



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

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

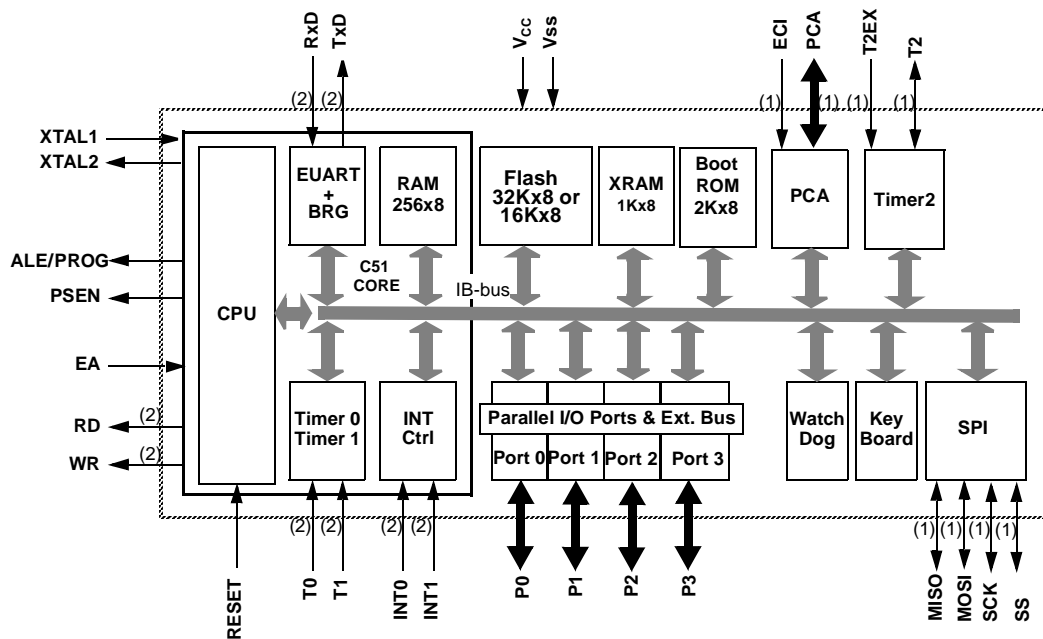
Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	80C51
Core Size	8-Bit
Speed	40MHz
Connectivity	SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-VQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51rb2-rlril

Block Diagram

Figure 1. Block Diagram



- Notes:
1. Alternate function of Port 1.
 2. Alternate function of Port 3.

Table 2. 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 3. System Management SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
PCON	87h	Power Control	SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL
AUXR	8Eh	Auxiliary Register 0	DPU	-	M0	XRS2	XRS1	XRS0	EXTRAM	AO
AUXR1	A2h	Auxiliary Register 1	-	-	ENBOOT	-	GF3	0	-	DPS
CKRL	97h	Clock Reload Register	CKRL7	CKRL6	CKRL5	CKRL4	CKRL3	CKRL2	CKRL1	CKRL0
CKCKON0	8Fh	Clock Control Register 0	-	WDTX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2
CKCKON1	AFh	Clock Control Register 1	-	-	-	-	-	-	-	SPIX2

Table 4. 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	-	-	-	-	-	ESPI	EI2C	KBD
IPH0	B7h	Interrupt Priority Control High 0	-	PPCH	PT2H	PHS	PT1H	PX1H	PT0H	PX0H
IPL0	B8h	Interrupt Priority Control Low 0	-	PPCL	PT2L	PLS	PT1L	PX1L	PT0L	PX0L
IPH1	B3h	Interrupt Priority Control High 1	-	-	-	-	-	SPIH	IE2CH	KBDH
IPL1	B2h	Interrupt Priority Control Low 1	-	-	-	-	-	SPIL	IE2CL	KBDL

Table 5. 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								

Table 16. CKCON1 Register

CKCON1 - Clock Control Register (AFh)

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

Bit Number	Bit Mnemonic	Description
7	-	Reserved
6	-	Reserved
5	-	Reserved
4	-	Reserved
3	-	Reserved
2	-	Reserved
1	-	Reserved
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 XXX0b

Not bit addressable

Table 17. AUXR1 register

AUXR1- Auxiliary Register 1(0A2h)

7	6	5	4	3	2	1	0
-	-	ENBOOT	-	GF3	0	-	DPS

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	ENBOOT	Enable Boot Flash Cleared to disable boot ROM. Set to map the boot ROM between F800h - 0FFFFh.
4	-	Reserved The value read from this bit is indeterminate. Do not set this bit.
3	GF3	This bit is a general-purpose user flag.⁽¹⁾
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 = XXXX XX0X0b

