCA Module Modes (CCAPMn Registers)

ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMm	ECCFn	Module Function
0	0	0	0	0	0	0	No Operation
X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
1	0	0	1	0	0	X	16-bit Software Timer / Compare mode.
1	0	0	1	1	0	X	16-bit High Speed Output
1	0	0	0	0	1	0	8-bit PWM
1	0	0	1	X	0	X	Watchdog Timer (module 4 only)

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output (See Table 58 & Table 59).

Table 58. CCAPnH Registers (n = 0-4)

CCAP0H - PCA Module 0 Compare/Capture Control Register High (0FAh)

CCAP1H - PCA Module 1 Compare/Capture Control Register High (0FBh)

CCAP2H - PCA Module 2 Compare/Capture Control Register High (0FCh)

CCAP3H - PCA Module 3 Compare/Capture Control Register High (0FDh)

CCAP4H - PCA Module 4 Compare/Capture Control Register High (0FEh)

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

Bit Number	Bit Mnemonic	Description
7-0	-	**PCA Module n Compare/Capture Control** CCAPnH Value

Reset Value = 0000 0000b
Not bit addressable

Table 59. CCAPnL Registers (n = 0-4)

CCAP0L - PCA Module 0 Compare/Capture Control Register Low (0EAh)

CCAP1L - PCA Module 1 Compare/Capture Control Register Low (0EBh)

CCAP2L - PCA Module 2 Compare/Capture Control Register Low (0ECh)

CCAP3L - PCA Module 3 Compare/Capture Control Register Low (0EDh)

CCAP4L - PCA Module 4 Compare/Capture Control Register Low (0EEh)

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

Bit Number	Bit Mnemonic	Description
7-0	-	**PCA Module n Compare/Capture Control** CCAPnL Value

Reset Value = 0000 0000b
Not bit addressable

Table 60. CH Register

CH - PCA Counter Register High (0F9h)

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

Bit Number	Bit Mnemonic	Description
7-0	-	**PCA counter** CH Value

Reset Value = 0000 0000b
Not bit addressable

Table 61. CL Register

CL - PCA Counter Register Low (0E9h)

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

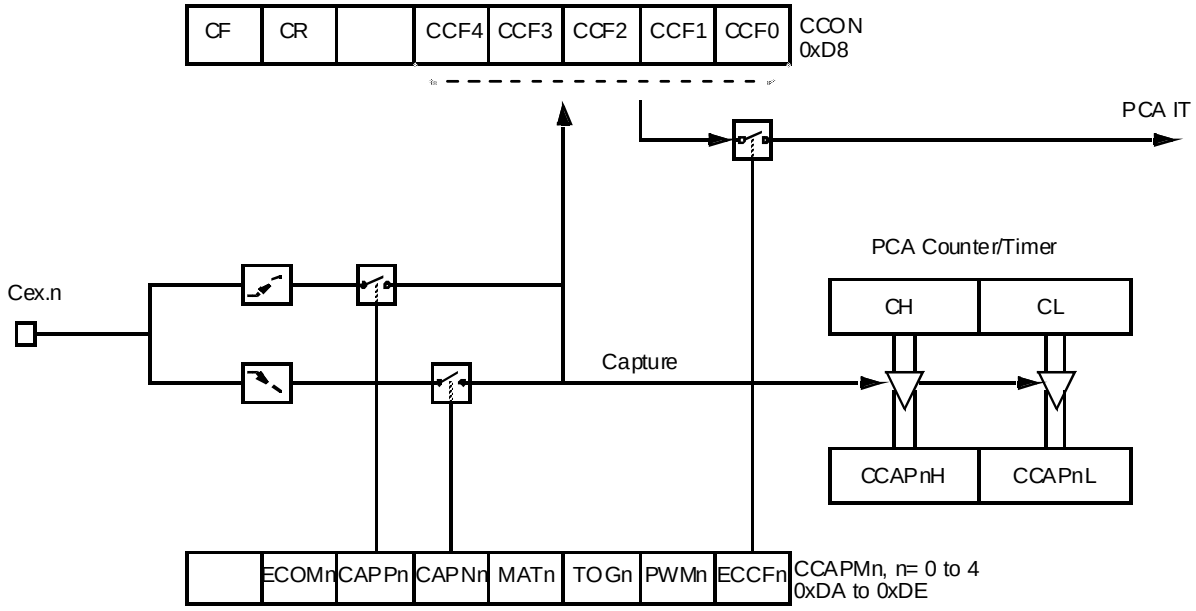
Bit Number	Bit Mnemonic	Description
7-0	-	**PCA Counter** CL Value

Reset Value = 0000 0000b
Not bit addressable

PCA Capture Mode

To use one of the PCA modules in the capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated (Refer to Figure 32).

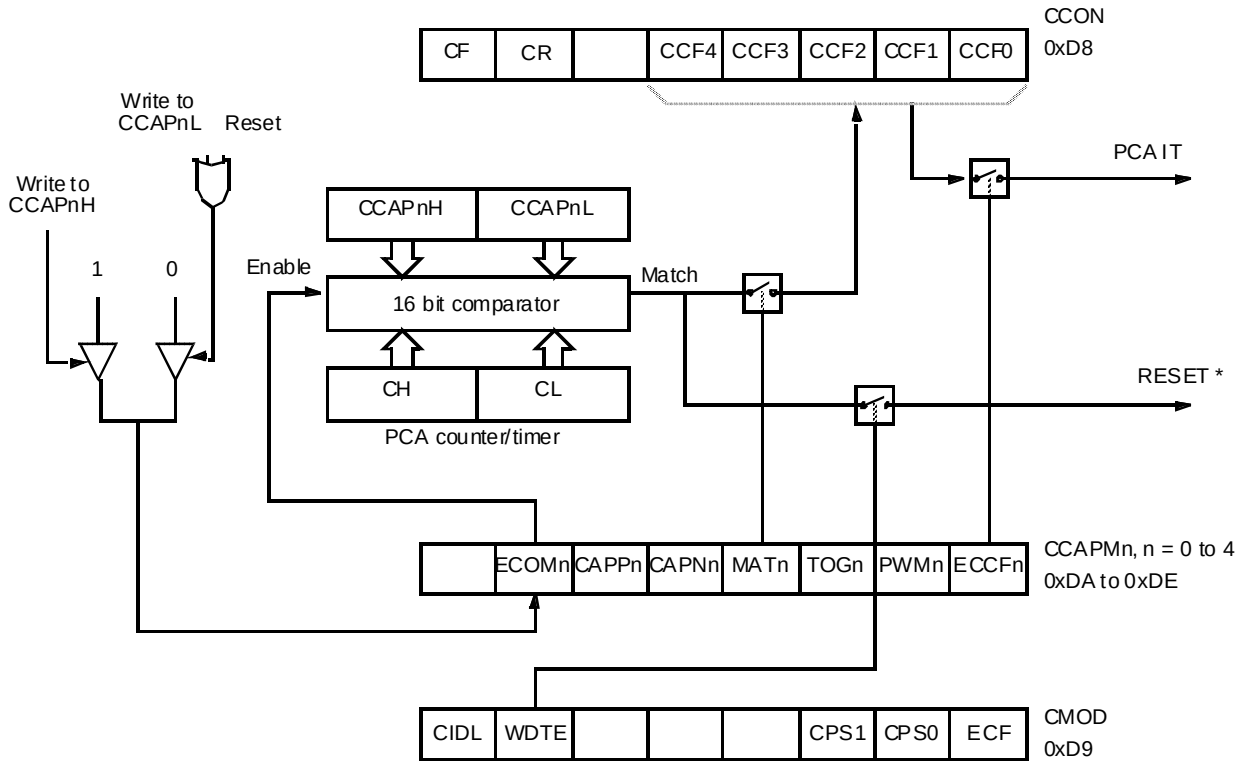
Figure 32. PCA Capture Mode



16-bit Software Timer/ Compare Mode

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set (See Figure 33).

Figure 33. PCA Compare Mode and PCA Watchdog Timer



Before enabling ECOM bit, CCAPnL and CCAPnH should be set with a non zero value, otherwise an unwanted match could happen. Writing to CCAPnH will set the ECOM bit.

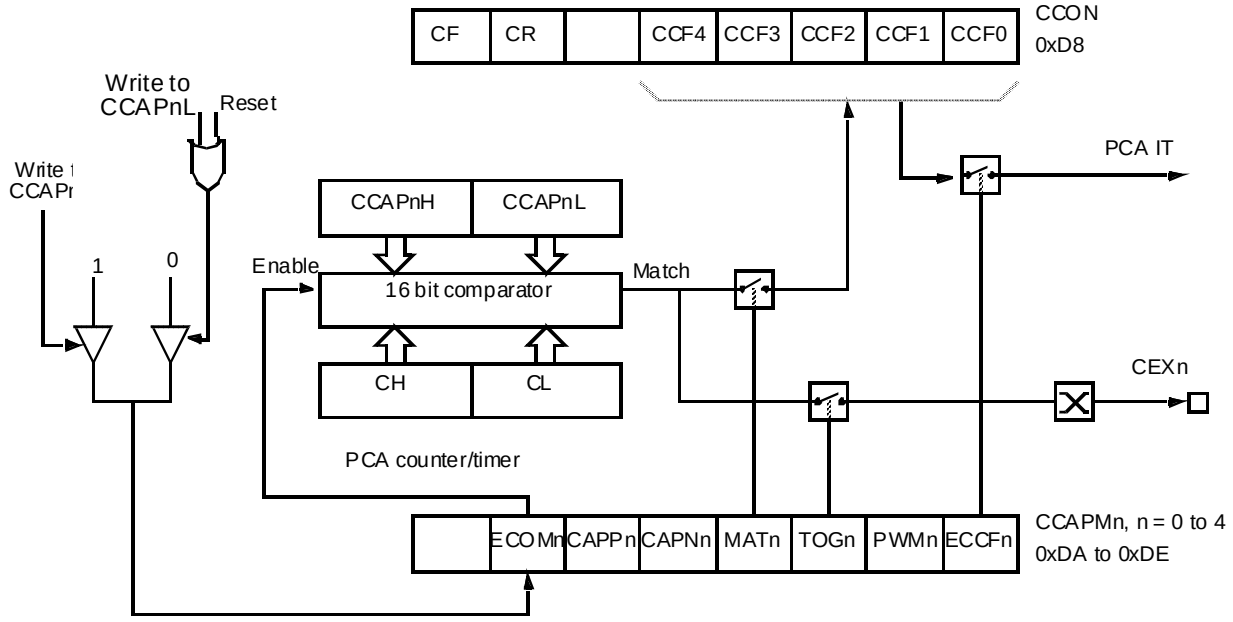
Once ECOM set, writing CCAPnL will clear ECOM so that an unwanted match doesn't occur while modifying the compare value. Writing to CCAPnH will set ECOM. For this reason, user software should write CCAPnL first, and then CCAPnH. Of course, the ECOM bit can still be controlled by accessing to CCAPMn register.

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 (See Figure 34).

A prior write must be done to CCAPnL and CCAPnH before writing the ECOMn bit.

Figure 34. PCA High Speed Output Mode



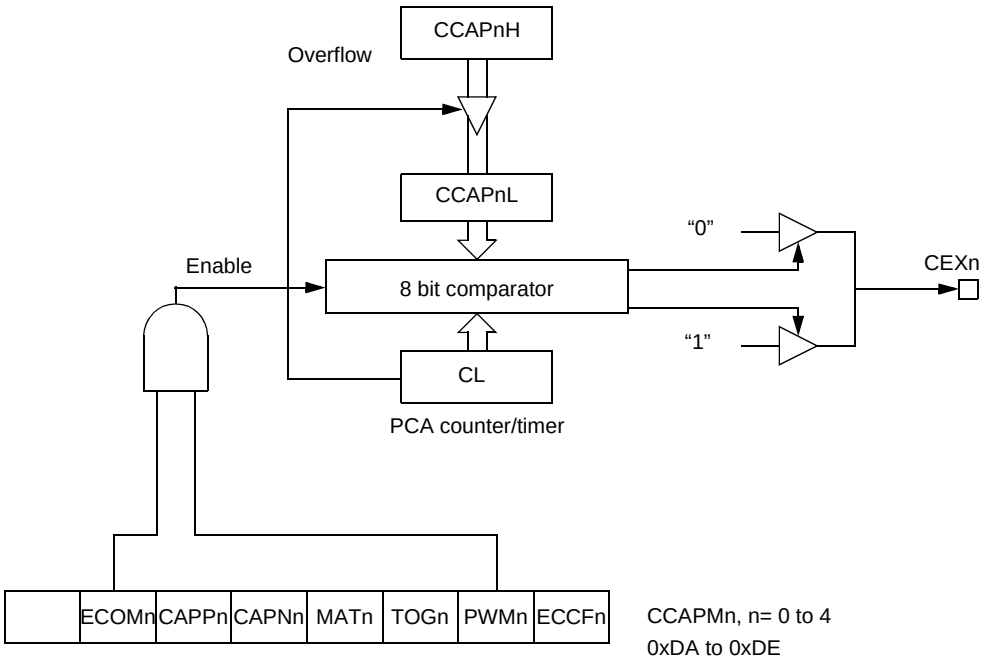
Before enabling ECOM bit, CCAPnL and CCAPnH should be set with a non zero value, otherwise an unwanted match could happen.

Once ECOM set, writing CCAPnL will clear ECOM so that an unwanted match doesn't occur while modifying the compare value. Writing to CCAPnH will set ECOM. For this reason, user software should write CCAPnL first, and then CCAPnH. Of course, the ECOM bit can still be controlled by accessing to CCAPMn register.

Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 35 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output 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 the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. This allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

Figure 35. PCA PWM Mode



PCA Watchdog Timer

An on-board watchdog timer is available with the PCA to improve the reliability of the system without increasing chip count. Watchdog timers are useful for systems that are susceptible 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. Figure 33 shows a diagram of how the watchdog works. 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.

In order to hold off the reset, the user has three options:

1. periodically change the compare value so it will never match the PCA timer,

2. periodically change the PCA timer value so it will never match the compare values, or

3. 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 option #3. If the program counter ever goes astray, a match will eventually occur and cause an internal reset. The second option is also not recommended if other PCA modules are being used. 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.

This watchdog timer won't generate a reset out on the reset pin.

Serial I/O Port

The serial I/O ports in the AT89C51RE2 are compatible with the serial I/O port in the 80C52. They provide both synchronous and asynchronous communication modes. They operates as a Universal Asynchronous Receiver and Transmitter (UART) in three full-duplex modes (Modes 1, 2 and 3). Asynchronous transmission and reception can occur simultaneously and at different baud rates

Both serial I/O port include the following enhancements:

- Framing error detection
- Automatic address recognition

As these improvements apply to both UART, most of the time in the following lines, there won't be any reference to UART_0 or UART_1, but only to UART, generally speaking.

Framing Error Detection

Framing bit error detection is provided for the three asynchronous modes (modes 1, 2 and 3). To enable the framing bit error detection feature, set SMOD0 bit in PCON register (See Figure 36) for UART 0 or set SMOD0_1 in BDRCON_1 register for UART 1 (See Figure 37).

Figure 36. UART 0 Framing Error Block Diagram

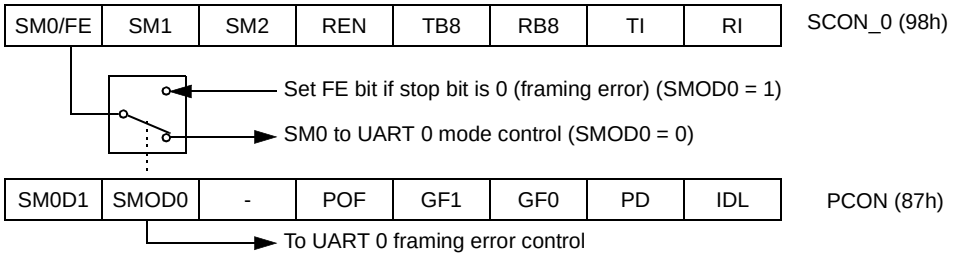
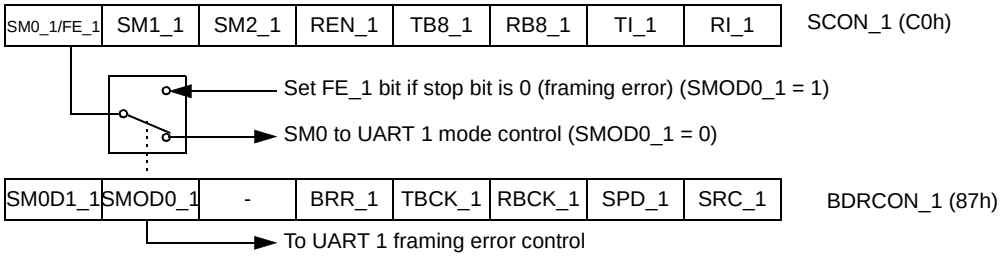


Figure 37. UART 1 Framing Error Block Diagram



When this feature is enabled, the receiver checks each incoming data frame for a valid stop bit. An invalid stop bit may result from noise on the serial lines or from simultaneous transmission by two CPUs. If a valid stop bit is not found, the Framing Error bit (FE) in SCON register (See Table 68.) bit is set.
Software may examine FE bit after each reception to check for data errors. Once set, only software or a reset can clear FE bit. Subsequently received frames with valid stop bits cannot clear FE bit. When FE feature is enabled, RI rises on stop bit instead of the last data bit (See Figure 38 and Figure 39).

Figure 38. UART Timings in Mode 1

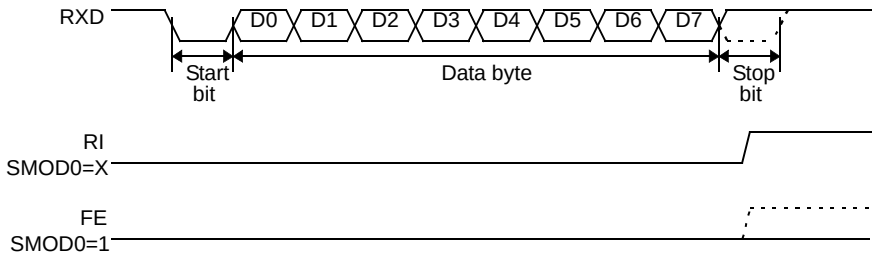
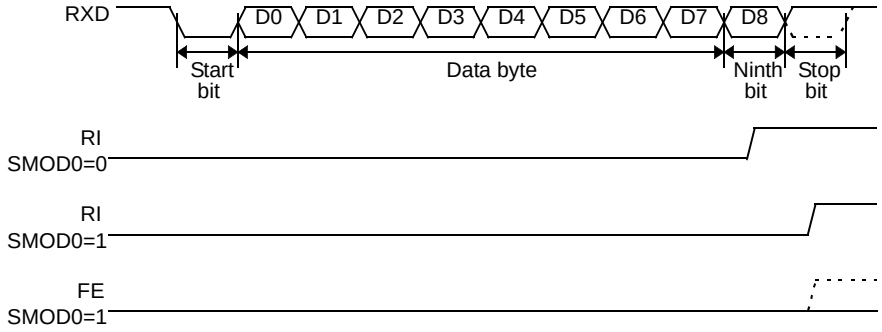


Figure 39. UART Timings in Modes 2 and 3



Automatic Address Recognition

The automatic address recognition feature is enabled when the multiprocessor communication feature is enabled (SM2 bit in SCON register is set).

Implemented in hardware, automatic address recognition enhances the multiprocessor communication feature by allowing the serial port to examine the address of each incoming command frame. Only when the serial port recognizes its own address, the receiver sets RI bit in SCON register to generate an interrupt. This ensures that the CPU is not interrupted by command frames addressed to other devices.

If desired, the user may enable the automatic address recognition feature in mode 1.In this configuration, the stop bit takes the place of the ninth data bit. Bit RI is set only when the received command frame address matches the device's address and is terminated by a valid stop bit.

To support automatic address recognition, a device is identified by a given address and a broadcast address.

Note: The multiprocessor communication and automatic address recognition features cannot be enabled in mode 0 (i. e. setting SM2 bit in SCON register in mode 0 has no effect).

Given Address

Each device has an individual address that is specified in SADDR register; the SADEN register is a mask byte that contains don't-care bits (defined by zeros) to form the device's given address. The don't-care bits provide the flexibility to address one or more slaves at a time. The following example illustrates how a given address is formed.

To address a device by its individual address, the SADEN mask byte must be `1111 1111b`.

For example:

SADDR0101 0110b
SADEN1111 1100b

Given0101 01XXb

The following is an example of how to use given addresses to address different slaves:

Slave A:SADDR1111 0001b
SADEN1111 1010b
Given1111 0X0Xb

Slave B:SADDR1111 0011b
SADEN1111 1001b
Given1111 0XX1b

Slave C:SADDR1111 0010b
SADEN1111 1101b
Given1111 00X1b

The SADEN byte is selected so that each slave may be addressed separately.
For slave A, bit 0 (the LSB) is a don't-care bit; for slaves B and C, bit 0 is a 1.To communicate with slave A only, the master must send an address where bit 0 is clear (e.g.. `1111 0000b`).
For slave A, bit 1 is a 1; for slaves B and C, bit 1 is a don't care bit. To communicate with slaves B and C, but not slave A, the master must send an address with bits 0 and 1 both set (e.g. `1111 0011b`).
To communicate with slaves A, B and C, the master must send an address with bit 0 set, bit 1 clear, and bit 2 clear (e.g. `1111 0001b`).

Broadcast Address

A broadcast address is formed from the logical OR of the SADDR and SADEN registers with zeros defined as don't-care bits, e.g.:

SADDR0101 0110b
SADEN1111 1100b

Broadcast =SADDR OR SADEN1111 111Xb

The use of don't-care bits provides flexibility in defining the broadcast address, however in most applications, a broadcast address is FFh. The following is an example of using broadcast addresses:

Slave A:SADDR1111 0001b
SADEN1111 1010b
Broadcast1111 1X11b,

Slave B:SADDR1111 0011b
SADEN1111 1001b
Broadcast1111 1X11B,

Slave C:SADDR=1111 0011b
SADEN1111 1101b
Broadcast1111 1111b

For slaves A and B, bit 2 is a don't care bit; for slave C, bit 2 is set. To communicate with all of the slaves, the master must send an address FFh. To communicate with slaves A and B, but not slave C, the master can send and address FBh.

Reset Addresses

On reset, the SADDR and SADEN registers are initialized to 00h, i. e. the given and broadcast addresses are `XXXX XXXXb` (all don't-care bits). This ensures that the serial port will reply to any address, and so, that it is backwards compatible with the 80C51 microcontrollers that do not support automatic address recognition.

