Features

- 80C51 Core :
  - 6 Clocks per Instruction
  - Speed up to 16 MHz
- 768 Bytes RAM
- AT83C5122: 32 Kbytes ROM
- AT83C5123: 30 Kbytes ROM
- AT85C5122: 32 Kbytes Code RAM and 32 Kbytes ROM
- AT89C5122 : 32 Kbytes Flash
- AT85EC5122, AT83EC5123, AT83EC5122: Additional 512 Bytes EEPROM (AT24C04)
- Multi-protocol Smart Card Interface
  - Certified According to ISO7816, EMV2000, GIE-CB and WHQM Standards
  - Asynchronous Protocols T = 0 and T = 1, with Direct and Inverse Modes
  - Step-up/Down Converter with Programmable Voltage Output: 5V and 3V (60 mA), 1.8V (20 mA)
  - 4 kV ESD Protection (MIL/STD 833 Class 3)
- Alternate Card Support with CLK, IO and RST
- USB Module with 7 Endpoints Programmable with In or Out Directions and with ISO, Bulk or Interrupt Transfers
- UART with Integrated Baud Rate Generator (BRG)
- 8 MHz On-chip Oscillator Analog PLL for 96 MHz Synthesis, Possible 48 MHz Clock Input
- Two 16-bit Timer/Counters: T0 and T1
- Hardware Watchdog and Power-fail Detector (PFD)
- Idle and Power-down Modes
- Self Powered USB
- Low Power
  - 30 mA Maximum Operating Current (at 32 MHz X1)
  - 100 µA Maximum Power-down Current at 5.4V (without Smart Card and USB)
- Voltage Range: 3.6 to 5.5V

- For AT8xC5122 version:
  - Keyboard Interrupt Interface on Port 5 (8 Bits)
  - Five 8-bit I/O Ports, One 6-bit
  - SPI Interface (Master Slave)
  - Packages: VQFP64, PLCC28
  - Seven LED Outputs with Programmable Current Sources: 2-4-10 mA

- For AT8xC5123 version:
  - Four LED Outputs with Programmable Current Sources: 2-4-10 mA
  - Two 8-bit I/O Ports, One 6-bit I/O port, One I/O bit (on LQP32 package)
  - Packages: LQFP32, PLCC28

C51 Microcontroller with USB and Smart Card Reader Interfaces

AT83C5122
AT83EC5122
AT85C5122
AT85EC5122
AT89C5122
AT83C5123
AT83EC5123

Preliminary

Rev. 4202B–SCR–07/03

Description

AT8xC5122 is a high-performance CMOS derivative of the 80C51 8-bit microcontrollers optimized for USB keyboard with smart card reader applications.

AT8xC5122 retains the features of the Atmel 80C51 with 32 Kbytes ROM capacity, 768 bytes of internal RAM, a 4-level interrupt system, two 16-bit timer/counters (T0/T1), a full duplex enhanced UART (EUART) with baud rate generator (BRG) and an on-chip oscillator.

In addition, AT8xC5122 has a USB 2.0 full-speed function controller with seven Endpoints, a multi-protocol smart card interface, a dual data pointer, seven programmable LED current sources (2-4-10 mA) and a hardware watchdog.