Not bit addressable

Note: 1. Bit 2 stuck at 0; this allows using 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

```

- 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 that 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 (WR) and P3.7 (RD).

The stack pointer (SP) may be located anywhere in the 256 Bytes RAM (lower and upper RAM) internal data memory. The stack may not be located in the XRAM.

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 20. 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 25. 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 26 and Table 27).

Table 26. 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 27. 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 28. 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 29. 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 (see Figure 13).

Figure 13. PCA Capture Mode

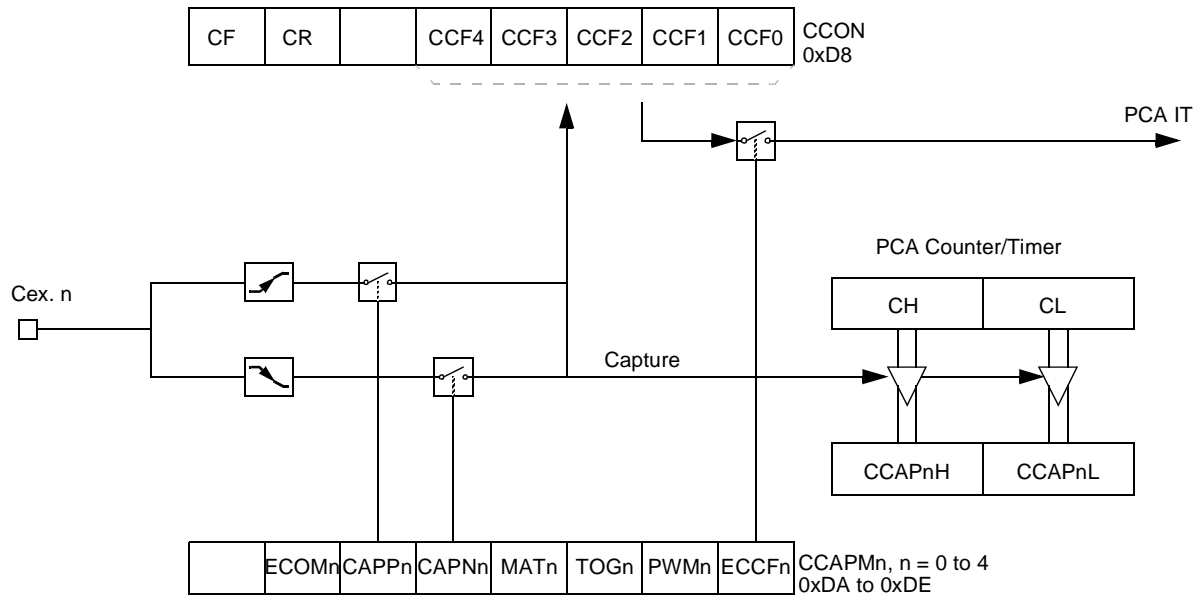


Table 33. SCON Register

SCON - Serial Control Register (98h)

7	6	5	4	3	2	1	0
FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit Number	Bit Mnemonic	Description																				
7	FE	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	Serial port Mode bit 0 Refer to SM1 for serial port mode selection. SMOD0 must be cleared to enable access to the SM0 bit.																				
6	SM1	Serial port Mode bit 1 <table><thead><tr><th>SM0</th><th>SM1</th><th>Mode</th><th>Baud Rate</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>Shift Register</td><td>$F_{XTAL}/12$ (or $F_{XTAL}/6$ in mode X2)</td></tr><tr><td>0</td><td>1</td><td>8-bit UART</td><td>Variable</td></tr><tr><td>1</td><td>0</td><td>9-bit UART</td><td>$F_{XTAL}/64$ or $F_{XTAL}/32$</td></tr><tr><td>1</td><td>1</td><td>9-bit UART</td><td>Variable</td></tr></tbody></table>	SM0	SM1	Mode	Baud Rate	0	0	Shift Register	$F_{XTAL}/12$ (or $F_{XTAL}/6$ in mode X2)	0	1	8-bit UART	Variable	1	0	9-bit UART	$F_{XTAL}/64$ or $F_{XTAL}/32$	1	1	9-bit UART	Variable
SM0	SM1	Mode	Baud Rate																			
0	0	Shift Register	$F_{XTAL}/12$ (or $F_{XTAL}/6$ in mode X2)																			
0	1	8-bit UART	Variable																			
1	0	9-bit UART	$F_{XTAL}/64$ or $F_{XTAL}/32$																			
1	1	9-bit UART	Variable																			
5	SM2	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	Reception Enable bit Clear to disable serial reception. Set to enable serial reception.																				
3	TB8	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	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, RB8 is the received stop bit. In mode 0 RB8 is not used.																				
1	TI	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	Receive Interrupt flag Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 18. and Figure 19. in the other modes.																				

Reset Value = 0000 0000b

Bit addressable

Table 34. 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 35. 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 20.), but also for mode 0 for UART, thanks to the bit SRC located in BDRCON register (Table 42.)

UART Registers

Table 36. SADEN Register

SADEN - Slave Address Mask Register for UART (B9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 37. SADDR Register

SADDR - Slave Address Register for UART (A9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 44. IENO Register

IENO - 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 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 45. 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 see PPCH for priority level.
5	PT2L	Timer 2 Overflow Interrupt Priority Bit see PT2H for priority level.
4	PSL	Serial Port Priority Bit see PSH for priority level.
3	PT1L	Timer 1 Overflow Interrupt Priority Bit see PT1H for priority level.
2	PX1L	External Interrupt 1 Priority Bit see PX1H for priority level.
1	PT0L	Timer 0 Overflow Interrupt Priority Bit see PT0H for priority level.
0	PX0L	External Interrupt 0 Priority Bit see PX0H for priority level.

Reset Value = X000 0000b

Bit addressable