Registers

Table 62. SADEN_0 Register

SADEN - Slave Address Mask Register UART 0(B9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b
Not bit addressable

Table 63. SADDR_0 Register

SADDR - Slave Address Register UART 0(A9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b
Not bit addressable

Table 64. SADEN_1 Register

SADEN_1 - Slave Address Mask Register UART 1(BAh)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b
Not bit addressable

Table 65. SADDR_1 Register

SADDR_1 - Slave Address Register UART 1(AAh)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b
Not bit addressable

Baud Rate Selection for UART 0 for Mode 1 and 3

The Baud Rate Generator for transmit and receive clocks can be selected separately via the T2CON and BDRCON_0 registers.

Figure 40. Baud Rate Selection for UART 0

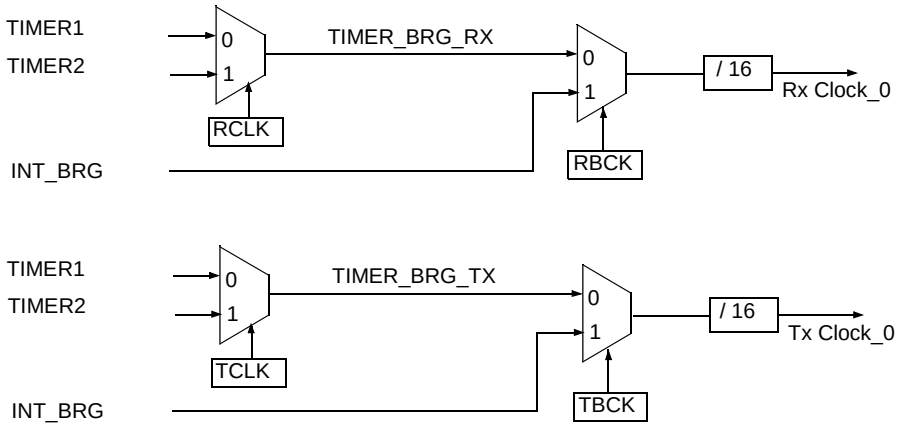


Table 66. Baud Rate Selection Table UART 0

TCLK (T2CON)	RCLK (T2CON)	TBCK (BDRCON)	RBCK (BDRCON)	Clock Source UART Tx	Clock Source UART Rx
0	0	0	0	Timer 1	Timer 1
1	0	0	0	Timer 2	Timer 1
0	1	0	0	Timer 1	Timer 2
1	1	0	0	Timer 2	Timer 2
X	0	1	0	INT_BRG	Timer 1
X	1	1	0	INT_BRG	Timer 2
0	X	0	1	Timer 1	INT_BRG
1	X	0	1	Timer 2	INT_BRG
X	X	1	1	INT_BRG	INT_BRG

Baud Rate Selection for UART 1 for Mode 1 and 3

The Baud Rate Generator for transmit and receive clocks can be selected separately via the T2CON and BDRCON_1 registers.

Figure 41. Baud Rate Selection for UART 1

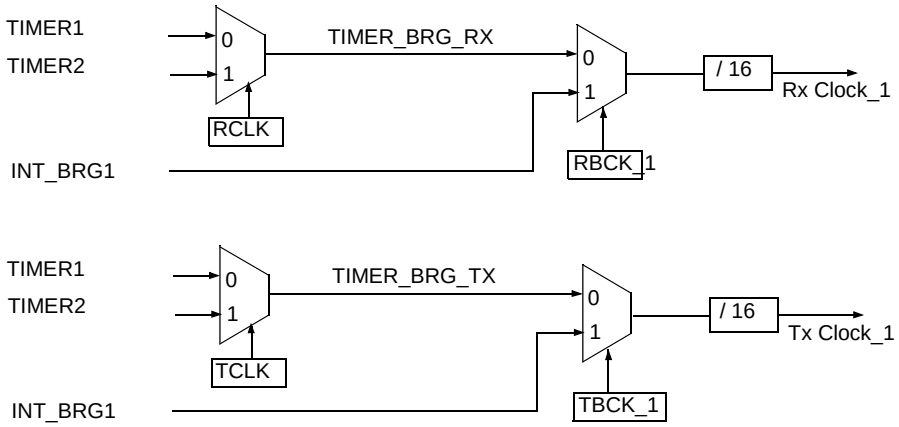


Table 67. Baud Rate Selection Table UART 1

TCLK (T2CON)	RCLK (T2CON)	TBCK_1 (BDRCON_1)	RBCK_1 (BDRCON_1)	Clock Source UART Tx_1	Clock Source UART Rx_1
0	0	0	0	Timer 1	Timer 1
1	0	0	0	Timer 2	Timer 1
0	1	0	0	Timer 1	Timer 2
1	1	0	0	Timer 2	Timer 2
X	0	1	0	INT_BRG_1	Timer 1
X	1	1	0	INT_BRG_1	Timer 2
0	X	0	1	Timer 1	INT_BRG_1
1	X	0	1	Timer 2	INT_BRG_1
X	X	1	1	INT_BRG_1	INT_BRG_1

Internal Baud Rate Generator (BRG)

The AT89C51RE2 implements two internal baudrate generators. Each one is dedicated to the corresponding UART. The configuration and operating mode for both BRG are similar. When an internal Baud Rate Generator is used, the Baud Rates are determined by the BRG overflow depending on the BRL (BRL or BRL_1 registers) reload value, the value of SPD (or SPD_1) bit (Speed Mode) in BDRCON (BDRCON_1) register and the value of the SMOD1 bit in PCON register.

Figure 42. Internal Baud Rate generator 0

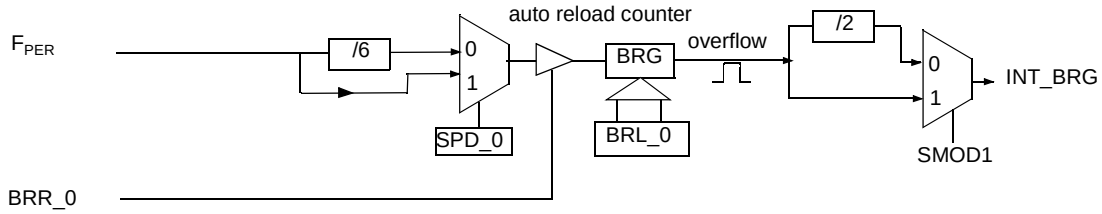
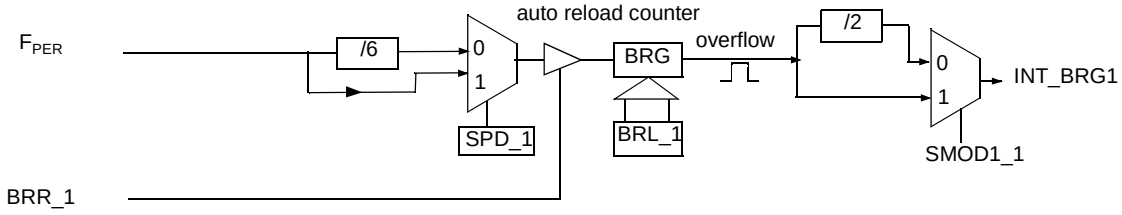


Figure 43. Internal Baud Rate generator 1



- The baud rate for UART is token by formula:

$$Baud\_Rate = \frac{2^{SMOD1} \cdot F_{PER}}{6^{(1-SPD)} \cdot 32 \cdot (256 - BRL)}$$

$$BRL = 256 - \frac{2^{SMOD1} \cdot F_{PER}}{6^{(1-SPD)} \cdot 32 \cdot Baud\_Rate}$$

Table 68. SCON_0 register

SCON_0 - Serial Control Register for UART 0(98h)

7	6	5	4	3	2	1	0
FE/SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0

Bit Number	Bit Mnemonic	Description
7	FE_0	**Framing Error bit (SMOD0=1)** Clear to reset the error state, not cleared by a valid stop bit. Set by hardware when an invalid stop bit is detected. SMOD0 must be set to enable access to the FE bit.
	SM0_0	**Serial port Mode bit 0** Refer to SM1_0 for serial port mode selection. SMOD0_0 must be cleared to enable access to the SM0_0 bit.
6	SM1_0	**Serial port Mode bit 1** SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register $F_{CPU\ PERIPH}/6$ 0 1 1 8-bit UART Variable 1 0 2 9-bit UART $F_{CPU\ PERIPH}$ /32 or /16 1 1 3 9-bit UART Variable
5	SM2_0	Serial port Mode 2 bit / Multiprocessor Communication Enable bit Clear to disable multiprocessor communication feature. Set to enable multiprocessor communication feature in mode 2 and 3, and eventually mode 1.This bit should be cleared in mode 0.
4	REN_0	**Reception Enable bit** Clear to disable serial reception. Set to enable serial reception.
3	TB8_0	Transmitter Bit 8 / Ninth bit to transmit in modes 2 and 3 Clear to transmit a logic 0 in the 9th bit. Set to transmit a logic 1 in the 9th bit.
2	RB8_0	**Receiver Bit 8 / Ninth bit received in modes 2 and 3** Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1. In mode 1, if SM2_0 = 0, RB8 is the received stop bit. In mode 0 RB8 is not used.
1	TI_0	**Transmit Interrupt flag** Clear to acknowledge interrupt. 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.
0	RI_0	**Receive Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 38. and Figure 39. in the other modes.

Reset Value = 0000 0000b
Bit addressable

Table 69. SCON_1 Register

SCON_1 - Serial Control Register for UART 1(C0h)

7	6	5	4	3	2	1	0
FE/SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1

Bit Number	Bit Mnemonic	Description
7	FE_1	**Framing Error bit (SMOD0=1)** Clear to reset the error state, not cleared by a valid stop bit. Set by hardware when an invalid stop bit is detected. SMOD0_1 must be set to enable access to the FE_1 bit.
	SM0_1	**Serial port Mode bit 0** Refer to SM1_1 for serial port mode selection. SMOD0_1 must be cleared to enable access to the SM0_1 bit.
6	SM1_1	**Serial port Mode bit 1** SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register $F_{CPU\ PERIPH}/6$ 0 1 1 8-bit UART Variable 1 0 2 9-bit UART $F_{CPU\ PERIPH}$ /32 or /16 1 1 3 9-bit UART Variable
5	SM2_1	Serial port Mode 2 bit / Multiprocessor Communication Enable bit Clear to disable multiprocessor communication feature. Set to enable multiprocessor communication feature in mode 2 and 3, and eventually mode 1.This bit should be cleared in mode 0.
4	REN_1	**Reception Enable bit** Clear to disable serial reception. Set to enable serial reception.
3	TB8_1	Transmitter Bit 8 / Ninth bit to transmit in modes 2 and 3 Clear to transmit a logic 0 in the 9th bit. Set to transmit a logic 1 in the 9th bit.
2	RB8_1	**Receiver Bit 8 / Ninth bit received in modes 2 and 3** Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1. In mode 1, if SM2_1 = 0, RB8 is the received stop bit. In mode 0 RB8 is not used.
1	TI_1	**Transmit Interrupt flag** Clear to acknowledge interrupt. 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.
0	RI_1	**Receive Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 38. and Figure 39. in the other modes.

Reset Value = 0000 0000b
Bit addressable

Table 70. Example of Computed Value When X2=1, SMOD1=1, SPD=1

Baud Rates	F_{osc} = 16. 384 MHz		F_{osc} = 24MHz	
	BRL	Error (%)	BRL	Error (%)
115200	247	1.23	243	0.16
57600	238	1.23	230	0.16
38400	229	1.23	217	0.16
28800	220	1.23	204	0.16
19200	203	0.63	178	0.16
9600	149	0.31	100	0.16
4800	43	1.23	-	-

Table 71. Example of Computed Value When X2=0, SMOD1=0, SPD=0

Baud Rates	F_{osc} = 16. 384 MHz		F_{osc} = 24MHz	
	BRL	Error (%)	BRL	Error (%)
4800	247	1.23	243	0.16
2400	238	1.23	230	0.16
1200	220	1.23	202	3.55
600	185	0.16	152	0.16

The baud rate generator can be used for mode 1 or 3 (refer to Figure 40.), but also for mode 0 for UART, thanks to the bit SRC located in BDRCON register (Table 78.)

UART Registers

Table 72. SBUF_0 register

SBUF_0 - Serial Buffer Register for UART 0(99h)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXb

Table 73. BRL_0 register

BRL_0 - Baud Rate Reload Register for the internal baud rate generator 0 (9Ah)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 74. SBUF_1 Register

SBUF - Serial Buffer Register for UART 1(C1h)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXb

Table 75. BRL_1 Register

BRL - Baud Rate Reload Register for the internal baud rate generator 1 (BBh)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 76. T2CON Register

T2CON - Timer 2 Control Register (C8h)

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 overflow Flag** Must be cleared by software. Set by hardware on timer 2 overflow, if RCLK = 0 and TCLK = 0.
6	EXF2	**Timer 2 External Flag** Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. When set, causes the CPU to vector to timer 2 interrupt routine when timer 2 interrupt is enabled. Must be cleared by software. EXF2 doesn't cause an interrupt in Up/down counter mode (DCEN = 1)
5	RCLK	**Receive Clock bit for UART** Cleared to use timer 1 overflow as receive clock for serial port in mode 1 or 3. Set to use timer 2 overflow as receive clock for serial port in mode 1 or 3.
4	TCLK	**Transmit Clock bit for UART** Cleared to use timer 1 overflow as transmit clock for serial port in mode 1 or 3. Set to use timer 2 overflow as transmit clock for serial port in mode 1 or 3.
3	EXEN2	**Timer 2 External Enable bit** Cleared to ignore events on T2EX pin for timer 2 operation. Set to cause a capture or reload when a negative transition on T2EX pin is detected, if timer 2 is not used to clock the serial port.
2	TR2	**Timer 2 Run control bit** Cleared to turn off timer 2. Set to turn on timer 2.
1	C/T2#	**Timer/Counter 2 select bit** Cleared for timer operation (input from internal clock system: $F_{CLK\ PERIPH}$). Set for counter operation (input from T2 input pin, falling edge trigger). Must be 0 for clock out mode.
0	CP/RL2#	**Timer 2 Capture/Reload bit** If RCLK=1 or TCLK=1, CP/RL2# is ignored and timer is forced to auto-reload on timer 2 overflow. Cleared to auto-reload on timer 2 overflows or negative transitions on T2EX pin if EXEN2=1. Set to capture on negative transitions on T2EX pin if EXEN2=1.

Reset Value = 0000 0000b
Bit addressable

Table 77. PCON Register

PCON - Power Control Register (87h)

7	6	5	4	3	2	1	0
SMOD1_0	**SMOD0_0**	**-**	**POF**	**GF1**	**GF0**	**PD**	**IDL**

Bit Number	Bit Mnemonic	Description
7	SMOD1_0	**Serial port Mode bit 1 for UART** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0_0	Serial port Mode bit 0 for UART Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-Off Flag** Cleared to recognize next reset type. Set by hardware when VCC rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
2	GF0	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
1	PD	**Power-Down mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle mode bit** Cleared by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00X1 0000b
Not bit addressable

Power-off flag reset value will be 1 only after a power on (cold reset). A warm reset doesn't affect the value of this bit.

Table 78. BDRCON_0 Register

BDRCON_0 - Baud Rate Control Register (9Bh)

7	6	5	4	3	2	1	0
-	-	-	BRR_0	TBCK_0	RBCK_0	SPD_0	SRC_0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	BRR_0	**Baud Rate Run Control bit** Cleared to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK_0	**Transmission Baud rate Generator Selection bit for UART** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
2	RBCK_0	**Reception Baud Rate Generator Selection bit for UART** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
1	SPD_0	**Baud Rate Speed Control bit for UART** Cleared to select the SLOW Baud Rate Generator. Set to select the FAST Baud Rate Generator.
0	SRC_0	**Baud Rate Source select bit in Mode 0 for UART** Cleared to select F_{OSC}/12 as the Baud Rate Generator ($F_{CLK\ PERIPH}$/6 in X2 mode). Set to select the internal Baud Rate Generator for UARTs in mode 0.

Reset Value = XXX0 0000b
Not bit addressable

Table 79. BDRCON_1 Register

BDRCON - Baud Rate Control Register (BCh)

7	6	5	4	3	2	1	0
SMOD1_1	**SMOD0_1**	**-**	**BRR_1**	**TBCK_1**	**RBCK_1**	**SPD_1**	**SRC_1**

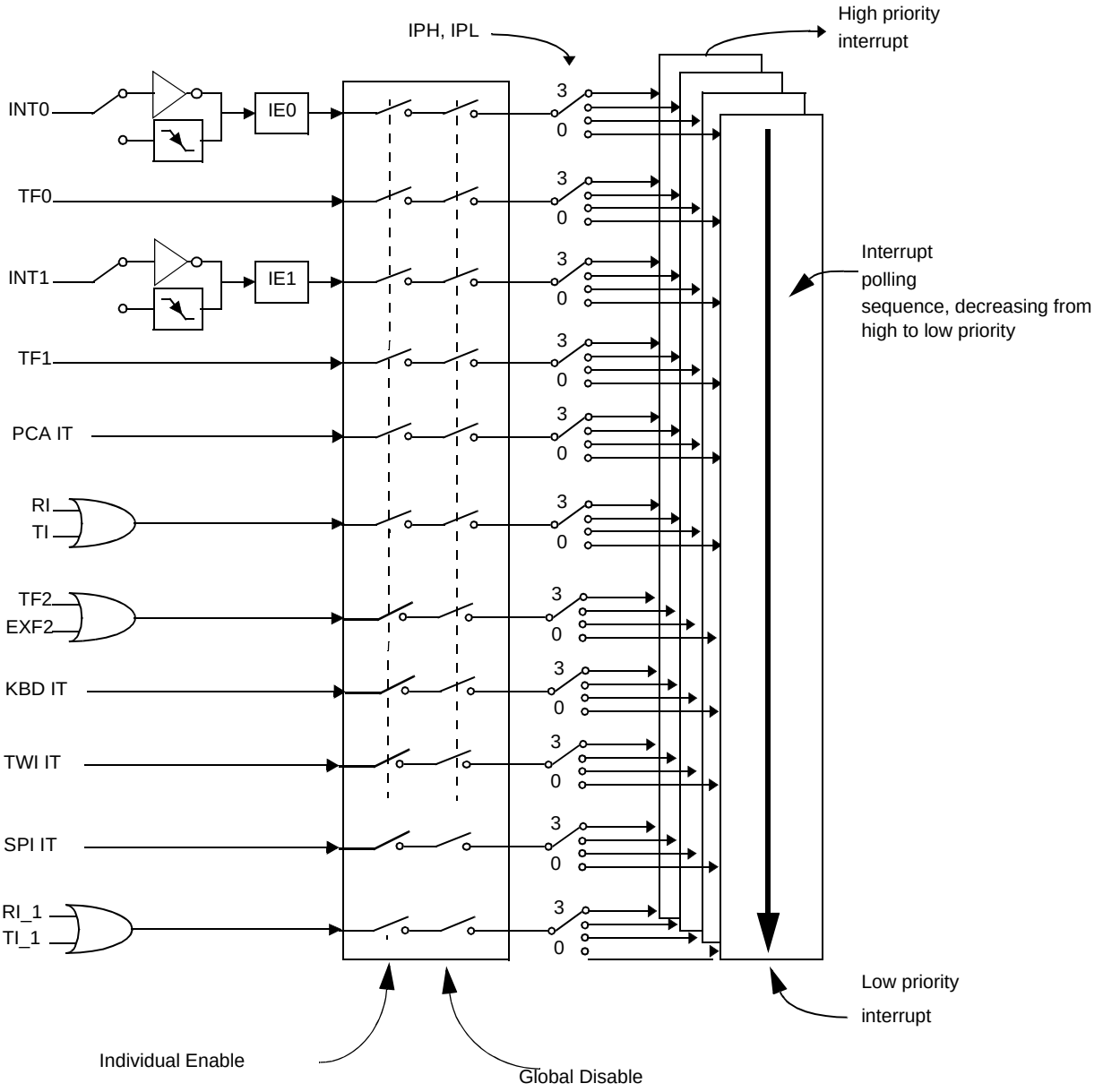
Bit Number	Bit Mnemonic	Description
7	SMOD1_1	**Serial port Mode bit 1 for UART 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0_1	Serial port Mode bit 0 for UART 1 Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	BRR_1	**Baud Rate Run Control bit** Cleared to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK_1	**Transmission Baud rate Generator Selection bit for UART 1** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
2	RBCK_1	**Reception Baud Rate Generator Selection bit for UART 1** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
1	SPD_1	**Baud Rate Speed Control bit for UART 1** Cleared to select the SLOW Baud Rate Generator. Set to select the FAST Baud Rate Generator.
0	SRC_1	**Baud Rate Source select bit in Mode 0 for UART 1** Cleared to select F_{OSC}/12 as the Baud Rate Generator ($F_{CLK\ PERIPH}$/6 in X2 mode). Set to select the internal Baud Rate Generator for UARTs in mode 0.

Reset Value = 0000 0000b
Not bit addressable

Interrupt System

The AT89C51RE2 has a total of 10 interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (timers 0, 1 and 2), two serial ports interrupts, SPI interrupt, Keyboard interrupt and the PCA global interrupt. These interrupts are shown in Figure 44.

Figure 44. Interrupt Control System



Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable register (Table 84 and Table 82). This register also contains a global disable bit, which must be cleared to disable all interrupts at once.

Each interrupt source can also be individually programmed to one out of four priority levels by setting or clearing a bit in the Interrupt Priority register (Table 85) and in the Interrupt Priority High register (Table 83 and Table 84) shows the bit values and priority levels associated with each combination.

Table 80. Priority Level Bit Values

iph. x	ipl. x	interrupt level priority
0	0	0 (lowest)
0	1	1
1	0	2
1	1	3 (highest)

A low-priority interrupt can be interrupted by a high priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level 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.

Table 81. IEN0 Register

IEN0 - Interrupt Enable Register (A8h)

7	6	5	4	3	2	1	0
EA	**EC**	**ET2**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

Bit Number	Bit Mnemonic	Description
7	EA	**Enable All interrupt bit** Cleared to disable all interrupts. Set to enable all interrupts.
6	EC	**PCA interrupt enable bit** Cleared to disable. Set to enable.
5	ET2	**Timer 2 overflow interrupt Enable bit** Cleared to disable timer 2 overflow interrupt. Set to enable timer 2 overflow interrupt.
4	ES	**Serial port 0 Enable bit** Cleared to disable serial port interrupt. Set to enable serial port interrupt.
3	ET1	**Timer 1 overflow interrupt Enable bit** Cleared to disable timer 1 overflow interrupt. Set to enable timer 1 overflow interrupt.
2	EX1	**External interrupt 1 Enable bit** Cleared to disable external interrupt 1. Set to enable external interrupt 1.
1	ET0	**Timer 0 overflow interrupt Enable bit** Cleared to disable timer 0 overflow interrupt. Set to enable timer 0 overflow interrupt.
0	EX0	**External interrupt 0 Enable bit** Cleared to disable external interrupt 0. Set to enable external interrupt 0.