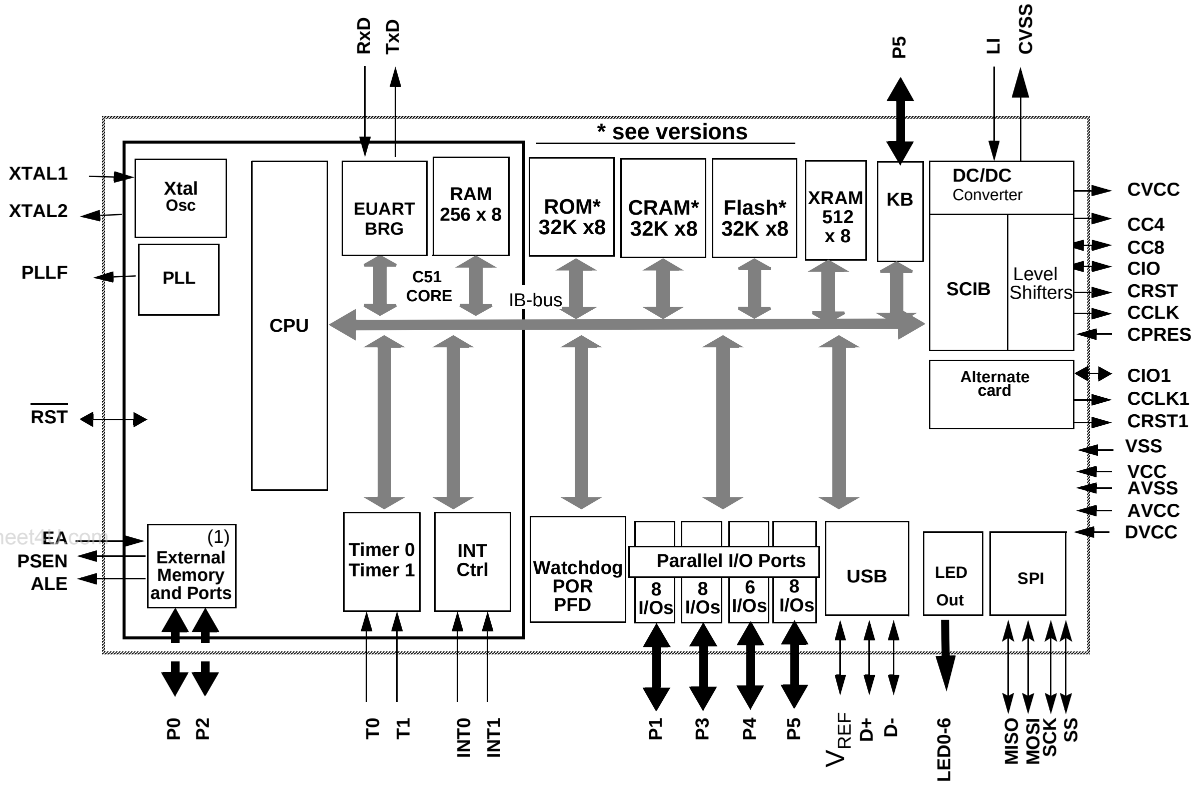
AT8xC5122 Flash RAM version and AT8xC5122 Code RAM version with 32 Kbytes memory can be loaded by In-System Programming (ISP) software residing in the on-chip ROM from USB or UART.

AT8xC5122 have 2 software-selectable modes of reduced activity for further reduction in power consumption.