Registers

Table 51. 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 KBKIE. 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

This register is read only access, all flags are automatically cleared by reading the register.

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 $96 \times T_{CLK\ PERIPH}$, 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 16 ms to 2 s @ $F_{OSCA} = 12\text{ MHz}$. To manage this feature, see WDTPRG register description, Table 59.

Table 59. 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.

Reset Recommendation to Prevent Flash Corruption

An example of bad initialization situation may occur in an instance where the bit ENBOOT in AUXR1 register is initialized from the hardware bit BLJB upon reset. Since this bit allows mapping of the bootloader in the code area, a reset failure can be critical.

If one wants the ENBOOT cleared in order to unmap the boot from the code area (yet due to a bad reset) the bit ENBOOT in SFRs may be set. If the value of Program Counter is accidentally in the range of the boot memory addresses then a Flash access (write or erase) may corrupt the Flash on-chip memory.

It is recommended to use an external reset circuitry featuring power supply monitoring to prevent system malfunction during periods of insufficient power supply voltage (power supply failure, power supply switched off).

Idle Mode

An instruction that sets PCON.0 indicates that it is the last instruction to be executed before going into Idle mode. In Idle mode, the internal clock signal is gated off to the CPU, but not to the interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator and all other registers maintain their data during idle. The port pins hold the logical states they had at the time Idle was activated. ALE and PSEN hold at logic high level.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into idle.

The flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during idle. For example, an instruction that activates idle can also set one or both flag bits. When idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

Power-down Mode

To save maximum power, a Power-down mode can be invoked by software (see Table 14, PCON register).

In Power-down mode, the oscillator is stopped and the instruction that invoked Power-down mode is the last instruction executed. The internal RAM and SFRs retain their value until the Power-down mode is terminated. V_{CC} can be lowered to save further power. Either a hardware reset or an external interrupt can cause an exit from Power-down. To properly terminate Power-down, the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize.

Only external interrupts $\overline{INT0}$, $\overline{INT1}$ and Keyboard Interrupts are useful to exit from Power-down. For that, interrupt must be enabled and configured as level or edge sensitive interrupt input. When Keyboard Interrupt occurs after a power down mode, 1024 clocks are necessary to exit to power down mode and enter in operating mode.

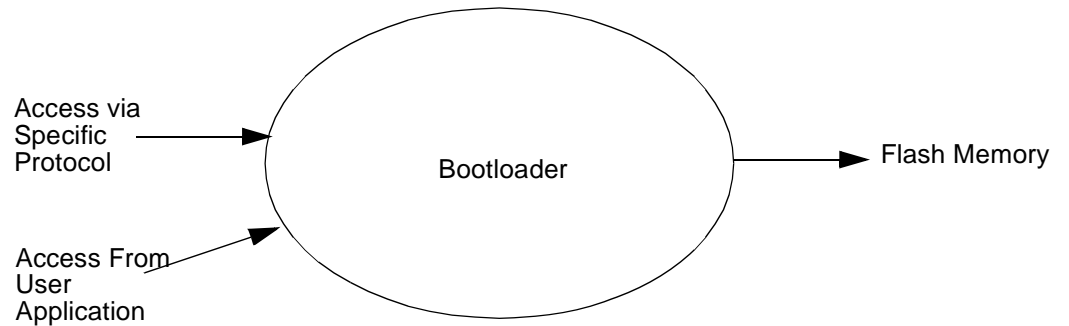
Holding the pin low restarts the oscillator but bringing the pin high completes the exit as detailed in Figure 34. When both interrupts are enabled, the oscillator restarts as soon as one of the two inputs is held low and power down exit will be completed when the first input will be released. In this case, the higher priority interrupt service routine is executed. Once the interrupt is serviced, the next instruction to be executed after RETI will

Bootloader Architecture

Introduction

The bootloader manages a communication according to a specific defined protocol to provide the whole access and service on Flash memory. Furthermore, all accesses and routines can be called from the user application.

Figure 36. Diagram Context Description



Acronyms

ISP: In-system Programming

SBV: Software Boot Vector

BSB: Boot Status Byte

SSB: Software Security Bit

HW : Hardware Byte

Example

Display data from address 0000h to 0020h