Reset Value = 0000 0000b
Bit addressable

Table 82. IPL0 Register

IPL0 - Interrupt Priority Register (B8h)

7	6	5	4	3	2	1	0
-	PPCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	PPCL	**PCA interrupt Priority bit** Refer to PPCH for priority level.
5	PT2L	**Timer 2 overflow interrupt Priority bit** Refer to PT2H for priority level.
4	PSL	**Serial port 0 Priority bit** Refer to PSH for priority level.
3	PT1L	**Timer 1 overflow interrupt Priority bit** Refer to PT1H for priority level.
2	PX1L	**External interrupt 1 Priority bit** Refer to PX1H for priority level.
1	PT0L	**Timer 0 overflow interrupt Priority bit** Refer to PT0H for priority level.
0	PX0L	**External interrupt 0 Priority bit** Refer to PX0H for priority level.

Reset Value = X000 0000b
Bit addressable

Table 83. IPH0 Register

IPH0 - Interrupt Priority High Register (B7h)

7	6	5	4	3	2	1	0
-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	PPCH	**PCA interrupt Priority high bit.** PPCH PPCL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
5	PT2H	**Timer 2 overflow interrupt Priority High bit** PT2H PT2L Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
4	PSH	**Serial port Priority High bit** PSH PSL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
3	PT1H	**Timer 1 overflow interrupt Priority High bit** PT1H PT1L Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
2	PX1H	**External interrupt 1 Priority High bit** PX1H PX1L Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
1	PT0H	**Timer 0 overflow interrupt Priority High bit** PT0H PT0L Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
0	PX0H	**External interrupt 0 Priority High bit** PX0H PX0L Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest

Reset Value = X000 0000b
Not bit addressable

Table 84. IEN1 Register

IEN1 - Interrupt Enable Register (B1h)

7	6	5	4	3	2	1	0
-	-	-	-	ES_1	ESPI	ETWI	EKBD

Bit Number	Bit Mnemonic	Description
7	-	**Reserved**
6	-	**Reserved**
5	-	**Reserved**
4	-	**Reserved**
3	ES_1	**Serial port 1 Enable bit** Cleared to disable serial port interrupt. Set to enable serial port interrupt.
2	ESPI	**SPI interrupt Enable bit** Cleared to disable SPI interrupt. Set to enable SPI interrupt.
1	ETWI	**TWI interrupt Enable bit** Cleared to disable TWI interrupt. Set to enable TWI interrupt.
0	EKBD	**Keyboard interrupt Enable bit** Cleared to disable keyboard interrupt. Set to enable keyboard interrupt.

Reset Value = XXXX 00x0b
Bit addressable

Table 85. IPL1 Register

IPL1 - Interrupt Priority Register (B2h)

7	6	5	4	3	2	1	0
-	-	-	-	PSL_1	SPIL	TWIL	KBDL

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	PSL_1	**Serial port 1 Priority bit** Refer to PSH_1 for priority level.
2	SPIL	**SPI interrupt Priority bit** Refer to SPIH for priority level.
1	TWIL	**TWI interrupt Priority bit** Refer to TWIH for priority level.
0	KBDL	**Keyboard interrupt Priority bit** Refer to KBDH for priority level.

Reset Value = XXXX 00X0b
Bit addressable

Table 86. IPH1 Register

IPH1 - Interrupt Priority High Register (B3h)

7	6	5	4	3	2	1	0
-	-	-	-	PSH_1	SPIH	TWIH	KBDH

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	PSH_1	**Serial port 1 Priority High bit** PSH_1 PSL_1 Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
2	SPIH	**SPI interrupt Priority High bit** SPIH SPIL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
1	TWIH	**TWI interrupt Priority High bit** TWIH TWIL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
0	KBDH	**Keyboard interrupt Priority High bit** KB DH KBDL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest

Reset Value = XXXX 00X0b
Not bit addressable

Interrupt Sources and Vector Addresses

Table 87. Interrupt Sources and Vector Addresses

Number	Polling Priority	Interrupt Source	Interrupt Request	Vector Address
0	0	Reset		0000h
1	1	INT0	IE0	0003h
2	2	Timer 0	TF0	000Bh
3	3	INT1	IE1	0013h
4	4	Timer 1	IF1	001Bh
5	6	UART0	RI+TI	0023h
6	7	Timer 2	TF2+EXF2	002Bh
7	5	PCA	CF + CCFn (n = 0-4)	0033h
8	8	Keyboard	KBDIT	003Bh
9	9	TWI	TWIIT	0043h
10	10	SPI	SPIIT	004Bh
11	11	UART1	RI_1+TI_1	0053h

Power Management

Introduction

Two power reduction modes are implemented in the AT89C51RE2. The Idle mode and the Power-Down mode. These modes are detailed in the following sections. In addition to these power reduction modes, the clocks of the core and peripherals can be dynamically divided by 2 using the X2 mode detailed in Section “Enhanced Features”, page 13.

Idle Mode

Idle mode is a power reduction mode that reduces the power consumption. In this mode, program execution halts. Idle mode freezes the clock to the CPU at known states while the peripherals continue to be clocked. The CPU status before entering Idle mode is preserved, i.e., the program counter and program status word register retain their data for the duration of Idle mode. The contents of the SFRs and RAM are also retained. The status of the Port pins during Idle mode is detailed in Table 88.

Entering Idle Mode

To enter Idle mode, set the IDL bit in PCON register (see Table 89). The AT89C51RE2 enters Idle mode upon execution of the instruction that sets IDL bit. The instruction that sets IDL bit is the last instruction executed.

Note: If IDL bit and PD bit are set simultaneously, the AT89C51RE2 enters Power-Down mode. Then it does not go in Idle mode when exiting Power-Down mode.

Exiting Idle Mode

There are two ways to exit Idle mode:

1. Generate an enabled interrupt.
   - Hardware clears IDL bit in PCON register which restores the clock to the CPU. Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Idle mode. The general purpose flags (GF1 and GF0 in PCON register) may be used to indicate whether an interrupt occurred during normal operation or during Idle mode. When Idle mode is exited by an interrupt, the interrupt service routine may examine GF1 and GF0.
2. Generate a reset.
   - A logic high on the RST pin clears IDL bit in PCON register directly and asynchronously. This restores the clock to the CPU. Program execution momentarily resumes with the instruction immediately following the instruction that activated the Idle mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the AT89C51RE2 and vectors the CPU to address C:0000h.

Note: During the time that execution resumes, the internal RAM cannot be accessed; however, it is possible for the Port pins to be accessed. To avoid unexpected outputs at the Port pins, the instruction immediately following the instruction that activated Idle mode should not write to a Port pin or to the external RAM.

Power-Down Mode

The Power-Down mode places the AT89C51RE2 in a very low power state. Power-Down mode stops the oscillator, freezes all clock at known states. The CPU status prior to entering Power-Down mode is preserved, i.e., the program counter, program status word register retain their data for the duration of Power-Down mode. In addition, the SFR and RAM contents are preserved. The status of the Port pins during Power-Down mode is detailed in Table 88.

Note: VCC may be reduced to as low as V_{RET} during Power-Down mode to further reduce power dissipation. Take care, however, that VDD is not reduced until Power-Down mode is invoked.

Entering Power-Down Mode

To enter Power-Down mode, set PD bit in PCON register. The AT89C51RE2 enters the Power-Down mode upon execution of the instruction that sets PD bit. The instruction that sets PD bit is the last instruction executed.

Exiting Power-Down Mode

Note: If VCC was reduced during the Power-Down mode, do not exit Power-Down mode until VCC is restored to the normal operating level.

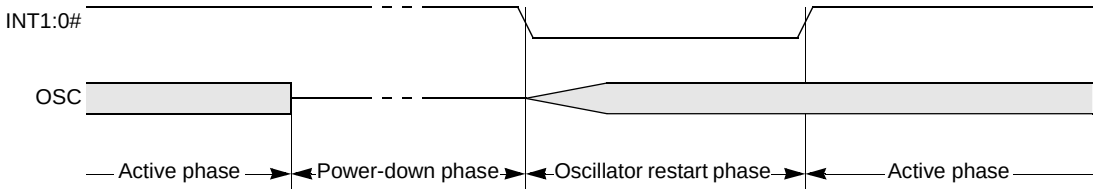
There are two ways to exit the Power-Down mode:

1. Generate an enabled external interrupt.
   - The AT89C51RE2 provides capability to exit from Power-Down using INT0#, INT1#. Hardware clears PD bit in PCON register which starts the oscillator and restores the clocks to the CPU and peripherals. Using INTx# input, execution resumes when the input is released (see Figure 45). Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power-Down mode.

Note: The external interrupt used to exit Power-Down mode must be configured as level sensitive (INT0# and INT1#) and must be assigned the highest priority. In addition, the duration of the interrupt must be long enough to allow the oscillator to stabilize. The execution will only resume when the interrupt is deasserted.

Note: Exit from power-down by external interrupt does not affect the SFRs nor the internal RAM content.

Figure 45. Power-Down Exit Waveform Using INT1:0#



2. Generate a reset.
   - A logic high on the RST pin clears PD bit in PCON register directly and asynchronously. This starts the oscillator and restores the clock to the CPU and peripherals. Program execution momentarily resumes with the instruction immediately following the instruction that activated Power-Down mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the AT89C51RE2 and vectors the CPU to address 0000h.

Note: During the time that execution resumes, the internal RAM cannot be accessed; however, it is possible for the Port pins to be accessed. To avoid unexpected outputs at the Port pins, the instruction immediately following the instruction that activated the Power-Down mode should not write to a Port pin or to the external RAM.

Note: Exit from power-down by reset redefines all the SFRs, but does not affect the internal RAM content.

Table 88. 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

Registers

Table 89. PCON Register

PCON (87:h) Power configuration Register

7	6	5	4	3	2	1	0
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial Port Mode bit 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	**Serial Port Mode bit 0** Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	reserved
4	POF	**Power-Off Flag** Cleared to recognize next reset type. Set by hardware when VCC rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General Purpose flag 1** One use is to indicate whether an interrupt occurred during normal operation or during Idle mode.
2	GF0	**General Purpose flag 0** One use is to indicate whether an interrupt occurred during normal operation or during Idle mode.
1	PD	**Power-Down Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Power-Down mode. If IDL and PD are both set, PD takes precedence.
0	IDL	**Idle Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Idle mode. If IDL and PD are both set, PD takes precedence.

Reset Value= XXXX 0000b

Oscillator

To optimize the power consumption and execution time needed for a specific task, an internal prescaler feature has been implemented between the oscillator and the CPU and peripherals.

Registers

Table 90. CKRL Register

CKRL – Clock Reload Register (97h)

7	6	5	4	3	2	1	0
CKRL7	CKRL6	CKRL5	CKRL4	CKRL3	CKRL2	CKRL1	CKRL0

Bit Number	Mnemonic	Description
7:0	CKRL	**Clock Reload Register** Prescaler value

Reset Value = 1111 1111b
Not bit addressable

Table 91. PCON Register

PCON – Power Control Register (87h)

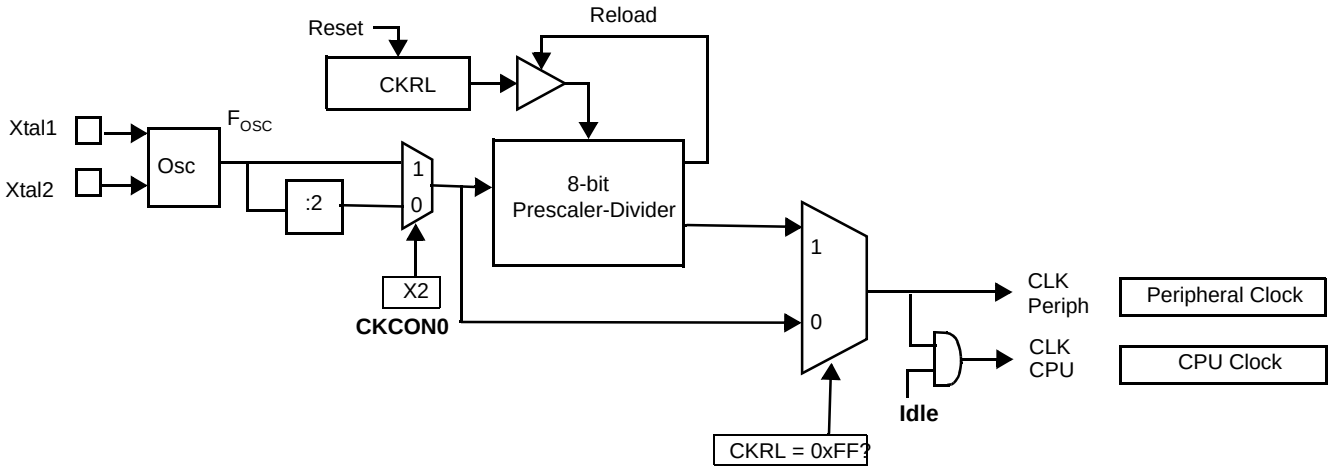
7	6	5	4	3	2	1	0
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial Port Mode bit 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	**Serial Port Mode bit 0** Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-off Flag** Cleared by software to recognize the next reset type. Set by hardware when V_{CC} rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General-purpose Flag** Cleared by software for general-purpose usage. Set by software for general-purpose usage.
2	GF0	**General-purpose Flag** Cleared by software for general-purpose usage. Set by software for general-purpose usage.
1	PD	**Power-down Mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle Mode bit** Cleared by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00X1 0000b Not bit addressable

Functional Block Diagram

Figure 46. Functional Oscillator Block Diagram



Prescaler Divider

- A hardware RESET puts the prescaler divider in the following state:
  - CKRL = FFh: $F_{CLK\ CPU} = F_{CLK\ PERIPH} = F_{OSC}/2$ (Standard C51 feature)
- Any value between FFh down to 00h can be written by software into CKRL register in order to divide frequency of the selected oscillator:
  - CKRL = 00h: minimum frequency
    $F_{CLK\ CPU} = F_{CLK\ PERIPH} = F_{OSC}/1020$ (Standard Mode)
    $F_{CLK\ CPU} = F_{CLK\ PERIPH} = F_{OSC}/510$ (X2 Mode)
  - CKRL = FFh: maximum frequency
    $F_{CLK\ CPU} = F_{CLK\ PERIPH} = F_{OSC}/2$ (Standard Mode)
    $F_{CLK\ CPU} = F_{CLK\ PERIPH} = F_{OSC}$ (X2 Mode)

$F_{CLK\ CPU}$ and $F_{CLK\ PERIPH}$

In X2 Mode, for CKRL<>0xFF:

$$F_{CPU} = F_{CLKPERIPH} = \frac{F_{OSC}}{2 \times (255 - CKRL)}$$

In X1 Mode, for CKRL<>0xFF then:

$$F_{CPU} = F_{CLKPERIPH} = \frac{F_{OSC}}{4 \times (255 - CKRL)}$$

Hardware Watchdog Timer

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 14-bit counter and the WatchDog Timer ReSeT (WDTRST) SFR. The WDT is by default disabled from exiting reset. To enable the WDT, user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When WDT 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 drive an output RESET HIGH pulse at the RST-pin.

Using the WDT

To enable the WDT, user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When WDT is enabled, the user needs to service it by writing to 01EH and 0E1H to WDTRST to avoid WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH) and this will reset the device. When WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycle. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST-pin. The RESET pulse duration is 96 x $T_{CLK\ PERIPH}$, where $T_{CLK\ PERIPH}$= 1/$F_{CLK\ PERIPH}$. 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.

To have a more powerful WDT, a 2^7 counter has been added to extend the Time-out capability, ranking from 16ms to 2s @ F_{OSCA} = 12MHz. To manage this feature, refer to WDTPRG register description, Table 92.

Table 92. WDTRST Register

WDTRST - Watchdog Reset Register (0A6h)

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

Reset Value = XXXX XXXXb

Write only, this SFR is used to reset/enable the WDT by writing 01EH then 0E1H in sequence.

Table 93. WDTPRG Register

WDTPRG - Watchdog Timer Out Register (0A7h)

7	6	5	4	3	2	1	0
-	-	-	-	-	S2	S1	S0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is undetermined. Do not try to set this bit.
6	-	
5	-	
4	-	
3	-	
2	S2	**WDT Time-out select bit 2**
1	S1	**WDT Time-out select bit 1**
0	S0	**WDT Time-out select bit 0**
		S2 S1 S0 Selected Time-out 0 0 0 (2^{14} - 1) machine cycles, 16. 3 ms @ F_{OSCA} =12 MHz 0 0 1 (2^{15} - 1) machine cycles, 32.7 ms @ F_{OSCA}=12 MHz 0 1 0 (2^{16} - 1) machine cycles, 65. 5 ms @ F_{OSCA}=12 MHz 0 1 1 (2^{17} - 1) machine cycles, 131 ms @ F_{OSCA}=12 MHz 1 0 0 (2^{18} - 1) machine cycles, 262 ms @ F_{OSCA}=12 MHz 1 0 1 (2^{19} - 1) machine cycles, 542 ms @ F_{OSCA}=12 MHz 1 1 0 (2^{20} - 1) machine cycles, 1.05 s @ F_{OSCA}=12 MHz 1 1 1 (2^{21} - 1) machine cycles, 2.09 s @ F_{OSCA}=12 MHz

Reset value = XXXX X000

WDT During Power Down and Idle

In Power Down mode the oscillator stops, which means the WDT also stops. While in Power Down mode the user does not need to service the WDT. There are 2 methods of exiting Power Down mode: by a hardware reset or via a level activated external interrupt which is enabled prior to entering Power Down mode. When Power Down is exited with hardware reset, servicing the WDT should occur as it normally should whenever the AT89C51RE2 is reset. Exiting Power Down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service routine.

To ensure that the WDT does not overflow within a few states of exiting of powerdown, it is better to reset the WDT just before entering powerdown.

In the Idle mode, the oscillator continues to run. To prevent the WDT from resetting the AT89C51RE2 while in Idle mode, the user should always set up a timer that will periodically exit Idle, service the WDT, and re-enter Idle mode.

Reduced EMI Mode

The ALE signal is used to demultiplex address and data buses on port 0 when used with external program or data memory. Nevertheless, during internal code execution, ALE signal is still generated. In order to reduce EMI, ALE signal can be disabled by setting AO bit.

The AO bit is located in AUXR register at bit location 0. As soon as AO is set, ALE is no longer output but remains active during MOVX and MOVC instructions and external fetches. During ALE disabling, ALE pin is weakly pulled high.

Table 94. AUXR Register

AUXR - Auxiliary Register (8Eh)

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	M0	**Pulse length** Cleared to stretch MOVX control: the RD/ and the WR/ pulse length is 6 clock periods (default). Set to stretch MOVX control: the RD/ and the WR/ pulse length is 30 clock periods.
4	XRS2	**XRAM Size** XRS2 XRS1 XRS0 XRAM size 0 0 0 256 bytes 0 0 1 512 bytes 0 1 0 768 bytes(default) 0 1 1 1024 bytes 1 0 0 1792 bytes
3	XRS1	
2	XRS0	
1	EXTRAM	**EXTRAM bit** Cleared to access internal XRAM using movx @ Ri/ @ DPTR. Set to access external memory. Programmed by hardware after Power-up regarding Hardware Security Byte (HSB), default setting, XRAM selected.
0	AO	**ALE Output bit** Cleared, ALE is emitted at a constant rate of 1/6 the oscillator frequency (or 1/3 if X2 mode is used). (default) Set, ALE is active only during a MOVX or MOVC instruction is used.

Reset Value = XX00 10'HSB. XRAM'0b
Not bit addressable

Keyboard Interface

The AT89C51RE2 implements a keyboard interface allowing the connection of a 8 x n matrix keyboard. It is based on 8 inputs with programmable interrupt capability on both high or low level. These inputs are available as alternate function of P1 and allow to exit from idle and power down modes.

The keyboard interface interfaces with the C51 core through 3 special function registers: KBLS, the Keyboard Level Selection register (Table 97), KBE, The Keyboard interrupt Enable register (Table 96), and KBF, the Keyboard Flag register (Table 95).

Interrupt

The keyboard inputs are considered as 8 independent interrupt sources sharing the same interrupt vector. An interrupt enable bit (KBD in IE1) allows global enable or disable of the keyboard interrupt (see Figure 47). As detailed in Figure 48 each keyboard input has the capability to detect a programmable level according to KBLS. x bit value. Level detection is then reported in interrupt flags KBF. x that can be masked by software using KBE. x bits.
This structure allow keyboard arrangement from 1 by n to 8 by n matrix and allow usage of P1 inputs for other purpose.

Figure 47. Keyboard Interface Block Diagram

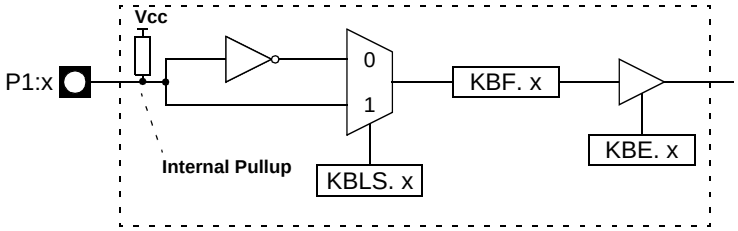
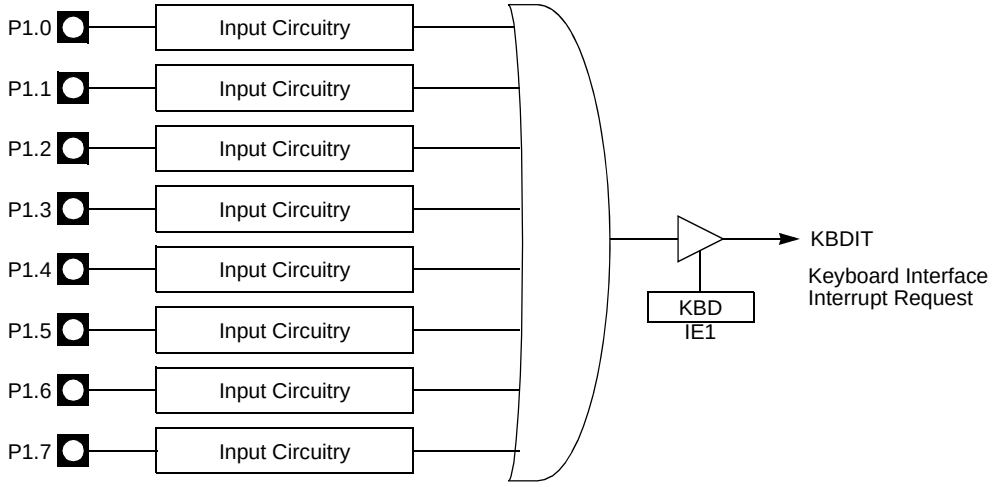


Figure 48. Keyboard Input Circuitry



Power Reduction Mode

P1 inputs allow exit from idle and power down modes as detailed in Section “Power Management”, page 118.