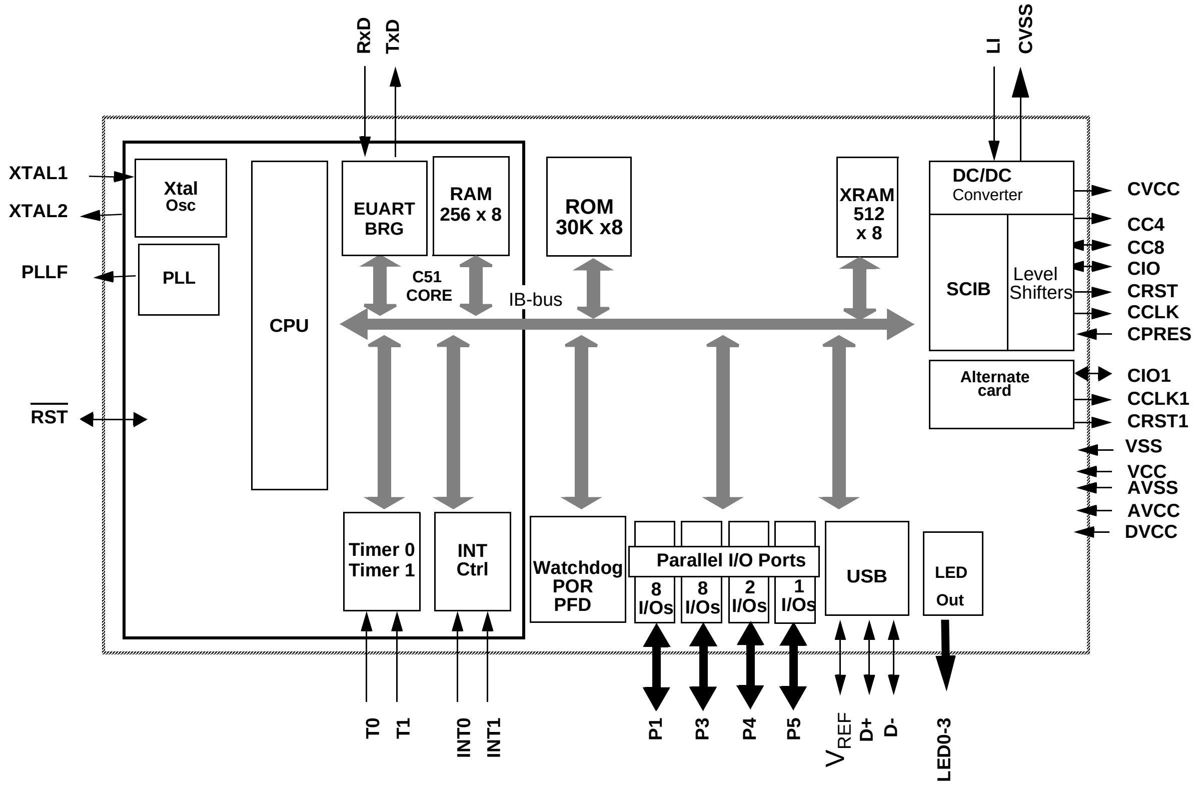
AT8xC5123 is a low pin count of the AT8xC5122. This version doesn't have the keyboard and the SPI interfaces. The PLCC28 packages for AT8xC5122 and AT8xC5123 have the same pinout. The AT8xC5123 is also proposed in a VQFP32 package.

AT8xC5122 and AT8xC5123 are proposed with a 512 bytes EEPROM (AT24C04) and respectively named AT8xEC5122 and AT8xEC5123.

AT8xC5122 Block Diagram



AT8xC5123 Block Diagram



Pin Description

Figure 1. VQFP64 Pinout (only for AT8xC5122)