```

HOST          : 05 0000 04 0000 0020 00 D7
BOOTLOADER    : 05 0000 04 0000 0020 00 D7
BOOTLOADER    0000=-----data----- CR LF    (16 data)
BOOTLOADER    0010=-----data----- CR LF    (16 data)
BOOTLOADER    0020=data CR LF                  ( 1 data)
  
```

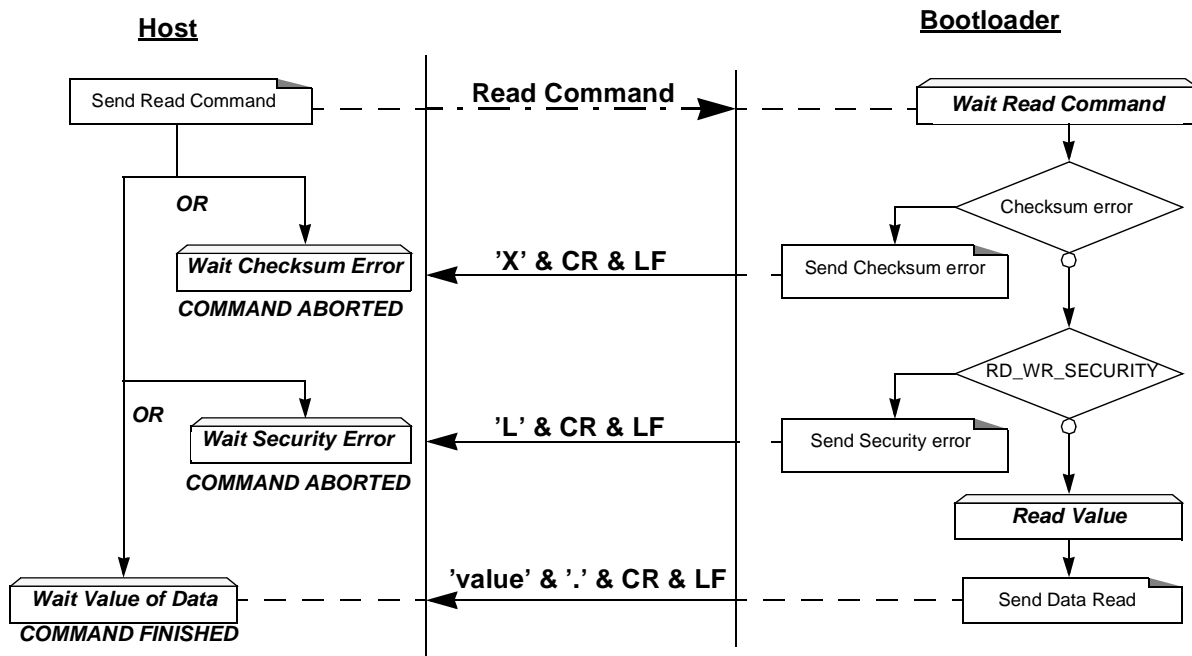
Read Function

This flow is similar for the following frames:

- Reading Frame
- EOF Frame/Atmel Frame (only reading Atmel Frame)

Description

Figure 45. Read Flow



Example

Read function (read SBV)

```

HOST          : 02 0000 05 07 02 F0
BOOTLOADER    : 02 0000 05 07 02 F0 Value . CR LF
  
```

Atmel Read function (read Bootloader version)

```

HOST          : 02 0000 01 02 00 FB
BOOTLOADER    : 02 0000 01 02 00 FB Value . CR LF
  
```

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 PSEN Low.

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

Table 75, Table 78, and Table 80 give the description of each AC symbols.

Table 77, Table 79 and Table 81 give the AC parameter for each range.

Table 76, Table 77 and Table 82 gives 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.

$x = 35 \text{ ns}$

$T = 50 \text{ ns}$

$T_{CCIV} = 4T - x = 165 \text{ ns}$

External Program Memory Characteristics

Table 75. 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 $\overline{\text{PSEN}}$
T_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width
T_{PLIV}	$\overline{\text{PSEN}}$ to Valid Instruction In
T_{PXIX}	Input Instruction Hold after $\overline{\text{PSEN}}$
T_{PXIZ}	Input Instruction Float after $\overline{\text{PSEN}}$
T_{AVIV}	Address to Valid Instruction In
T_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float

Symbol	Type	Standard Clock	X2 Clock	X Parameter for - M Range	X Parameter for - L 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_{RHDZ}	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

External Data Memory Write Cycle