Registers

Table 95. KBF Register

KBF-Keyboard Flag Register (9Eh)

7	6	5	4	3	2	1	0
KBF7	KBF6	KBF5	KBF4	KBF3	KBF2	KBF1	KBF0

Bit Number	Bit Mnemonic	Description
7	KBF7	**Keyboard line 7 flag** Set by hardware when the Port line 7 detects a programmed level. It generates a Keyboard interrupt request if the KBKBIE. 7 bit in KBIE register is set. Must be cleared by software.
6	KBF6	**Keyboard line 6 flag** Set by hardware when the Port line 6 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 6 bit in KBIE register is set. Must be cleared by software.
5	KBF5	**Keyboard line 5 flag** Set by hardware when the Port line 5 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 5 bit in KBIE register is set. Must be cleared by software.
4	KBF4	**Keyboard line 4 flag** Set by hardware when the Port line 4 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 4 bit in KBIE register is set. Must be cleared by software.
3	KBF3	**Keyboard line 3 flag** Set by hardware when the Port line 3 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 3 bit in KBIE register is set. Must be cleared by software.
2	KBF2	**Keyboard line 2 flag** Set by hardware when the Port line 2 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 2 bit in KBIE register is set. Must be cleared by software.
1	KBF1	**Keyboard line 1 flag** Set by hardware when the Port line 1 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 1 bit in KBIE register is set. Must be cleared by software.
0	KBF0	**Keyboard line 0 flag** Set by hardware when the Port line 0 detects a programmed level. It generates a Keyboard interrupt request if the KBIE. 0 bit in KBIE register is set. Must be cleared by software.

Reset Value= 0000 0000b

Table 96. KBE Register

KBE-Keyboard Input Enable Register (9Dh)

7	6	5	4	3	2	1	0
KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0

Bit Number	Bit Mnemonic	Description
7	KBE7	**Keyboard line 7 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 7 bit in KBF register to generate an interrupt request.
6	KBE6	**Keyboard line 6 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 6 bit in KBF register to generate an interrupt request.
5	KBE5	**Keyboard line 5 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 5 bit in KBF register to generate an interrupt request.
4	KBE4	**Keyboard line 4 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 4 bit in KBF register to generate an interrupt request.
3	KBE3	**Keyboard line 3 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 3 bit in KBF register to generate an interrupt request.
2	KBE2	**Keyboard line 2 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 2 bit in KBF register to generate an interrupt request.
1	KBE1	**Keyboard line 1 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 1 bit in KBF register to generate an interrupt request.
0	KBE0	**Keyboard line 0 Enable bit** Cleared to enable standard I/O pin. Set to enable KBF. 0 bit in KBF register to generate an interrupt request.

Reset Value= 0000 0000b

Table 97. KBLS Register

KBLS-Keyboard Level Selector Register (9Ch)

7	6	5	4	3	2	1	0
KBLS7	**KBLS6**	**KBLS5**	**KBLS4**	**KBLS3**	**KBLS2**	**KBLS1**	**KBLS0**

Bit Number	Bit Mnemonic	Description
7	KBLS7	**Keyboard line 7 Level Selection bit** Cleared to enable a low level detection on Port line 7. Set to enable a high level detection on Port line 7.
6	KBLS6	**Keyboard line 6 Level Selection bit** Cleared to enable a low level detection on Port line 6. Set to enable a high level detection on Port line 6.
5	KBLS5	**Keyboard line 5 Level Selection bit** Cleared to enable a low level detection on Port line 5. Set to enable a high level detection on Port line 5.
4	KBLS4	**Keyboard line 4 Level Selection bit** Cleared to enable a low level detection on Port line 4. Set to enable a high level detection on Port line 4.
3	KBLS3	**Keyboard line 3 Level Selection bit** Cleared to enable a low level detection on Port line 3. Set to enable a high level detection on Port line 3.
2	KBLS2	**Keyboard line 2 Level Selection bit** Cleared to enable a low level detection on Port line 2. Set to enable a high level detection on Port line 2.
1	KBLS1	**Keyboard line 1 Level Selection bit** Cleared to enable a low level detection on Port line 1. Set to enable a high level detection on Port line 1.
0	KBLS0	**Keyboard line 0 Level Selection bit** Cleared to enable a low level detection on Port line 0. Set to enable a high level detection on Port line 0.

Reset Value= 0000 0000b

2-wire Interface (TWI)

This section describes the 2-wire interface. The 2-wire bus is a bi-directional 2-wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (IC) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. The serial data transfer is limited to 400 Kbit/s in standard mode. Various communication configuration can be designed using this bus. Figure 49 shows a typical 2-wire bus configuration. All the devices connected to the bus can be master and slave.

Figure 49. 2-wire Bus Configuration

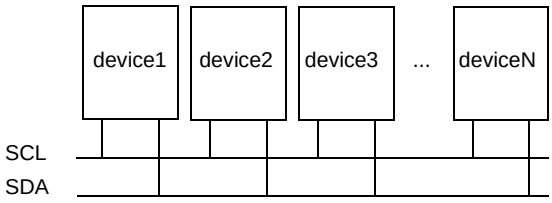
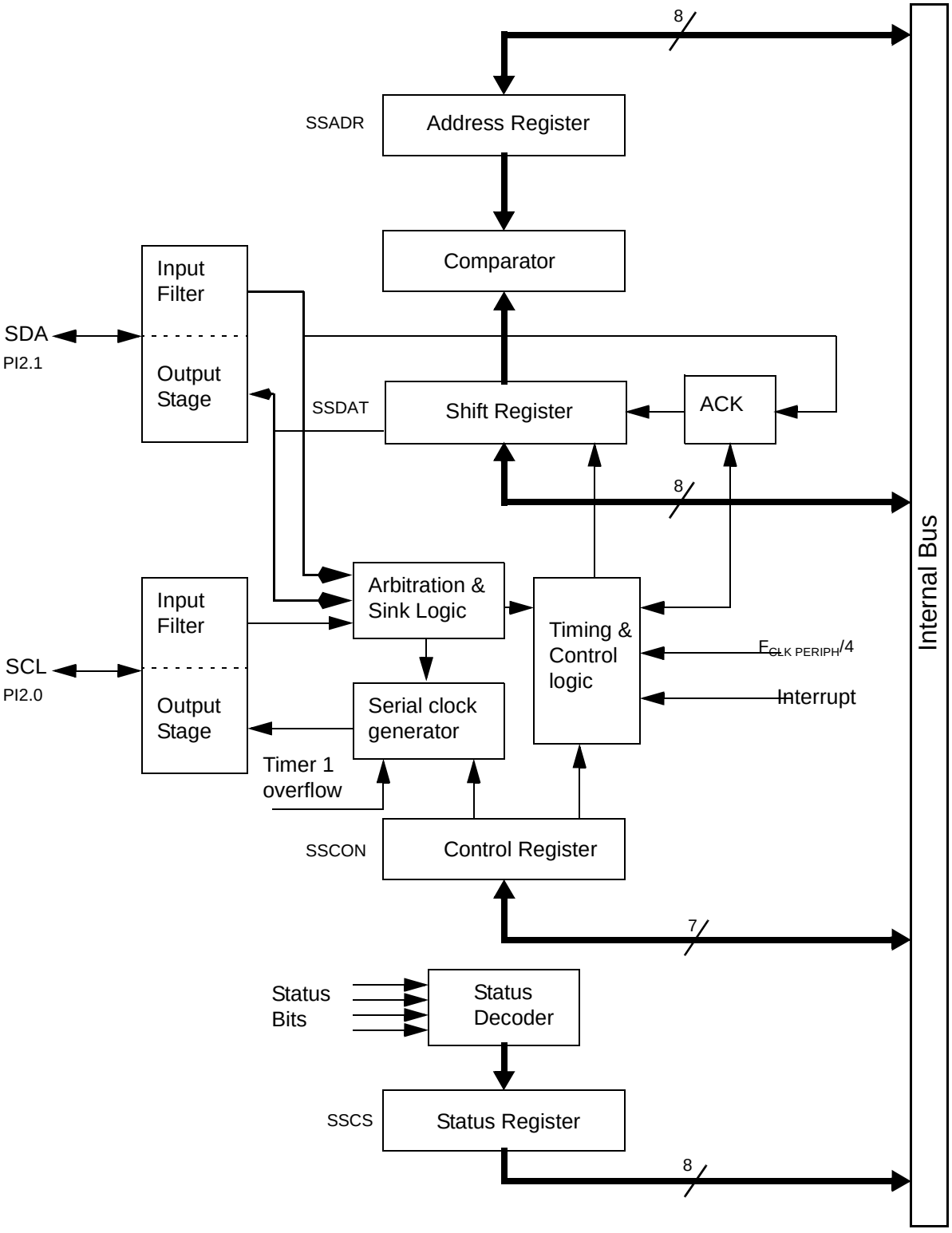


Figure 50. Block Diagram



Description

The CPU interfaces to the 2-wire logic via the following four 8-bit special function registers: the Synchronous Serial Control register (SSCON; Table 107), the Synchronous Serial Data register (SSDAT; Table 108), the Synchronous Serial Control and Status register (SSCS; Table 109) and the Synchronous Serial Address register (SSADR Table 112).

SSCON is used to enable the TWI interface, to program the bit rate (see Table 100), to enable slave modes, to acknowledge or not a received data, to send a START or a STOP condition on the 2-wire bus, and to acknowledge a serial interrupt. A hardware reset disables the TWI module.

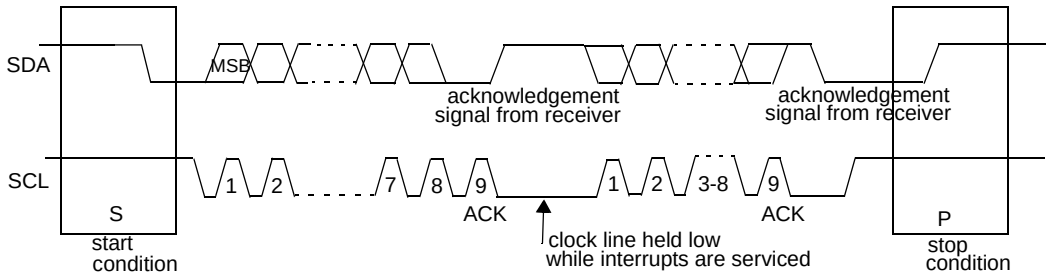
SSCS contains a status code which reflects the status of the 2-wire logic and the 2-wire bus. The three least significant bits are always zero. The five most significant bits contains the status code. There are 26 possible status codes. When SSCS contains F8h, no relevant state information is available and no serial interrupt is requested. A valid status code is available in SSCS one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software. to Table 106. give the status for the master modes and miscellaneous states.

SSDAT contains a byte of serial data to be transmitted or a byte which has just been received. It is addressable while it is not in process of shifting a byte. This occurs when 2-wire logic is in a defined state and the serial interrupt flag is set. Data in SSDAT remains stable as long as SI is set. While data is being shifted out, data on the bus is simultaneously shifted in; SSDAT always contains the last byte present on the bus.

SSADR may be loaded with the 7-bit slave address (7 most significant bits) to which the TWI module will respond when programmed as a slave transmitter or receiver. The LSB is used to enable general call address (00h) recognition.

Figure 51 shows how a data transfer is accomplished on the 2-wire bus.

Figure 51. Complete Data Transfer on 2-wire Bus



The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave transmitter
- Slave receiver

Data transfer in each mode of operation is shown in Table to Table 106 and Figure 52. to Figure 55.. These figures contain the following abbreviations:

S : START condition

R : Read bit (high level at SDA)

W: Write bit (low level at SDA)

A: Acknowledge bit (low level at SDA)

$\overline{A}$: Not acknowledge bit (high level at SDA)

Data: 8-bit data byte

P : STOP condition

In Figure 52 to Figure 55, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in SSCS. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When the serial interrupt routine is entered, the status code in SSCS is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Table to Table 106.

Master Transmitter Mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (Figure 52). Before the master transmitter mode can be entered, SSCON must be initialised as follows:

Table 98. SSCON Initialization

CR2	SSIE	STA	STO	SI	AA	CR1	CR0
bit rate	1	0	0	0	X	bit rate	bit rate

CR0, CR1 and CR2 define the internal serial bit rate if external bit rate generator is not used. SSIE must be set to enable TWI. STA, STO and SI must be cleared.

The master transmitter mode may now be entered by setting the STA bit. The 2-wire logic will now test the 2-wire bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI bit in SSCON) is set, and the status code in SSCS will be 08h. This status must be used to vector to an interrupt routine that loads SSDAT with the slave address and the data direction bit (SLA+W).

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, SI is set again and a number of status code in SSCS are possible. There are 18h, 20h or 38h for the master mode and also 68h, 78h or B0h if the slave mode was enabled (AA=logic 1). The appropriate action to be taken for each of these status code is detailed in Table . This scheme is repeated until a STOP condition is transmitted.

SSIE, CR2, CR1 and CR0 are not affected by the serial transfer and are referred to Table 7 to Table 11. After a repeated START condition (state 10h) the TWI module may switch to the master receiver mode by loading SSDAT with SLA+R.

Master Receiver Mode

In the master receiver mode, a number of data bytes are received from a slave transmitter (Figure 53). The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the interrupt routine must load SSDAT with the 7-bit slave address and the data direction bit (SLA+R). The serial interrupt flag SI must then be cleared before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, the serial interrupt flag is set again and a number of status code in SSCS are possible. There are 40h, 48h or 38h for the master mode and also 68h, 78h or B0h if the slave mode was enabled (AA=logic 1). The appropriate action to be taken for each of these status code is detailed in Table . This scheme is repeated until a STOP condition is transmitted.

SSIE, CR2, CR1 and CR0 are not affected by the serial transfer and are referred to Table 7 to Table 11. After a repeated START condition (state 10h) the TWI module may switch to the master transmitter mode by loading SSDAT with SLA+W.

Slave Receiver Mode

In the slave receiver mode, a number of data bytes are received from a master transmitter (Figure 54). To initiate the slave receiver mode, SSADR and SSCON must be loaded as follows:

Table 99. SSADR: Slave Receiver Mode Initialization

A6	A5	A4	A3	A2	A1	A0	GC
own slave address							

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

Table 100. SSCON: Slave Receiver Mode Initialization

CR2	SSIE	STA	STO	SI	AA	CR1	CR0
bit rate	1	0	0	0	1	bit rate	bit rate

CR0, CR1 and CR2 have no effect in the slave mode. SSIE must be set to enable the TWI. The AA bit must be set to enable the own slave address or the general call address acknowledgement. STA, STO and SI must be cleared.

When SSADR and SSCON have been initialised, the TWI module waits until it is addressed by its own slave address followed by the data direction bit which must be at logic 0 (W) for the TWI to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag is set and a valid status code can be read from SSCS. This status code is used to vector to an interrupt service routine.The appropriate action to be taken for each of these status code is detailed in Table . The slave receiver mode may also be entered if arbitration is lost while TWI is in the master mode (states 68h and 78h).

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

Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (Figure 55). Data transfer is initialized as in the slave receiver mode. When SSADR and SSCON have been initialized, the TWI module waits until it is addressed by

its own slave address followed by the data direction bit which must be at logic 1 (R) for TWI to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag is set and a valid status code can be read from SSCS. This status code is used to vector to an interrupt service routine. The appropriate action to be taken for each of these status code is detailed in Table . The slave transmitter mode may also be entered if arbitration is lost while the TWI module is in the master mode.

If the AA bit is reset during a transfer, the TWI module will transmit the last byte of the transfer and enter state C0h or C8h. the TWI module is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1's as serial data. While AA is reset, the TWI module does not respond to its own slave address. However, the 2-wire bus is still monitored and address recognition may be resume at any time by setting AA. This means that the AA bit may be used to temporarily isolate the TWI module from the 2-wire bus.

Miscellaneous States

There are two SSCS codes that do not correspond to a define TWI hardware state (Table 106). These codes are discuss hereafter.

Status F8h indicates that no relevant information is available because the serial interrupt flag is not set yet. This occurs between other states and when the TWI module is not involved in a serial transfer.

Status 00h indicates that a bus error has occurred during a TWI serial transfer. A bus error is caused when a START or a STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions happen during the serial transfer of an address byte, a data byte, or an acknowledge bit. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes the TWI module to enter the not addressed slave mode and to clear the STO flag (no other bits in SSCON are affected). The SDA and SCL lines are released and no STOP condition is transmitted.

Notes

the TWI module interfaces to the external 2-wire bus via two port pins: SCL (serial clock line) and SDA (serial data line). To avoid low level asserting on these lines when the TWI module is enabled, the output latches of SDA and SLC must be set to logic 1.

Table 101. Bit Frequency Configuration

			Bit Frequency (kHz)		
CR2	**CR1**	**CR0**	**F_{OSCA}= 12 MHz**	**F_{OSCA} = 16 MHz**	**F_{OSCA} divided by**
0	0	0	47	62.5	256
0	0	1	53.5	71.5	224
0	1	0	62.5	83	192
0	1	1	75	100	160
1	0	0	-	-	Unused
1	0	1	100	133.3	120
1	1	0	200	266.6	60
1	1	1	0.5 <. < 62.5	0.67 <. < 83	96 · (256 - reload valueTimer 1) (reload value range: 0-254 in mode 2)

Figure 52. Format and State in the Master Transmitter Mode

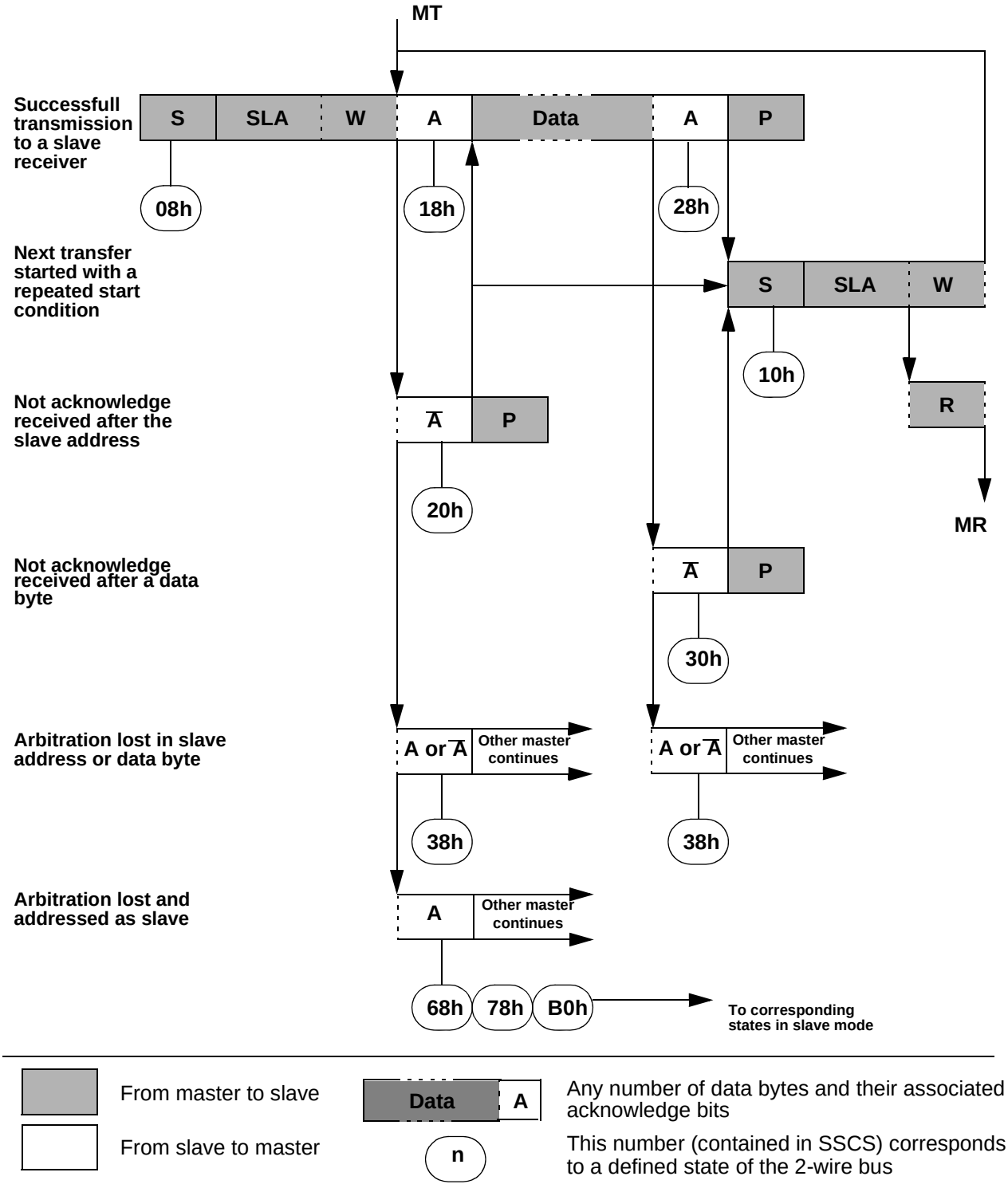


Table 102. Status in Master Transmitter Mode

Status Code SSSTA	Status of the Two-wire Bus and Two-wire Hardware	Application software response					Next Action Taken by Two-wire Hardware
		To/From SSDAT	To SSCON				
			SSSTA	SSSTO	SSI	SSAA	
08h	A START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted.
10h	A repeated START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted.
		Write SLA+R	X	0	0	X	SLA+R will be transmitted. Logic will switch to master receiver mode
18h	SLA+W has been transmitted; ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
20h	SLA+W has been transmitted; NOT ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
28h	Data byte has been transmitted; ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
30h	Data byte has been transmitted; NOT ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
38h	Arbitration lost in SLA+W or data bytes	No SSDAT action	0	0	0	X	Two-wire bus will be released and not addressed slave mode will be entered.
		No SSDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.