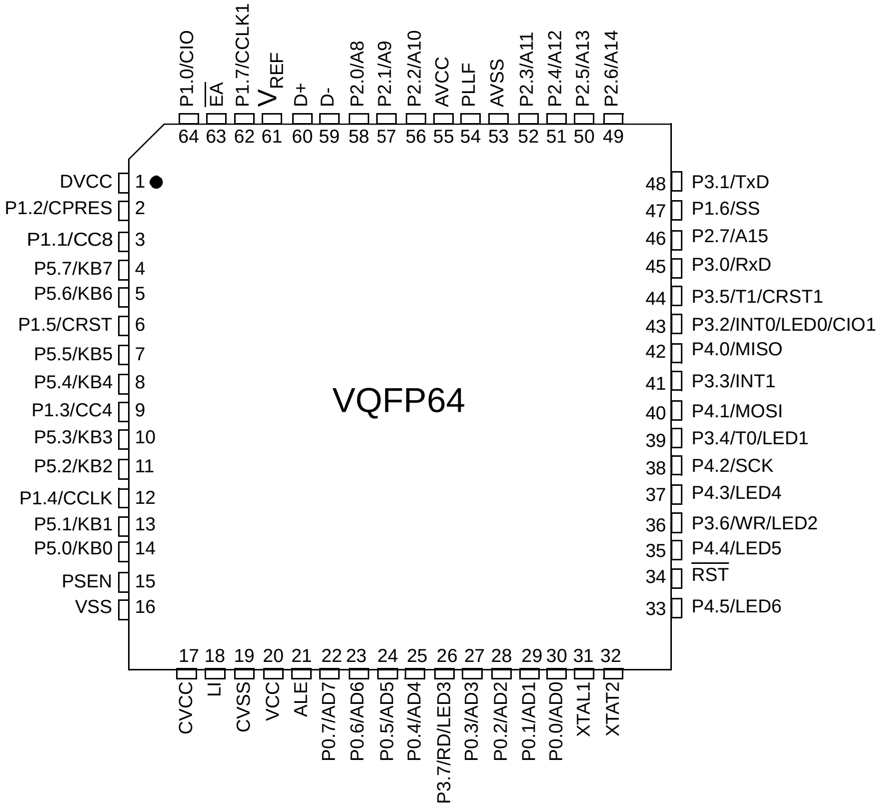


Figure 2. PLCC28 Pinout

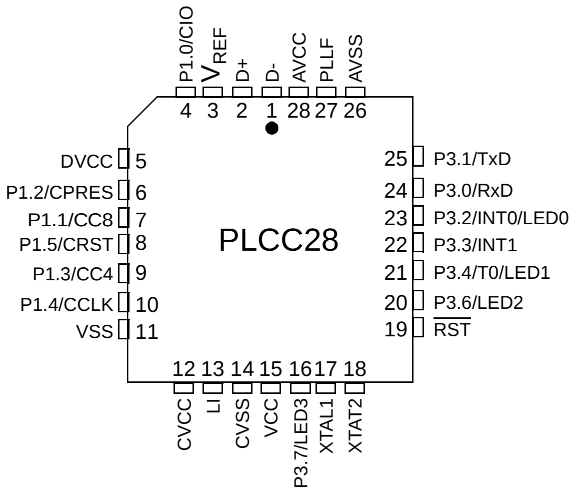


Figure 3. LQFP32 Pinout (only for AT83C5123)

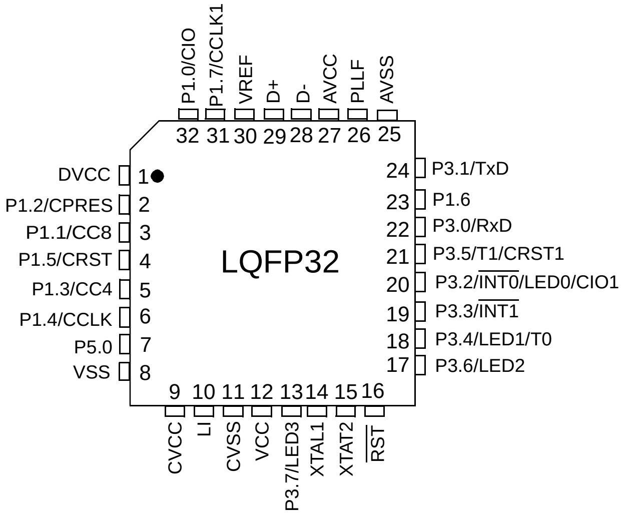
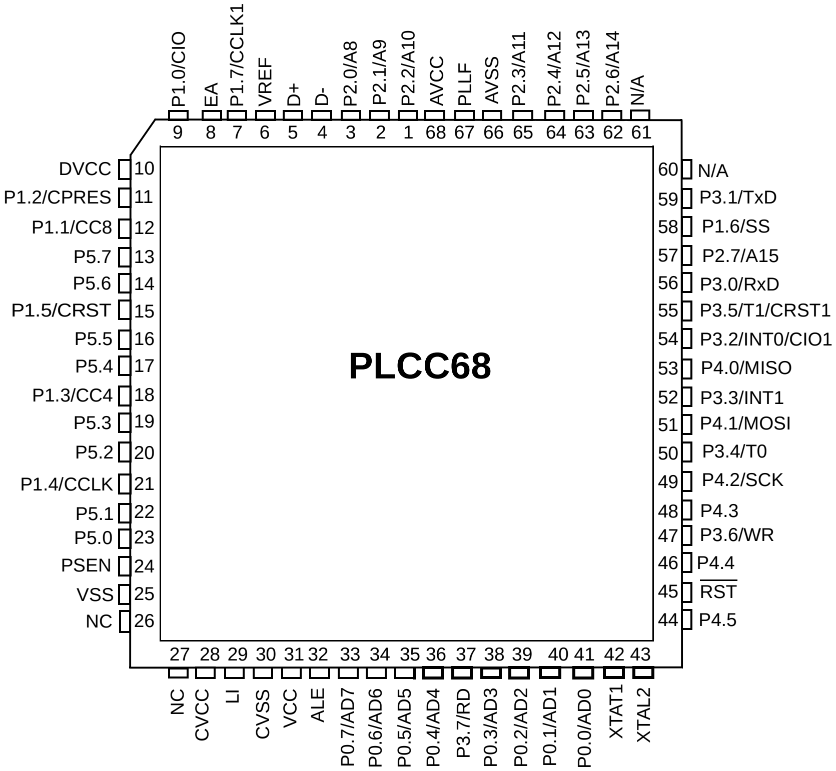