Figure 53. Format and State in the Master Receiver Mode

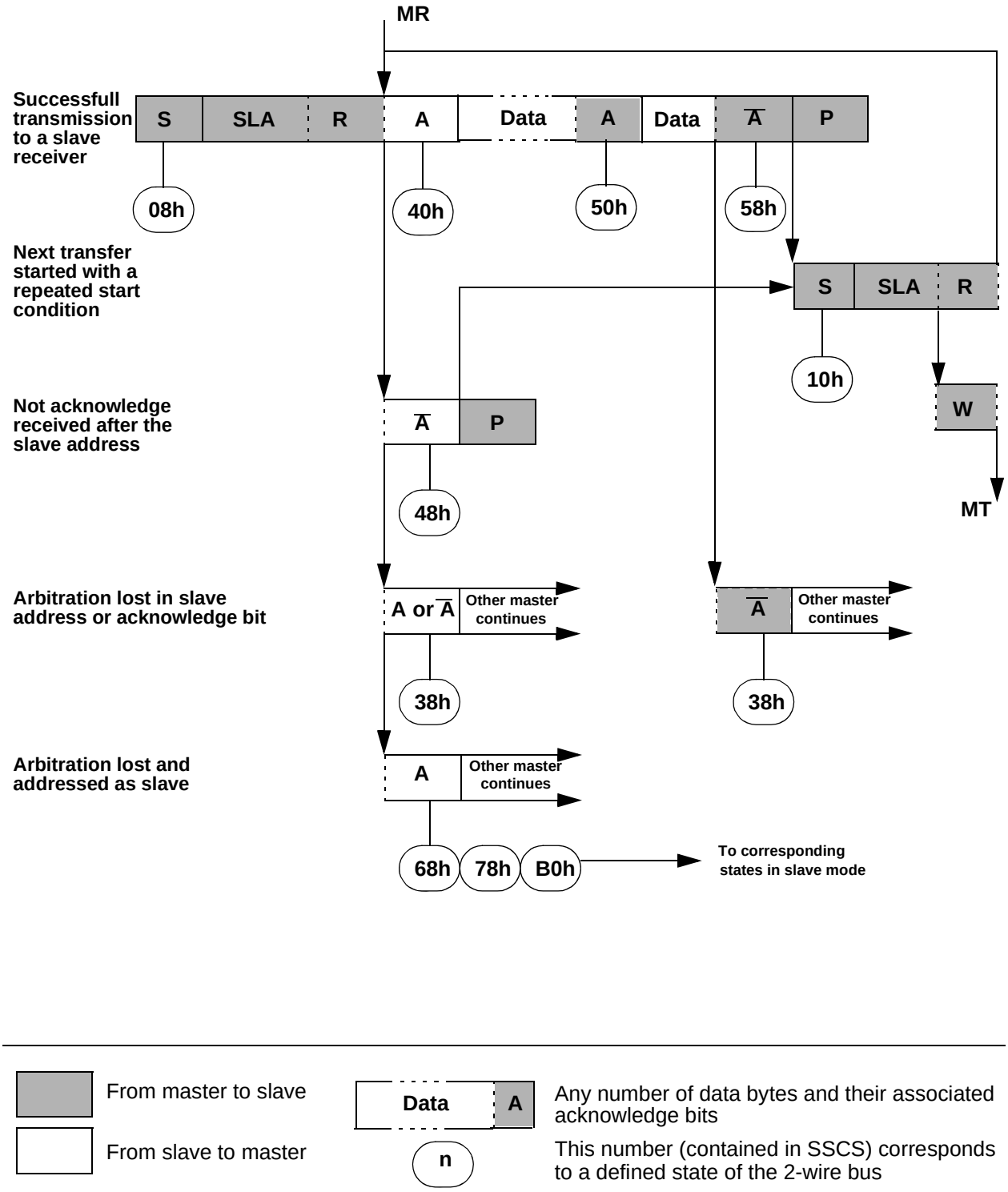


Table 103. Status in Master Receiver Mode

Status Code SSSTA	Status of the Two-wire Bus and Two-wire Hardware	Application software response					Next Action Taken by Two-wire Hardware
		To/From SSDAT	To SSCON				
			SSSTA	SSSTO	SSI	SSAA	
08h	A START condition has been transmitted	Write SLA+R	X	0	0	X	SLA+R will be transmitted.
10h	A repeated START condition has been transmitted	Write SLA+R	X	0	0	X	SLA+R will be transmitted.
		Write SLA+W	X	0	0	X	SLA+W will be transmitted. Logic will switch to master transmitter mode.
38h	Arbitration lost in SLA+R or NOT ACK bit	No SSDAT action	0	0	0	X	Two-wire bus will be released and not addressed slave mode will be entered.
		No SSDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.
40h	SLA+R has been transmitted; ACK has been received	No SSDAT action	0	0	0	0	Data byte will be received and NOT ACK will be returned.
		No SSDAT action	0	0	0	1	Data byte will be received and ACK will be returned.
48h	SLA+R has been transmitted; NOT ACK has been received	No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
50h	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received and NOT ACK will be returned.
		Read data byte	0	0	0	1	Data byte will be received and ACK will be returned.
58h	Data byte has been received; NOT ACK has been returned	Read data byte	1	0	0	X	Repeated START will be transmitted.
		Read data byte	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.

Figure 54. Format and State in the Slave Receiver Mode

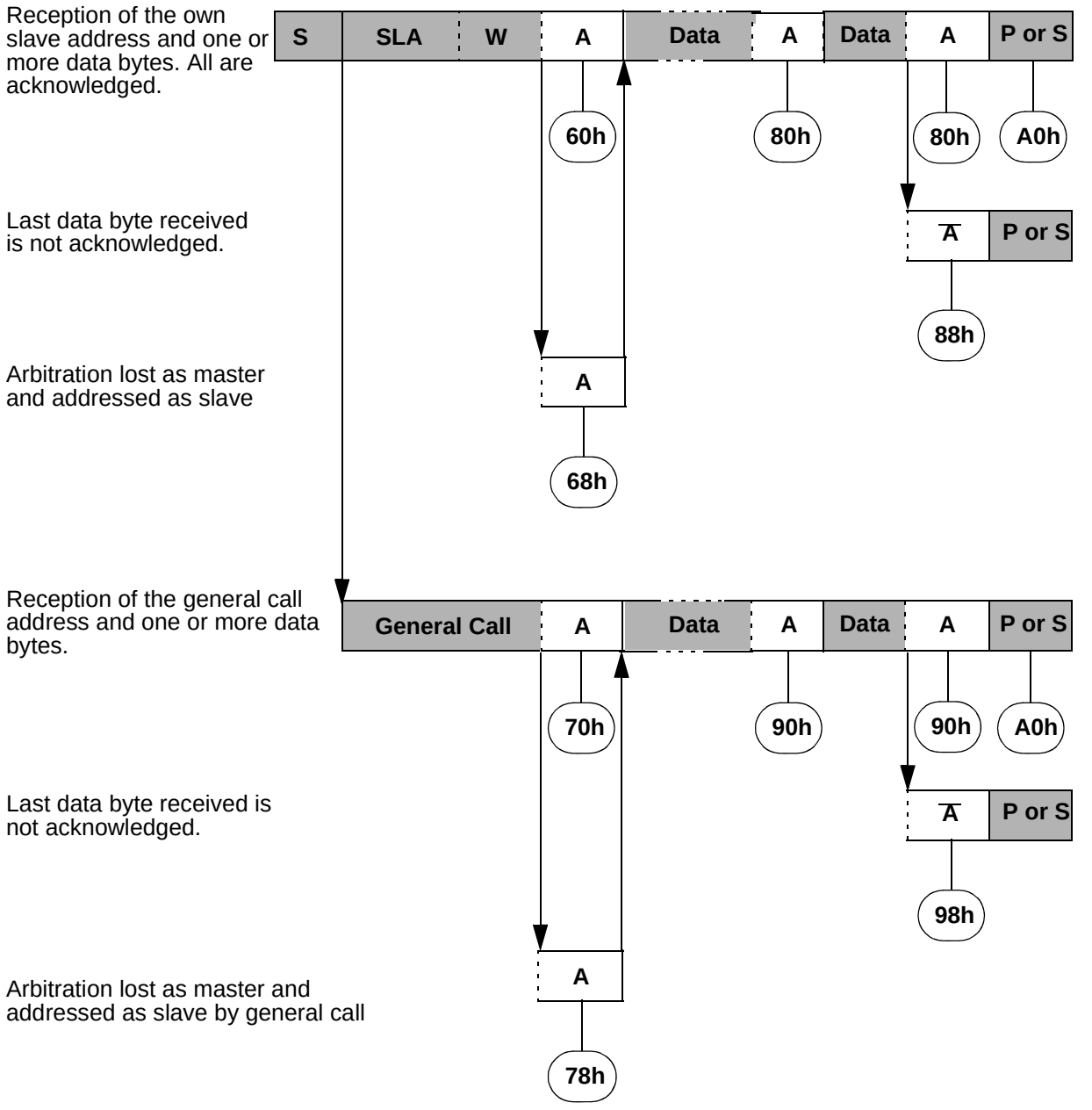


Table 104. Status in Slave Receiver Mode

Status Code (SSCS)	Status of the 2-wire bus and 2-wire hardware	Application Software Response					Next Action Taken By 2-wire Software
		To/from SSDAT	To SSCON				
			STA	STO	SI	AA	
60h	Own SLA+W has been received; ACK has been returned	No SSDAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
68h	Arbitration lost in SLA+R/W as master; own SLA+W has been received; ACK has been returned	No SSDAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
70h	General call address has been received; ACK has been returned	No SSDAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
78h	Arbitration lost in SLA+R/W as master; general call address has been received; ACK has been returned	No SSDAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
80h	Previously addressed with own SLA+W; data has been received; ACK has been returned	No SSDAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
88h	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		Read data byte or	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1
		Read data byte or	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free
90h	Previously addressed with general call; data has been received; ACK has been returned	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned

Table 104. Status in Slave Receiver Mode (Continued)

Status Code (SSCS)	Status of the 2-wire bus and 2-wire hardware	Application Software Response					Next Action Taken By 2-wire Software
		To/from SSDAT	To SSCON				
			STA	STO	SI	AA	
98h	Previously addressed with general call; data has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		Read data byte or	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1
		Read data byte or	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free
A0h	A STOP condition or repeated START condition has been received while still addressed as slave	No SSDAT action or	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action or	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1
		No SSDAT action or	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free

Figure 55. Format and State in the Slave Transmitter Mode

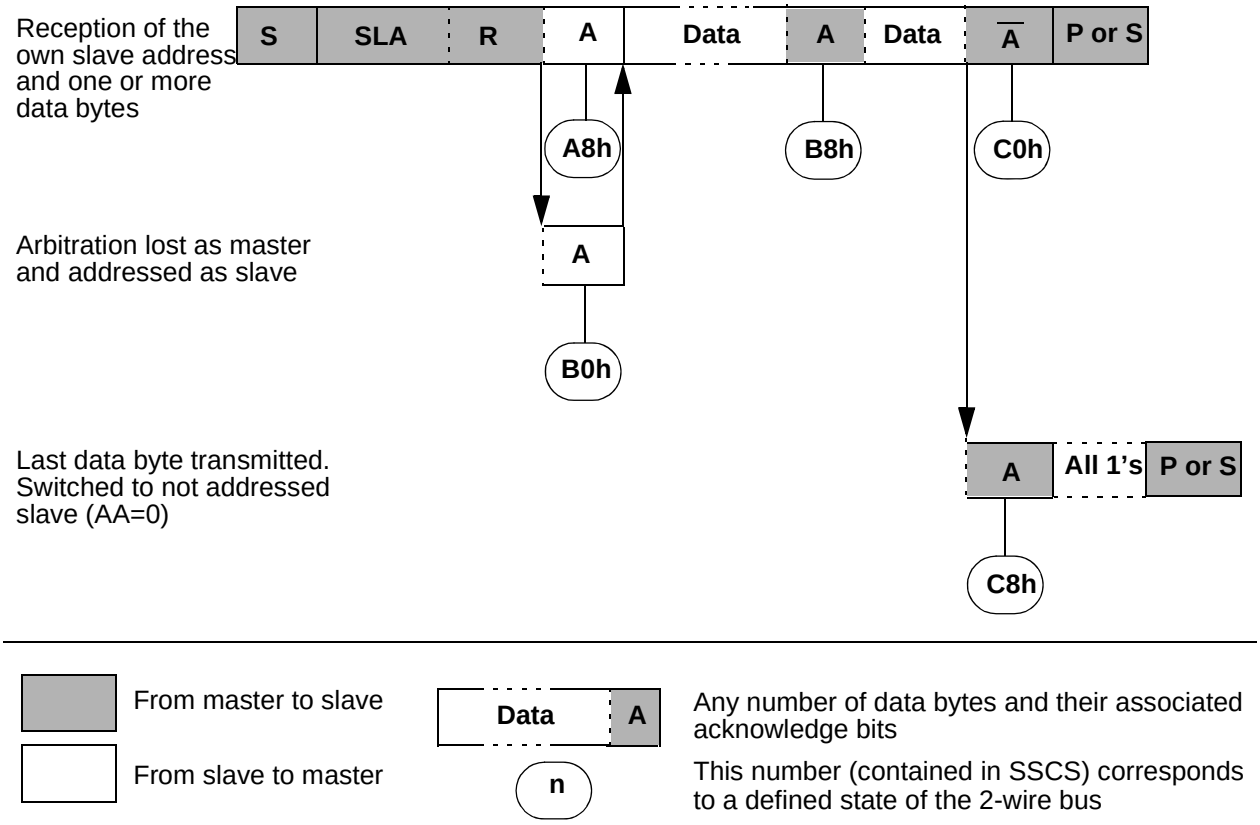


Table 105. Status in Slave Transmitter Mode

Status Code (SSCS)	Status of the 2-wire bus and 2-wire hardware	Application Software Response					Next Action Taken By 2-wire Software
		To/from SSDAT	To SSCON				
			STA	STO	SI	AA	
A8h	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and NOT ACK will be received
		Load data byte	X	0	0	1	Data byte will be transmitted and ACK will be received
B0h	Arbitration lost in SLA+R/W as master; own SLA+R has been received; ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and NOT ACK will be received
		Load data byte	X	0	0	1	Data byte will be transmitted and ACK will be received
B8h	Data byte in SSDAT has been transmitted; NOT ACK has been received	Load data byte or	X	0	0	0	Last data byte will be transmitted and NOT ACK will be received
		Load data byte	X	0	0	1	Data byte will be transmitted and ACK will be received

Table 105. Status in Slave Transmitter Mode (Continued)

Status Code (SSCS)	Status of the 2-wire bus and 2-wire hardware	Application Software Response					Next Action Taken By 2-wire Software
		To/from SSDAT	To SSCON				
			STA	STO	SI	AA	
C0h	Data byte in SSDAT has been transmitted; NOT ACK has been received	No SSDAT action or	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action or	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1
		No SSDAT action or	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free
C8h	Last data byte in SSDAT has been transmitted (AA=0); ACK has been received	No SSDAT action or	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action or	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1
		No SSDAT action or	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free

Table 106. Miscellaneous Status

Status Code (SSCS)	Status of the 2-wire bus and 2-wire hardware	Application Software Response					Next Action Taken By 2-wire Software
		To/from SSDAT	To SSCON				
			STA	STO	SI	AA	
F8h	No relevant state information available; SI= 0	No SSDAT action	No SSCON action				Wait or proceed current transfer
00h	Bus error due to an illegal START or STOP condition	No SSDAT action	0	1	0	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and STO is reset.

Registers

Table 107. SSCON Register

SSCON - Synchronous Serial Control register (93h)

7	6	5	4	3	2	1	0
CR2	SSIE	STA	STO	SI	AA	CR1	CR0

Bit Number	Bit Mnemonic	Description
7	CR2	**Control Rate bit 2** See Table 101.
6	SSIE	**Synchronous Serial Interface Enable bit** Clear to disable the TWI module. Set to enable the TWI module.
5	STA	**Start flag** Set to send a START condition on the bus.
4	ST0	**Stop flag** Set to send a STOP condition on the bus.
3	SI	**Synchronous Serial Interrupt flag** Set by hardware when a serial interrupt is requested. Must be cleared by software to acknowledge interrupt.
2	AA	**Assert Acknowledge flag** Clear in master and slave receiver modes, to force a not acknowledge (high level on SDA). Clear to disable SLA or GCA recognition. Set to recognise SLA or GCA (if GC set) for entering slave receiver or transmitter modes. Set in master and slave receiver modes, to force an acknowledge (low level on SDA). This bit has no effect when in master transmitter mode.
1	CR1	**Control Rate bit 1** See Table 101.
0	CR0	**Control Rate bit 0** See Table 101.

Table 108. SSDAT (95h) - Syncrhonous Serial Data register (read/write)

SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7	SD7	Address bit 7 or Data bit 7.
6	SD6	Address bit 6 or Data bit 6.
5	SD5	Address bit 5 or Data bit 5.
4	SD4	Address bit 4 or Data bit 4.
3	SD3	Address bit 3 or Data bit 3.
2	SD2	Address bit 2 or Data bit 2.

Bit Number	Bit Mnemonic	Description
1	SD1	Address bit 1 or Data bit 1.
0	SD0	Address bit 0 (R/W) or Data bit 0.

Table 109. SSCS (094h) read - Synchronous Serial Control and Status Register

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

Table 110. SSCS Register: Read Mode - Reset Value = F8h

Bit Number	Bit Mnemonic	Description
0	0	Always zero
1	0	Always zero
2	0	Always zero
3	SC0	**Status Code bit 0** See to Table 106.
4	SC1	**Status Code bit 1** See to Table 106.
5	SC2	**Status Code bit 2** See to Table 106.
6	SC3	**Status Code bit 3** See to Table 106.
7	SC4	**Status Code bit 4** See to Table 106.

Table 111. SSADR (096h) - Synchronus Serial Address Register (read/write)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Table 112. SSADR Register - Reset value = FEh

Bit Number	Bit Mnemonic	Description
7	A7	Slave Address bit 7
6	A6	Slave Address bit 6
5	A5	Slave Address bit 5
4	A4	Slave Address bit 4
3	A3	Slave Address bit 3
2	A2	Slave Address bit 2
1	A1	Slave Address bit 1

Bit Number	Bit Mnemonic	Description
0	GC	General Call bit Clear to disable the general call address recognition. Set to enable the general call address recognition.

Serial Port Interface (SPI)

The Serial Peripheral Interface Module (SPI) allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs.

Features

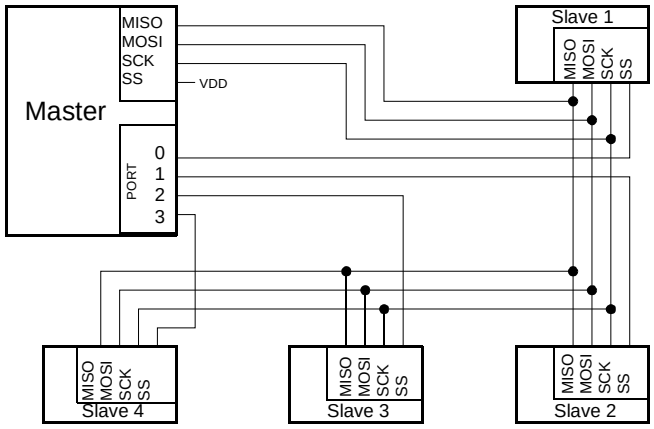
Features of the SPI Module include the following:

- Full-duplex, three-wire synchronous transfers
- Master or Slave operation
- Six programmable Master clock rates in master mode
- Serial clock with programmable polarity and phase
- Master Mode fault error flag with MCU interrupt capability

Signal Description

Figure 56 shows a typical SPI bus configuration using one Master controller and many Slave peripherals. The bus is made of three wires connecting all the devices.

Figure 56. SPI Master/Slaves Interconnection



The Master device selects the individual Slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the Slave devices.

Master Output Slave Input (MOSI)

This 1-bit signal is directly connected between the Master Device and a Slave Device. The MOSI line is used to transfer data in series from the Master to the Slave. Therefore, it is an output signal from the Master, and an input signal to a Slave. A Byte (8-bit word) is transmitted most significant bit (MSB) first, least significant bit (LSB) last.

Master Input Slave Output (MISO)

This 1-bit signal is directly connected between the Slave Device and a Master Device. The MISO line is used to transfer data in series from the Slave to the Master. Therefore, it is an output signal from the Slave, and an input signal to the Master. A Byte (8-bit word) is transmitted most significant bit (MSB) first, least significant bit (LSB) last.

SPI Serial Clock (SCK)

This signal is used to synchronize the data transmission both in and out of the devices through their MOSI and MISO lines. It is driven by the Master for eight clock cycles which allows to exchange one Byte on the serial lines.

Slave Select ($\overline{SS}$)

Each Slave peripheral is selected by one Slave Select pin ($\overline{SS}$). This signal must stay low for any message for a Slave. It is obvious that only one Master ($\overline{SS}$ high level) can drive the network. The Master may select each Slave device by software through port pins (Figure 57). To prevent bus conflicts on the MISO line, only one slave should be selected at a time by the Master for a transmission.

In a Master configuration, the $\overline{SS}$ line can be used in conjunction with the MODF flag in the SPI Status register (SPSCR) to prevent multiple masters from driving MOSI and SCK (see Error conditions).

A high level on the $\overline{SS}$ pin puts the MISO line of a Slave SPI in a high-impedance state.

The $\overline{SS}$ pin could be used as a general-purpose if the following conditions are met:

- The device is configured as a Master and the SSDIS control bit in SPCON is set. This kind of configuration can be found when only one Master is driving the network and there is no way that the $\overline{SS}$ pin could be pulled low. Therefore, the MODF flag in the SPSCR will never be set(1).
- The Device is configured as a Slave with CPHA and SSDIS control bits set(2). This kind of configuration can happen when the system includes one Master and one Slave only. Therefore, the device should always be selected and there is no reason that the Master uses the $\overline{SS}$ pin to select the communicating Slave device.

Note: 1. Clearing SSDIS control bit does not clear MODF.

2. Special care should be taken not to set SSDIS control bit when CPHA ='0' because in this mode, the $\overline{SS}$ is used to start the transmission.

Baud Rate

In Master mode, the baud rate can be selected from a baud rate generator which is controlled by three bits in the SPCON register: SPR2, SPR1 and SPR0.The Master clock is selected from one of seven clock rates resulting from the division of the internal clock by 4, 8, 16, 32, 64 or 128.

Table 113 gives the different clock rates selected by SPR2:SPR1:SPR0.

In Slave mode, the maximum baud rate allowed on the SCK input is limited to $F_{sys}/4$

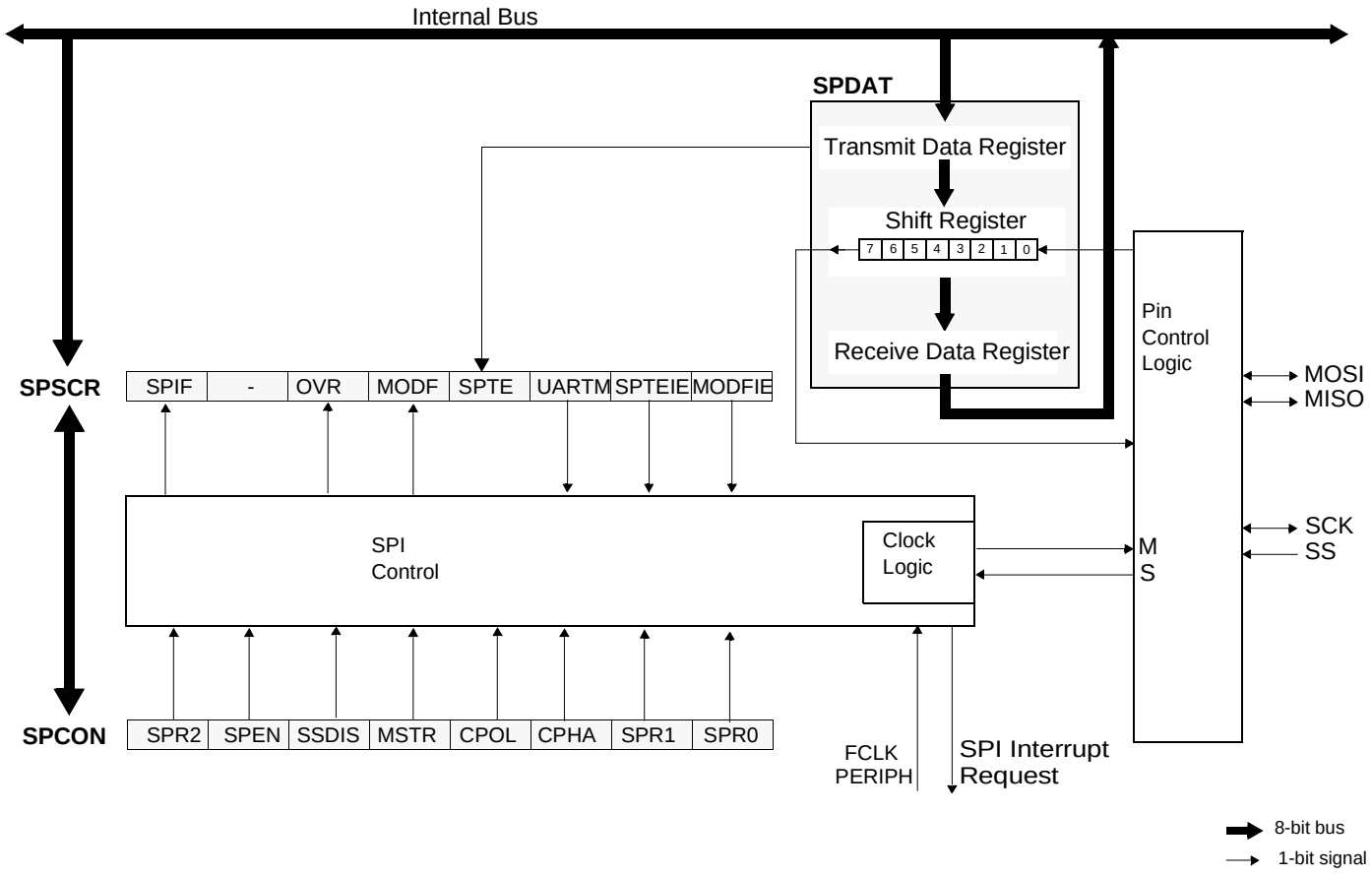
Table 113. SPI Master Baud Rate Selection

SPR2	SPR1	SPR0	Clock Rate	Baud Rate Divisor (BD)
0	0	0	Don't Use	No BRG
0	0	1	$F_{CLK\ PERIPH}$ /4	4
0	1	0	$F_{CLK\ PERIPH}$/8	8
0	1	1	$F_{CLK\ PERIPH}$ /16	16
1	0	0	$F_{CLK\ PERIPH}$ /32	32
1	0	1	$F_{CLK\ PERIPH}$ /64	64
1	1	0	$F_{CLK\ PERIPH}$ /128	128
1	1	1	Don't Use	No BRG

Functional Description

Figure 57 shows a detailed structure of the SPI Module.

Figure 57. SPI Module Block Diagram



Operating Modes

The Serial Peripheral Interface can be configured in one of the two modes: Master mode or Slave mode. The configuration and initialization of the SPI Module is made through two registers:

- The Serial Peripheral Control register (SPCON)
- The Serial Peripheral Status and Control Register (SPSCR)

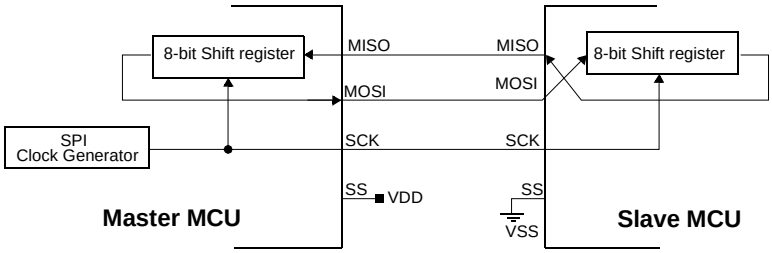
Once the SPI is configured, the data exchange is made using:

- The Serial Peripheral DATa register (SPDAT)

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line (SCK) synchronizes shifting and sampling on the two serial data lines (MOSI and MISO). A Slave Select line ($\overline{SS}$) allows individual selection of a Slave SPI device; Slave devices that are not selected do not interfere with SPI bus activities.

When the Master device transmits data to the Slave device via the MOSI line, the Slave device responds by sending data to the Master device via the MISO line. This implies full-duplex transmission with both data out and data in synchronized with the same clock (Figure 58).

Figure 58. Full-Duplex Master-Slave Interconnection



Master Mode

The SPI operates in Master mode when the Master bit, MSTR (1), in the SPCON register is set. Only one Master SPI device can initiate transmissions. Software begins the transmission from a Master SPI Module by writing to the Serial Peripheral Data Register (SPDAT). If the shift register is empty, the Byte is immediately transferred to the shift register. The Byte begins shifting out on MOSI pin under the control of the serial clock, SCK. Simultaneously, another Byte shifts in from the Slave on the Master's MISO pin. The transmission ends when the Serial Peripheral transfer data flag, SPIF, in SPSCR becomes set. At the same time that SPIF becomes set, the received Byte from the Slave is transferred to the receive data register in SPDAT. Software clears SPIF by reading the Serial Peripheral Status register (SPSCR) with the SPIF bit set, and then reading the SPDAT.

Slave Mode

The SPI operates in Slave mode when the Master bit, MSTR (2), in the SPCON register is cleared. Before a data transmission occurs, the Slave Select pin, $\overline{SS}$, of the Slave device must be set to'0'. $\overline{SS}$ must remain low until the transmission is complete.

In a Slave SPI Module, data enters the shift register under the control of the SCK from the Master SPI Module. After a Byte enters the shift register, it is immediately transferred to the receive data register in SPDAT, and the SPIF bit is set. To prevent an overflow condition, Slave software must then read the SPDAT before another Byte enters the shift register (3). A Slave SPI must complete the write to the SPDAT (shift register) at least one bus cycle before the Master SPI starts a transmission. If the write to the data register is late, the SPI transmits the data already in the shift register from the previous transmission.

Transmission Formats

Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPCON: the Clock Polarity (CPOL (4)) and the Clock Phase (CPHA[4]). CPOL defines the default SCK line level in idle state. It has no significant effect on the transmission format. CPHA defines the edges on which the input data are sampled and the edges on which the output data are shifted (Figure 59 and Figure 60). The clock phase and polarity should be identical for the Master SPI device and the communicating Slave device.

1. The SPI Module should be configured as a Master before it is enabled (SPEN set). Also, the Master SPI should be configured before the Slave SPI.
2. The SPI Module should be configured as a Slave before it is enabled (SPEN set).
3. The maximum frequency of the SCK for an SPI configured as a Slave is the bus clock speed.
4. Before writing to the CPOL and CPHA bits, the SPI should be disabled (SPEN ='0').

Figure 59. Data Transmission Format (CPHA = 0)

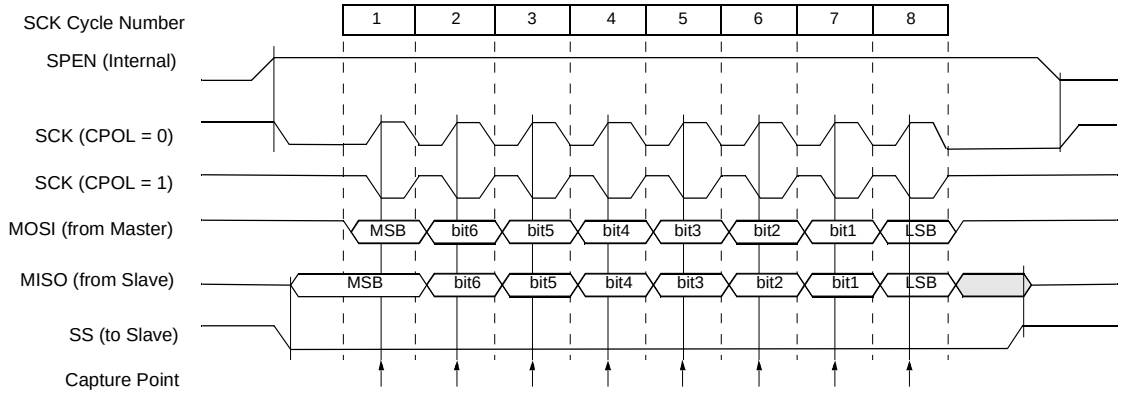


Figure 60. Data Transmission Format (CPHA = 1)

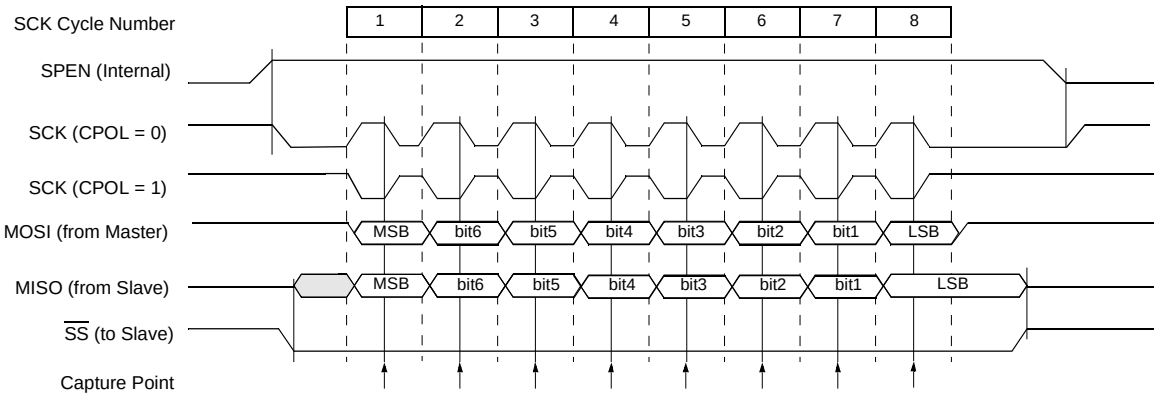
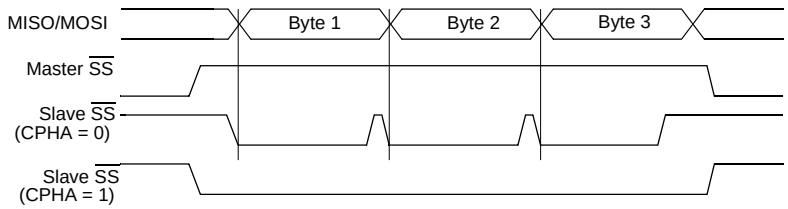


Figure 61. CPHA/SS Timing



As shown in Figure 59, the first SCK edge is the MSB capture strobe. Therefore, the Slave must begin driving its data before the first SCK edge, and a falling edge on the SS pin is used to start the transmission. The SS pin must be toggled high and then low between each Byte transmitted (Figure 61).

Figure 60 shows an SPI transmission in which CPHA is'1'. In this case, the Master begins driving its MOSI pin on the first SCK edge. Therefore, the Slave uses the first SCK edge as a start transmission signal. The SS pin can remain low between transmissions (Figure 61). This format may be preferred in systems having only one Master and only one Slave driving the MISO data line.

Queuing transmission

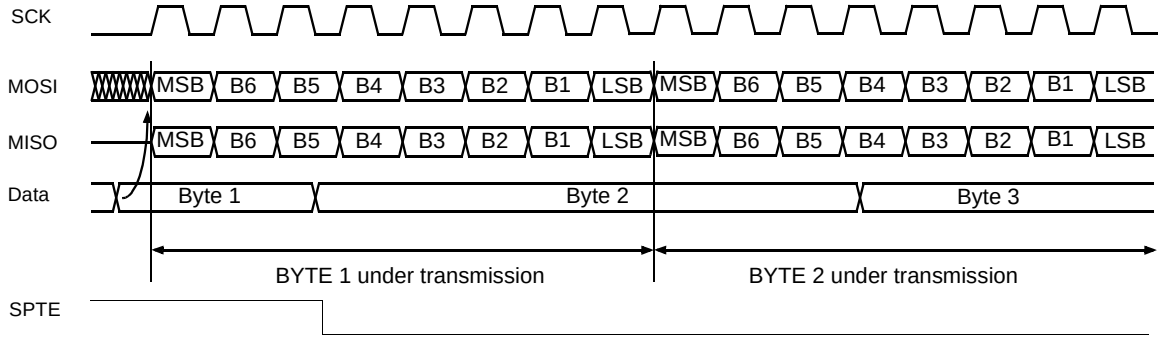
For an SPI configured in master or slave mode, a queued data byte must be transmitted/received immediately after the previous transmission has completed.

When a transmission is in progress a new data can be queued and sent as soon as transmission has been completed. So it is possible to transmit bytes without latency, useful in some applications.

The SPTE bit in SPSCR is set as long as the transmission buffer is free. It means that the user application can write SPDAT with the data to be transmitted until the SPTE becomes cleared.

Figure 62 shows a queuing transmission in master mode. Once the Byte 1 is ready, it is immediately sent on the bus. Meanwhile an other byte is prepared (and the SPTE is cleared), it will be sent at the end of the current transmission. The next data must be ready before the end of the current transmission.

Figure 62. Queuing Transmission In Master Mode



In slave mode it is almost the same except it is the external master that start the transmission.

Also, in slave mode, if no new data is ready, the last value received will be the next data byte transmitted.

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.

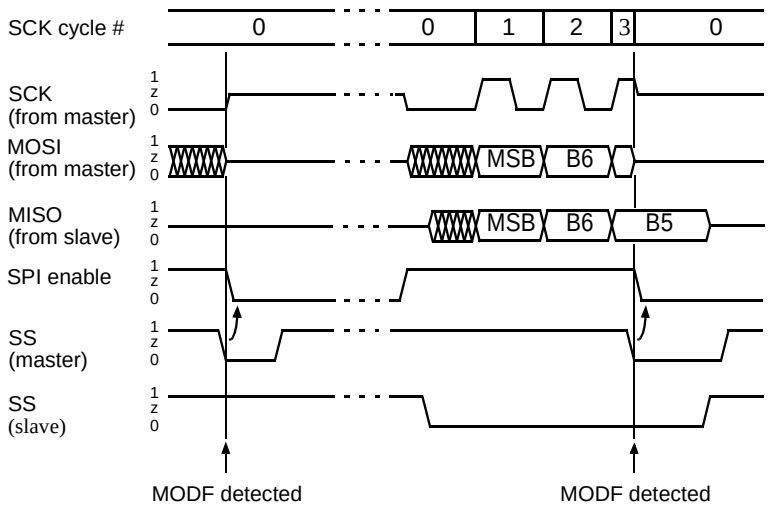
- Mode fault detection in Master mode:

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 63. Mode Fault Conditions in Master Mode (Cpha ='1'/Cpol ='0')



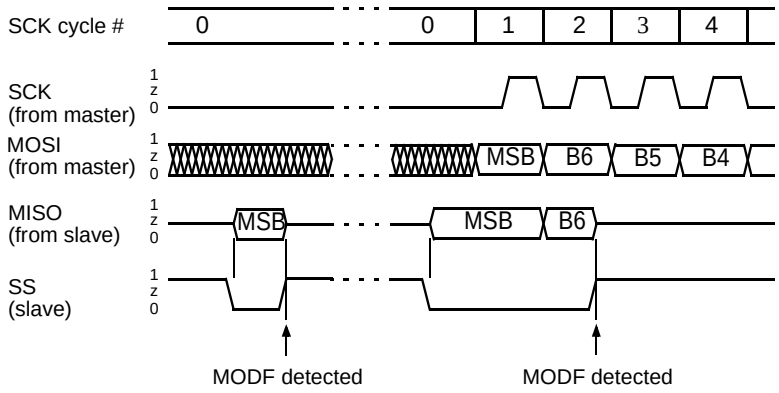
Note: When SS is discarded (SS disabled) it is not possible to detect a MODF error in master mode because the SPI is internally unselected and the SS pin is a general purpose I/O.

- 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.

Figure 64. Mode Fault Conditions in Slave Mode



Note: when SS is discarded (SS disabled) it is not possible to detect a MODF error in slave mode because the SPI is internally selected. Also the SS pin becomes a general purpose I/O.

OverRun Condition

This error mean that the speed is not adapted for the running application:

An OverRun condition occurs when a byte has been received whereas the previous one has not been read by the application yet.
The last byte (which generate the overrun error) does not overwrite the unread data so that it can still be read. Therefore, an overrun error always indicates the loss of data.

Interrupts

Three SPI status flags can generate a CPU interrupt requests:

Table 114. SPI Interrupts

Flag	Request
SPIF (SPI data transfer)	SPI Transmitter Interrupt Request
MODF (Mode Fault)	SPI mode-fault Interrupt Request
SPTE (Transmit register empty)	SPI transmit register empty Interrupt Request

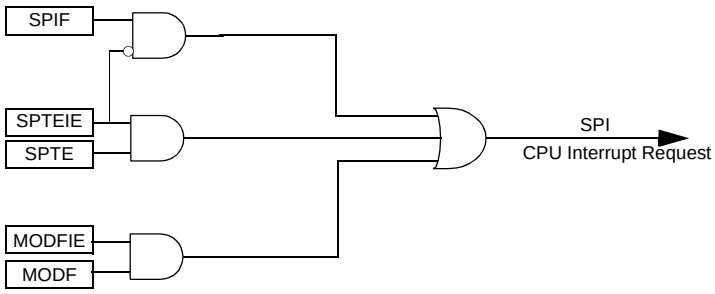
Serial Peripheral data transfer flag, SPIF: This bit is set by hardware when a transfer has been completed. SPIF bit generates transmitter CPU interrupt request only when SPTEIE is disabled.

Mode Fault flag, MODF: This bit is set to indicate that the level on the $\overline{SS}$ is inconsistent with the mode of the SPI (in both master and slave modes).

Serial Peripheral Transmit Register empty flag, SPTE: This bit is set when the transmit buffer is empty (other data can be loaded is SPDAT). SPTE bit generates transmitter CPU interrupt request only when SPTEIE is enabled.

Note: While using SPTE interruption for “burst mode” transfers (SPTEIE=’1’), the user software application should take care to clear SPTEIE, during the last but one data reception (to be able to generate an interrupt on SPIF flag at the end of the last data reception).

Figure 65. SPI Interrupt Requests Generation



Registers

Three registers in the SPI module provide control, status and data storage functions. These registers are describe in the following paragraphs.

Serial Peripheral Control Register (SPCON)

- The Serial Peripheral Control Register does the following:
- Selects one of the Master clock rates
- Configure the SPI Module as Master or Slave
- Selects serial clock polarity and phase
- Enables the SPI Module
- Frees the SS pin for a general-purpose

Table 115 describes this register and explains the use of each bit

Table 115. SPCON Register

SPCON - Serial Peripheral Control Register (0D4H)

7	6	5	4	3	2	1	0
SPR2	**SPEN**	**SSDIS**	**MSTR**	**CPOL**	**CPHA**	**SPR1**	**SPR0**

Bit Number	Bit Mnemonic	Description
7	SPR2	**Serial Peripheral Rate 2** Bit with SPR1 and SPR0 define the clock rate (See bits SPR1 and SPR0 for detail).
6	SPEN	**Serial Peripheral Enable** Cleared to disable the SPI interface (internal reset of the SPI). Set to enable the SPI interface.
5	SSDIS	**$\overline{SS}$ Disable** Cleared to enable $\overline{SS}$ in both Master and Slave modes. Set to disable $\overline{SS}$ in both Master and Slave modes. In Slave mode, this bit has no effect if CPHA ='0'. When SSDIS is set, no MODF interrupt request is generated.
4	MSTR	**Serial Peripheral Master** Cleared to configure the SPI as a Slave. Set to configure the SPI as a Master.

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 116. SPSCR Register

SPSCR - Serial Peripheral Status and Control register (C4H)

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 117) 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 117. SPDAT Register

SPDAT - Serial Peripheral Data Register (C5H)

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.

Power Monitor

The POR/PFD function monitors the internal power-supply of the CPU core memories and the peripherals, and if needed, suspends their activity when the internal power supply falls below a safety threshold. This is achieved by applying an internal reset to them.

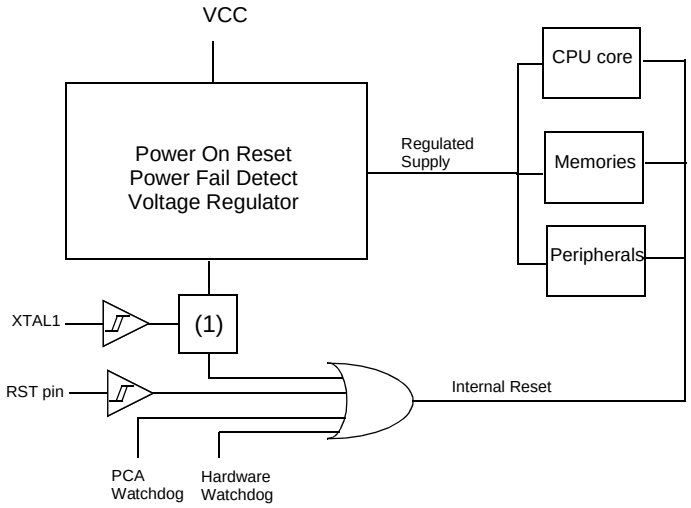
By generating the Reset the Power Monitor insures a correct start up when AT89C51RE2 is powered up.

Description

In order to startup and maintain the microcontroller in correct operating mode, V_{CC} has to be stabilized in the V_{CC} operating range and the oscillator has to be stabilized with a nominal amplitude compatible with logic level VIH/VIL.

These parameters are controlled during the three phases: power-up, normal operation and power going down. See Figure 66.

Figure 66. Power Monitor Block Diagram

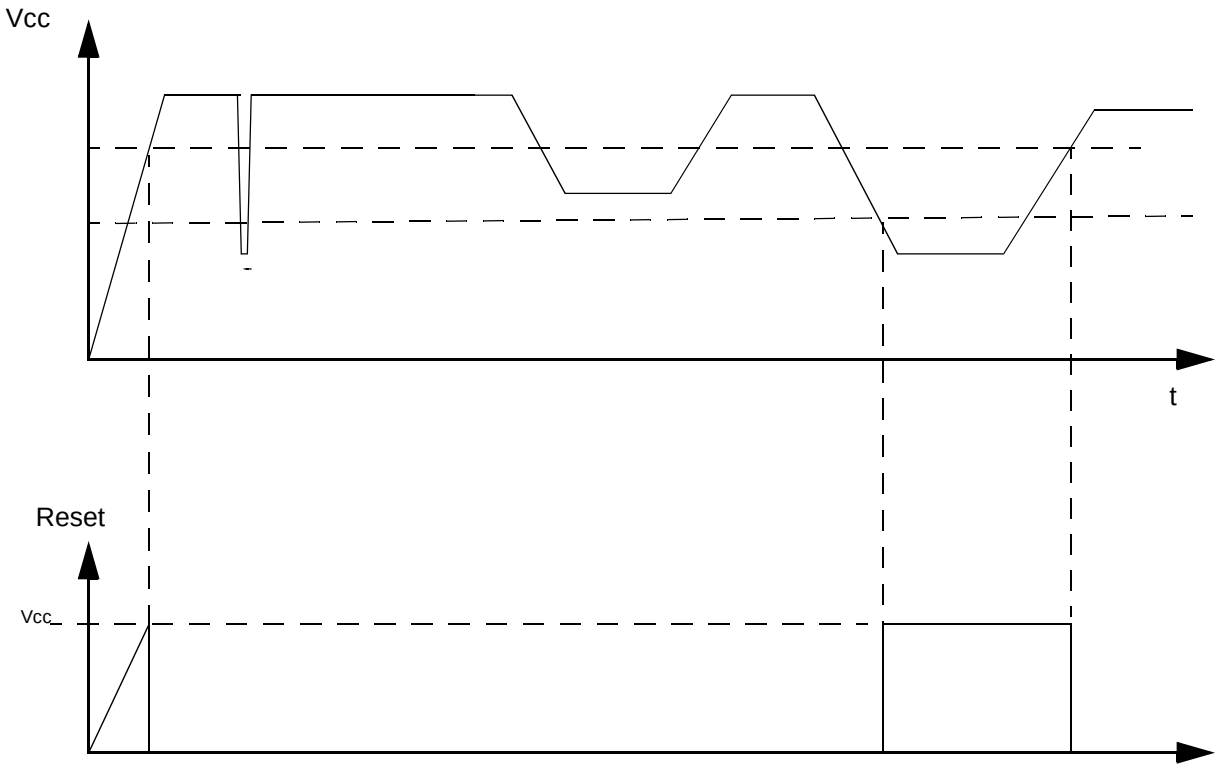


Note: 1. Once XTAL1 High and low levels reach above and below VIH/VIL. a 1024 clock period delay will extend the reset coming from the Power Fail Detect. If the power falls below the Power Fail Detect threshold level, the Reset will be applied immediately.

The Voltage regulator generates a regulated internal supply for the CPU core the memories and the peripherals. Spikes on the external Vcc are smoothed by the voltage regulator.

The Power fail detect monitor the supply generated by the voltage regulator and generate a reset if this supply falls below a safety threshold as illustrated in the Figure 67 below.

Figure 67. Power Fail Detect



When the power is applied, the Power Monitor immediately asserts a reset. Once the internal supply after the voltage regulator reach a safety level, the power monitor then looks at the XTAL clock input. The internal reset will remain asserted until the Xtal1 levels are above and below VIH and VIL. Further more. An internal counter will count 1024 clock periods before the reset is de-asserted.

If the internal power supply falls below a safety level, a reset is immediately asserted.

Power-off Flag

The power-off flag allows the user to distinguish between a “cold start” reset and a “warm start” reset.

A cold start reset is the one induced by V_{CC} switch-on. A warm start reset occurs while V_{CC} is still applied to the device and could be generated for example by an exit from power-down.

The power-off flag (POF) is located in PCON register (Table 118). POF is set by hardware when V_{CC} rises from 0 to its nominal voltage. The POF can be set or cleared by software allowing the user to determine the type of reset.

Table 118. PCON Register

PCON - Power Control Register (87h)

7	6	5	4	3	2	1	0
SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial port Mode bit 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	**Serial port Mode bit 0** Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-Off Flag** Cleared to recognize next reset type. Set by hardware when V_{CC} rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
2	GF0	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
1	PD	**Power-Down mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle mode bit** Cleared by hardware when interrupt or reset occurs. Set to enter idle mode.

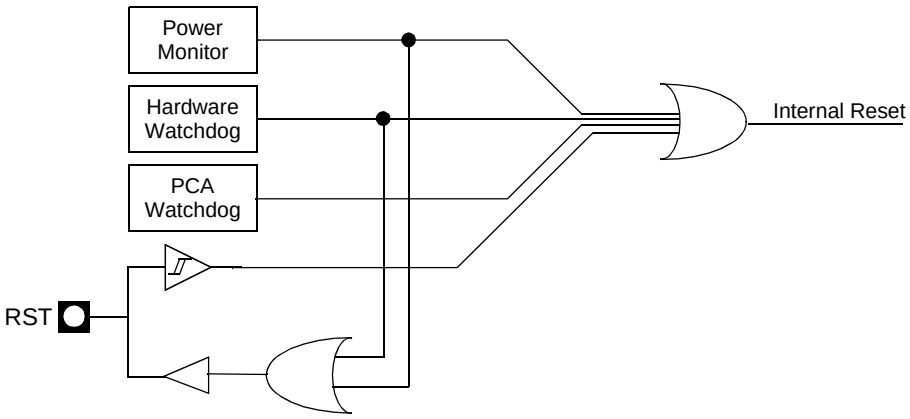
Reset Value = 00X1 0000b
Not bit addressable

Reset

Introduction

The reset sources are: Power Management, Hardware Watchdog, PCA Watchdog and Reset input.

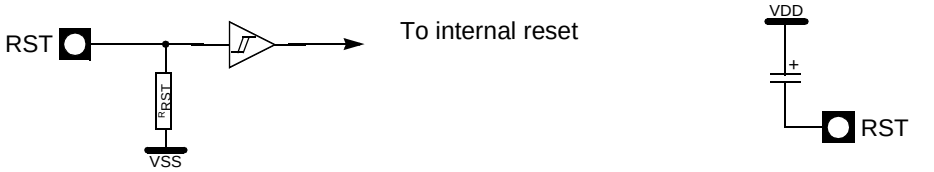
Figure 68. Reset schematic



Reset Input

The Reset input can be used to force a reset pulse longer than the internal reset controlled by the Power Monitor. RST input has a pull-down resistor allowing power-on reset by simply connecting an external capacitor to V_{CC} as shown in Figure 69. Resistor value and input characteristics are discussed in the Section “DC Characteristics” of the AT89C51RE2 datasheet.

Figure 69. Reset Circuitry and Power-On Reset



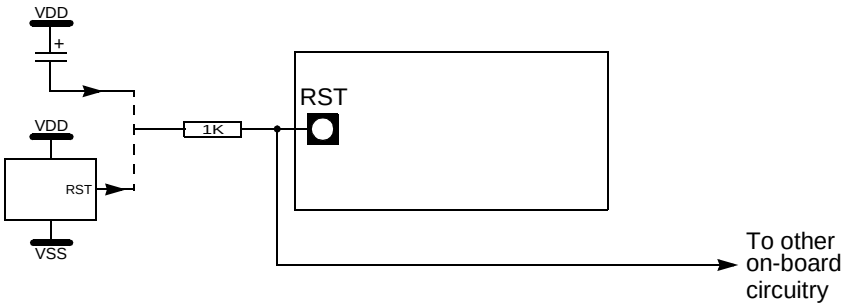
a. RST input circuitry

b. Power-on Reset

Reset Output

As detailed in Section “Hardware Watchdog Timer”, page 124, the WDT generates a 96-clock period pulse on the RST pin. In order to properly propagate this pulse to the rest of the application in case of external capacitor or power-supply supervisor circuit, a 1 kΩ resistor must be added as shown Figure 70.

Figure 70. Recommended Reset Output Schematic



Electrical Characteristics

Absolute Maximum Ratings

I = industrial	-40°C to 85°C
Storage Temperature	-65°C to + 150°C
Voltage on V_{CC} to V_{SS} (standard voltage)	-0.5V to + 6.5V
Voltage on V_{CC} to V_{SS} (low voltage)	-0.5V to + 4.5V
Voltage on Any Pin to V_{SS}	-0.5V to V_{CC} + 0.5V
Power Dissipation	1 W[2]

Note: Stresses at or 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 other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.
Power dissipation is based on the maximum allowable die temperature and the thermal resistance of the package.

DC Parameters

T_A = -40°C to +85°C; V_{SS} = 0V; V_{CC} =2.7V to 5.5V; F = 0 to 40 MHz

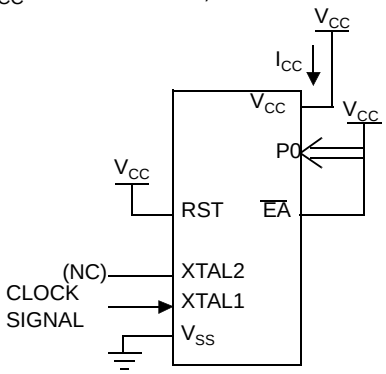
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5		0.2 V_{CC} - 0.1	V	
V_{IH}	Input High Voltage except RST, XTAL1	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage RST, XTAL1	0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage, ports 1, 2, 3, 4 [6]			0.3 0.45 1.0	V V V	V_{CC} = 4.5V to 5.5V I_{OL} = 100 µA[4] I_{OL} = 1.6 mA[4] I_{OL} = 3.5 mA[4]
				0.45	V	V_{CC} = 2.7V to 5.5V I_{OL} = 0.8 mA[4]
V_{OL1}	Output Low Voltage, port 0, ALE, $\overline{PSEN}$ [6]			0.3 0.45 1.0	V V V	V_{CC} = 4.5V to 5.5V I_{OL} = 200 µA[4] I_{OL} = 3.2 mA[4] I_{OL} = 7.0 mA[4]
				0.45	V	V_{CC} = 2.7V to 5.5V I_{OL} = 1.6 mA[4]
V_{OH}	Output High Voltage, ports 1, 2, 3, 4	V_{CC} - 0.3 V_{CC} - 0.7 V_{CC} - 1.5			V V V	V_{CC} = 5V ± 10% I_{OH} = -10 µA I_{OH} = -30 µA I_{OH} = -60 µA
		0.9 V_{CC}			V	V_{CC} = 2.7V to 5.5V I_{OH} = -10 µA
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	V_{CC} = 5V ± 10% I_{OH} = -200 µA I_{OH} = -3.2 mA I_{OH} = -7.0 mA
		0.9 V_{CC}			V	V_{CC} = 2.7V to 5.5V I_{OH} = -10 µA

T_A = -40°C to +85°C; V_{SS} = 0V; V_{CC} =2.7V to 5.5V; F = 0 to 40 MHz (Continued)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
R_{RST}	RST Pull-down Resistor	50	200(5)	250	kΩ	
I_{IL}	Logical 0 Input Current ports 1, 2, 3, 4 and 5			-50	µA	V_{IN} = 0.45V
I_{LI}	Input Leakage Current			±10	µA	0.45V < V_{IN} < V_{CC}
I_{TL}	Logical 1 to 0 Transition Current, ports 1, 2, 3, 4			-650	µA	V_{IN} = 2.0V
C_{IO}	Capacitance of I/O Buffer			10	pF	F_C = 3 MHz T_A = 25°C
I_{PD}	Power-down Current		75	150	µA	2.7 < V_{CC} < 5.5V(3)
I_{CCOP}	Power Supply Current on normal mode			0.4 x Frequency (MHz) + 5	mA	V_{CC} = 5.5V(1)
I_{CCIDLE}	Power Supply Current on idle mode			0.3 x Frequency (MHz) + 5	mA	V_{CC} = 5.5V(2)
$I_{CCWRITE}$	Power Supply Current on flash write			0.8 x Frequency (MHz) + 15	mA	V_{CC} = 5.5V
t_{WRITE}	Flash programming time		7	10	ms	2.7 < V_{CC} < 5.5V

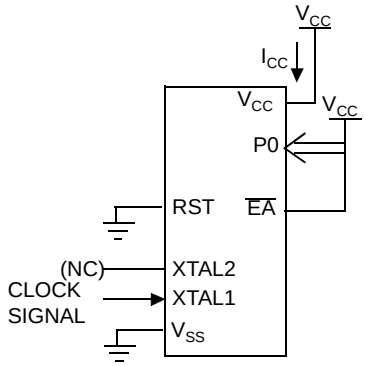
Notes:
1. Operating I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns (see Figure 74), V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} - 0.5V; XTAL2 N.C.; $\overline{EA}$ = RST = Port 0 = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator used (see Figure 71).
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 72).
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 73).
4. Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} 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 100 pF), 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. Typical values are based on a limited number of samples and are not guaranteed. The values listed are at room temperature and 5V.
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 and 3: 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.

Figure 71. I_{CC} Test Condition, Active Mode



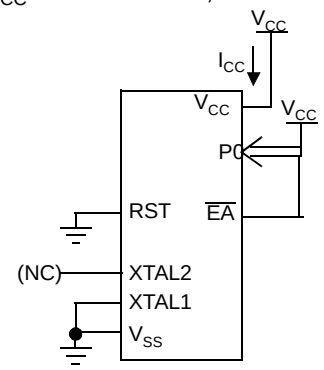
All other pins are disconnected.

Figure 72. I_{CC} Test Condition, Idle Mode



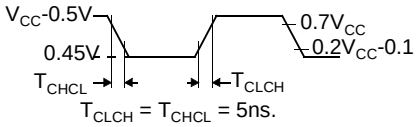
All other pins are disconnected.

Figure 73. I_{CC} Test Condition, Power-down Mode



All other pins are disconnected.

Figure 74. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes



AC Parameters

Explanation of the AC Symbols

Each timing symbol has 5 characters. The first character is always a “T” (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example:T_{AVLL} = Time for Address Valid to ALE Low.
T_{LLPL} = Time for ALE Low to $\overline{PSEN}$ Low.

(Load Capacitance for port 0, ALE and PSEN = 100 pF; Load Capacitance for all other outputs = 80 pF.)

Table 119 Table 122, and Table 125 give the description of each AC symbols.

Table 120, Table 121, Table 123 and Table 126 gives the range for each AC parameter.

Table 120, Table 121 and Table 127 give the frequency derating formula of the AC parameter for each speed range description. To calculate each AC symbols. Take the x value in the corresponding column (-M or -L) and use this value in the formula.

Example: T_{LLIU} for -M and 20 MHz, Standard clock.
x = 35 ns
T 50 ns
T_{CCIV} = 4T - x = 165 ns

External Program Memory Characteristics

Table 119. 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 120. AC Parameters for a Fix Clock

Symbol	-M(1)		-L(2)		Units
	Min	Max	Min	Max	
T	25		25		ns
T_{LHLL}	35		35		ns
T_{AVLL}	5		5		ns
T_{LLAX}	5		5		ns
T_{LLIV}		n 65		65	ns
T_{LLPL}	5		5		ns
T_{PLPH}	50		50		ns
T_{PLIV}		30		30	ns
T_{PXIX}	0		0		ns
T_{PXIZ}		10		10	ns
T_{AVIV}		80		80	ns
T_{PLAZ}		10		10	ns

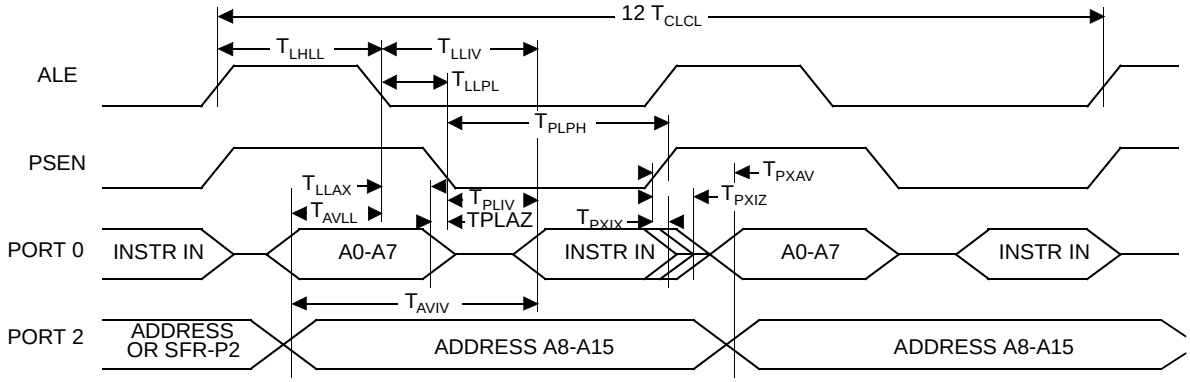
Notes: 1. ‘ -L ‘ refers to 2V - 5.5V version.
2. ‘ -M ’ refers to 4.5V to 5.5V version.

Table 121. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X parameter for -M(1) range	X parameter for -L(2) range	Units
T_{LHLL}	Min	2 T - x	T - x	15	15	ns
T_{AVLL}	Min	T - x	0.5 T - x	20	20	ns
T_{LLAX}	Min	T - x	0.5 T - x	20	20	ns
T_{LLIV}	Max	4 T - x	2 T - x	35	35	ns
T_{LLPL}	Min	T - x	0.5 T - x	15	15	ns
T_{PLPH}	Min	3 T - x	1.5 T - x	25	25	ns
T_{PLIV}	Max	3 T - x	1.5 T - x	45	45	ns
T_{PXIX}	Min	x	x	0	0	ns
T_{PXIZ}	Max	T - x	0.5 T - x	15	15	ns
T_{AVIV}	Max	5 T - x	2.5 T - x	45	45	ns
T_{PLAZ}	Max	x	x	10	10	ns

Notes: 1. ‘ -L ‘ refers to 2V - 5.5V version.
2. ‘ -M ’ refers to 4.5V to 5.5V version.

External Program Memory Read Cycle



External Data Memory Characteristics

Table 122. Symbol Description

Symbol	Parameter
T_{RLRH}	$\overline{RD}$ Pulse Width
T_{WLWH}	$\overline{WR}$ Pulse Width
T_{RLDV}	$\overline{RD}$ to Valid Data In
T_{RHDX}	Data Hold After $\overline{RD}$
T_{RHDZ}	Data Float After $\overline{RD}$
T_{LLDV}	ALE to Valid Data In
T_{AVDV}	Address to Valid Data In
T_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$
T_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$
T_{QVWX}	Data Valid to $\overline{WR}$ Transition
T_{QVWH}	Data Set-up to $\overline{WR}$ High
T_{WHQX}	Data Hold After $\overline{WR}$
T_{RLAZ}	$\overline{RD}$ Low to Address Float
T_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE high

Table 123. AC Parameters for a Fix Clock

Symbol	-M[1]		-L[2]		Units
	Min	Max	Min	Max	
T_{RLRH}	125		125		ns
T_{WLWH}	125		125		ns
T_{RLDV}		95		95	ns
T_{RHDX}	0		0		ns
T_{RHDZ}		25		25	ns
T_{LLDV}		155		155	ns
T_{AVDV}		160		160	ns
T_{LLWL}	45	105	45	105	ns
T_{AVWL}	70		70		ns
T_{QVWX}	5		5		ns
T_{QVWH}	155		155		ns
T_{WHQX}	10		10		ns
T_{RLAZ}	0		0		ns
T_{WHLH}	5	45	5	45	ns

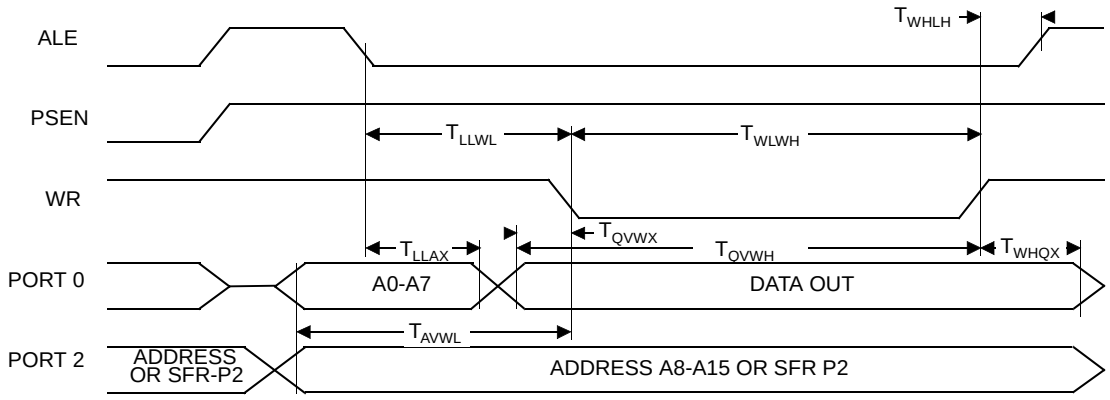
Notes: 1. ‘ -L ‘ refers to 2V - 5.5V version.
2. ‘ -M ’ refers to 4.5V to 5.5V version.

Table 124. AC Parameters for a Variable Clock

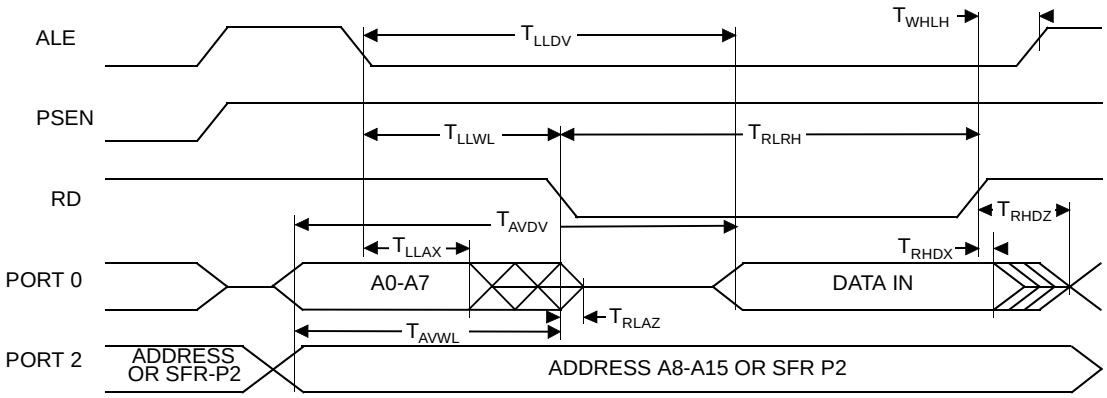
Symbol	Type	Standard Clock	X2 Clock	X parameter for -M(1) range	X parameter for -L(2) range	Units
T_{RLRH}	Min	6 T - x	3 T - x	25	25	ns
T_{WLWH}	Min	6 T - x	3 T - x	25	25	ns
T_{RLDV}	Max	5 T - x	2.5 T - x	30	30	ns
T_{RHDX}	Min	x	x	0	0	ns
T_{RHDZ}	Max	2 T - x	T - x	25	25	ns
T_{LLDV}	Max	8 T - x	4T -x	45	45	ns
T_{AVDV}	Max	9 T - x	4.5 T - x	65	65	ns
T_{LLWL}	Min	3 T - x	1.5 T - x	30	30	ns
T_{LLWL}	Max	3 T + x	1.5 T + x	30	30	ns
T_{AVWL}	Min	4 T - x	2 T - x	30	30	ns
T_{QVWX}	Min	T - x	0.5 T - x	20	20	ns
T_{QVWH}	Min	7 T - x	3.5 T - x	20	20	ns
T_{WHQX}	Min	T - x	0.5 T - x	15	15	ns
T_{RLAZ}	Max	x	x	0	0	ns
T_{WHLH}	Min	T - x	0.5 T - x	20	20	ns
T_{WHLH}	Max	T + x	0.5 T + x	20	20	ns

Notes: 1. ' -L ' refers to 2V - 5.5V version.
2. ' -M ' refers to 4.5V to 5.5V version.

External Data Memory Write Cycle



External Data Memory Read Cycle



Serial Port Timing - Shift Register Mode

Table 125. Symbol Description

Symbol	Parameter
T_{XLXL}	Serial port clock cycle time
T_{QVHX}	Output data set-up to clock rising edge
T_{XHQX}	Output data hold after clock rising edge
T_{XHDX}	Input data hold after clock rising edge
T_{XHDV}	Clock rising edge to input data valid

Table 126. AC Parameters for a Fix Clock

Symbol	-M(1)		-L(2)		Units
	Min	Max	Min	Max	
T_{XLXL}	300		300		ns
T_{QVHX}	200		200		ns
T_{XHQX}	30		30		ns
T_{XHDX}	0		0		ns
T_{XHDV}		117		117	ns

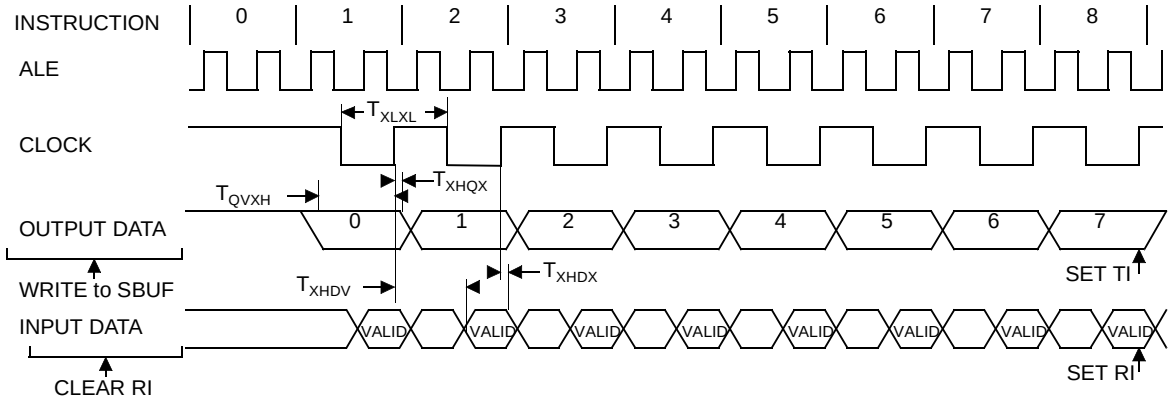
Notes: 1. ‘ -L ‘ refers to 2V - 5.5V version.
2. ‘ -M ’ refers to 4.5V to 5.5V version.

Table 127. AC Parameters for a Variable Clock

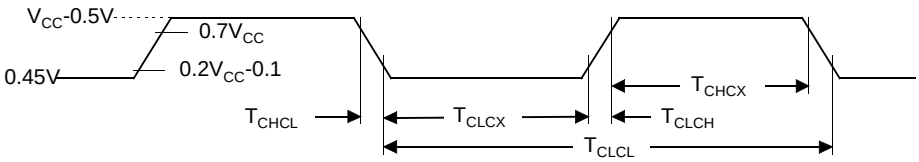
Symbol	Type	Standard Clock	X2 Clock	X Parameter For -M(1) Range	X Parameter For -L(2) Range	Units
T_{XLXL}	Min	12 T	6 T			ns
T_{QVHX}	Min	10 T - x	5 T - x	50	50	ns
T_{XHQX}	Min	2 T - x	T - x	20	20	ns
T_{XHDX}	Min	x	x	0	0	ns
T_{XHDV}	Max	10 T - x	5 T- x	133	133	ns

Notes: 1. ' -L ' refers to 2V - 5.5V version.
2. ' -M ' refers to 4.5V to 5.5V version.

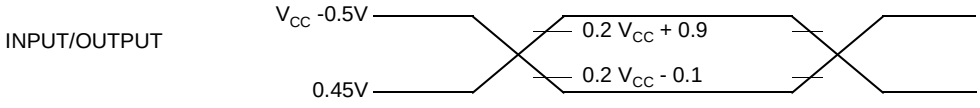
Shift Register Timing Waveforms



External Clock Drive Waveforms

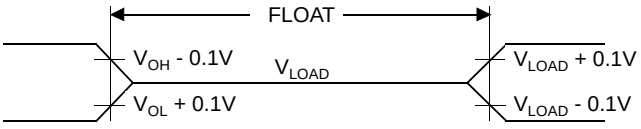


AC Testing Input/Output Waveforms



AC inputs during testing are driven at V_{CC} - 0.5 for a logic "1" and 0.45V for a logic "0". Timing measurement are made at V_{IH} min. for a logic "1" and V_{IL} max for a logic "0".

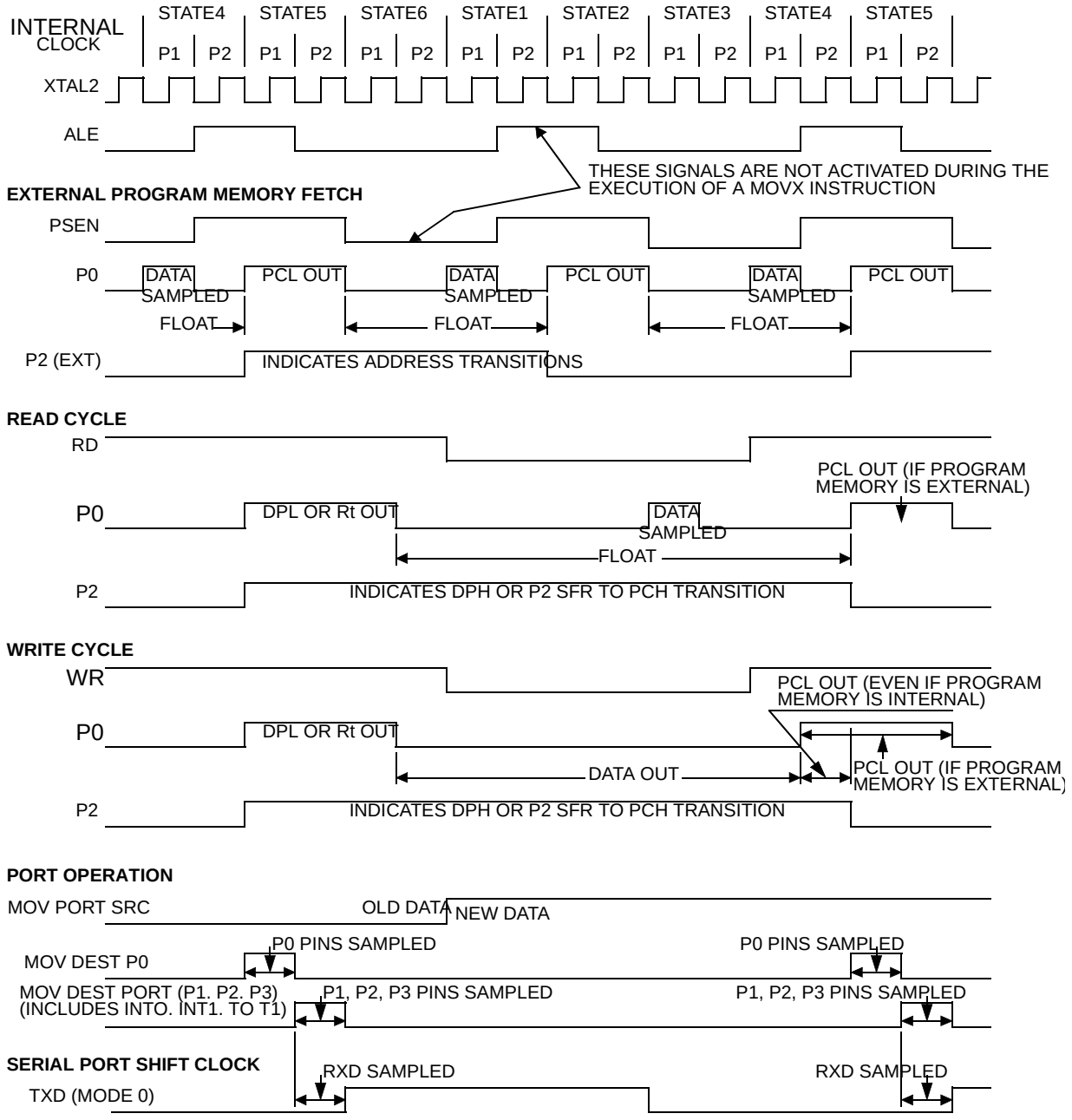
Float Waveforms



For timing purposes as port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

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

Figure 75. Internal Clock Signals



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 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

Flash Memory

Table 128. Timing Symbol Definitions

Signals	
S (Hardware condition)	PSEN#,EA
R	RST
B	FBUSY flag

Conditions	
L	Low
V	Valid
X	No Longer Valid

Table 129. Memory AC Timing

VDD = 3V to 5.5V, TA = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
T_{SVRL}	Input PSEN# Valid to RST Edge	50			ns
T_{RLSX}	Input PSEN# Hold after RST Edge	50			ns
T_{BHBL}	Flash Internal Busy (Programming) Time		10		ms
N_{FCY}	Number of Flash Erase/Write Cycles	100 000			cycles
T_{FDR}	Flash Retention Time	10			years

Figure 76. Flash Memory – ISP Waveforms

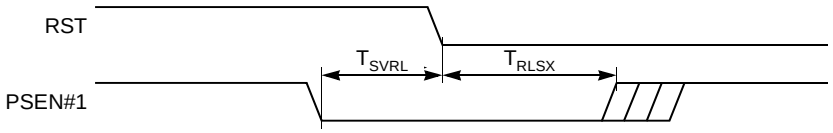
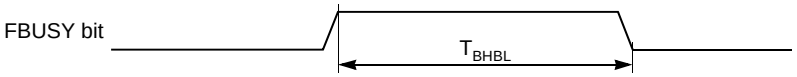


Figure 77. Flash Memory – Internal Busy Waveforms



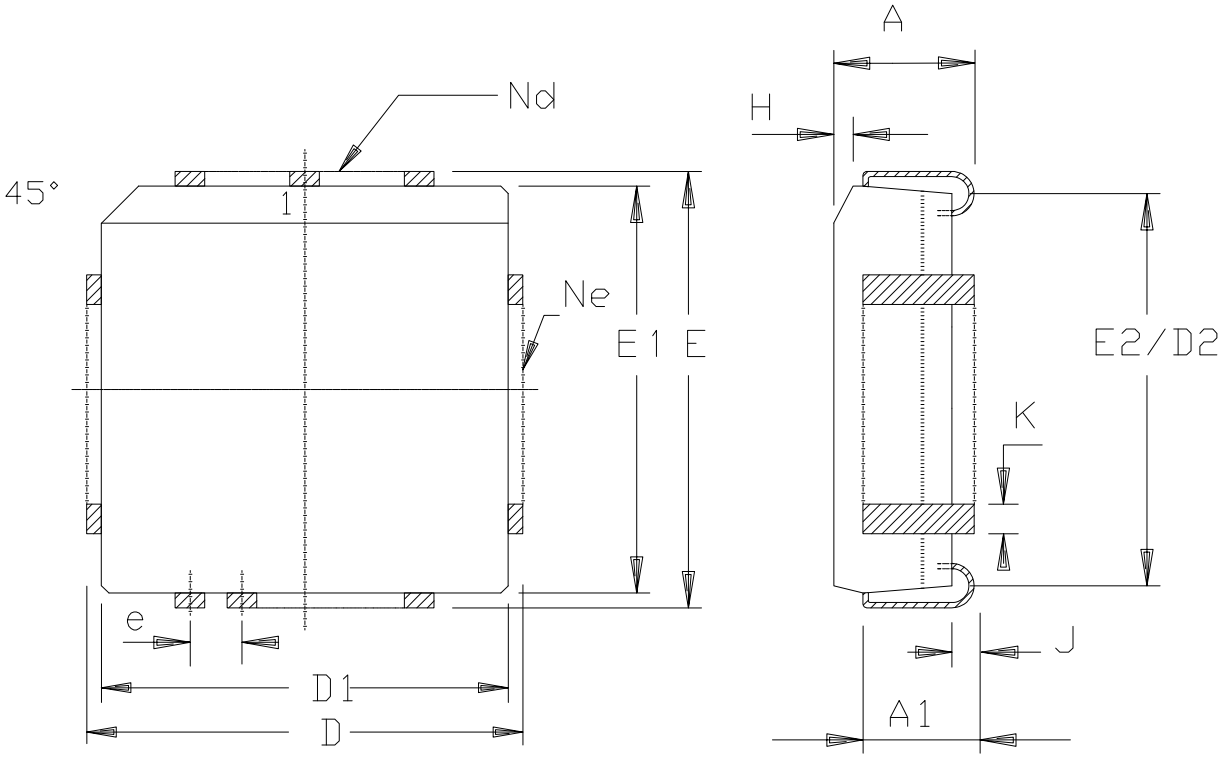
Ordering Information

Table 130. Possible Order Entries

Part Number	Supply Voltage	Temperature Range	Package
AT89C51RE2-SLSUM	2.7V-5.5V	Industrial & Green	PLCC44
AT89C51RE2-RLTUM			VQFP44
AT89C51RE2-SLSEM	2.7V-5.5V	Engineering Samples	PLCC44
AT89C51RE2-RLTEM			VQFP44

Packaging Information

PLCC44



	MM		INCH	
A	4.20	4.57	.165	.180
A1	2.29	3.04	.090	.120
D	17.40	17.65	.685	.695
D1	16.44	16.66	.647	.656
D2	14.99	16.00	.590	.630
E	17.40	17.65	.685	.695
E1	16.44	16.66	.647	.656
E2	14.99	16.00	.590	.630
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	11		11	
Ne	11		11	
PKG STD	00			

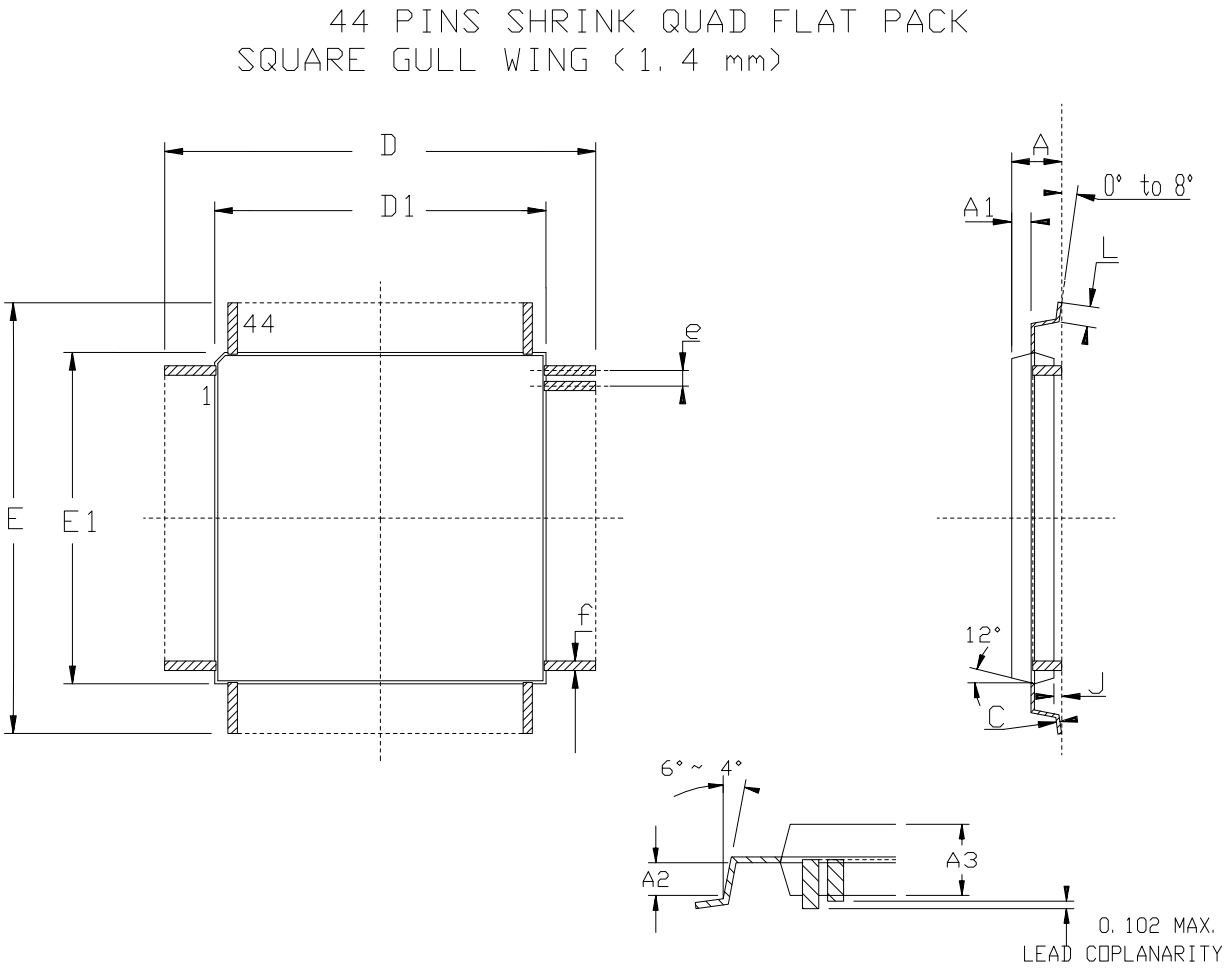
STANDARD NOTES FOR PLCC

1/ CONTROLLING DIMENSIONS : INCHES

2/ DIMENSIONING AND TOLERANCING PER ANSI Y 14.5M - 1982.

3/ "D" AND "E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTUSIONS. MOLD FLASH OR PROTUSIONS SHALL NOT EXCEED 0.20 mm (.008 INCH) PER SIDE.

VQFP44



	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	11.90	12.10	.468	.476
D1	9.90	10.10	.390	.398
E	11.90	12.10	.468	.476
E1	9.90	10.10	.390	.398
J	0.05	–	.002	–
L	0.45	0.75	.018	.030

STANDARD NOTES FOR PQFP/ VQFP / TQFP / DQFP

1/ CONTROLLING DIMENSIONS : INCHES

2/ ALL DIMENSIONING AND TOLERANCING CONFORM TO ANSI Y 14.5M - 1982.

3/ "D1 AND E1" DIMENSIONS DO NOT INCLUDE MOLD PROTUSIONS. MOLD PROTUSIONS SHALL NOT EXCEED 0.25 mm (0.010 INCH). THE TOP PACKAGE BODY SIZE MAY BE SMALLER THAN THE BOTTOM PACKAGE BODY SIZE BY AS MUCH AS 0.15 mm.

4/ DATUM PLANE "H" LOCATED AT MOLD PARTING LINE AND COINCIDENT WITH LEAD, WHERE LEAD EXITS PLASTIC BODY AT BOTTOM OF PARTING LINE.

5/ DATUM "A" AND "D" TO BE DETERMINED AT DATUM PLANE H.

6/ DIMENSION " f " DOES NOT INCLUDE DAMBAR PROTUSION ALLOWABLE DAMBAR PROTUSION SHALL BE 0.08mm/.003" TOTAL IN EXCESS OF THE " f " DIMENSION AT MAXIMUM MATERIAL CONDITION . DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

Document Revision History

Changes from 7663B to 7663C

1. Modified ordering information.
2. Various grammatical corections throughout document.

Changes from 7663C to 7663D

1. TWI interface added.

Changes from 7663D to 7663E

1. Removed 64 and 68 pins package product version.
2. Minor correction on Table 69 on page 102.

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

7663E–8051–10/08