Figure 4. PLCC68 Pinout (Engineering package only for AT85C5122, check availability with ATMEL sales office)



Signals

All the signals are detailed in Table 1:

Table 1. Pinout Description

Port	VQFP64	LQFP32	PLCC68	PLCC28	Internal Power Supply	ESD	I/O	Reset Level	Alt	Reset Config	Conf 1	Conf 2	Conf 3	Led
P0.0	30	-	41	-	VCC	2KV	I/O	Float	AD0	P0		KB_OUT	Push-pull	
P0.1	29	-	40	-	VCC	2KV	I/O	Float	AD1	P0		KB_OUT	Push-pull	
P0.2	28	-	39	-	VCC	2KV	I/O	Float	AD2	P0		KB_OUT	Push-pull	
P0.3	27	-	38	-	VCC	2KV	I/O	Float	AD3	P0		KB_OUT	Push-pull	
P0.4	25	-	36	-	VCC	2KV	I/O	Float	AD4	P0		KB_OUT	Push-pull	
P0.5	24	-	35	-	VCC	2KV	I/O	Float	AD5	P0		KB_OUT	Push-pull	
P0.6	23	-	34	-	VCC	2KV	I/O	Float	AD6	P0		KB_OUT	Push-pull	
P0.7	22	-	33	-	VCC	2KV	I/O	Float	AD7	P0		KB_OUT	Push-pull	
P1.0	64	32	9	4	CVCC	4KV	I/O	0	CIO	Port51	CVCC inactive at reset			
P1.1	3	3	12	7	CVCC	4KV	I/O	0	CC8	Port51	CVCC inactive at reset			
P1.2	2	2	11	6	VCC	2KV	I/O	1	CPRES	Port51	Weak & medium pull-up can be disconnected			
P1.3	9	5	18	9	CVCC	4KV	I/O	0	CC4	Port51	CVCC inactive at reset			
P1.4	12	6	21	10	CVCC	4KV	O	0	CCLK	Push-pull	CVCC inactive at reset			
P1.5	6	4	15	8	CVCC	4KV	O	0	CRST	Push-pull	CVCC inactive at reset			
P1.6	47	23	58	-	VCC	2KV	I/O	1	SS	Port51				
P1.7	62	31	7	-	VCC	2KV	I/O	1	CCLK1	Port51				
P2.0	58	-	3	-	VCC	2KV	I/O	1	A8	Port51	Push-pull	KB_OUT	Input WPU	
P2.1	57	-	2	-	VCC	2KV	I/O	1	A9	Port51	Push-pull	KB_OUT	Input WPU	
P2.2	56	-	1	-	VCC	2KV	I/O	1	A10	Port51	Push-pull	KB_OUT	Input WPU	
P2.3	52	-	65	-	VCC	2KV	I/O	1	A11	Port51	Push-pull	KB_OUT	Input WPU	
P2.4	51	-	64	-	VCC	2KV	I/O	1	A12	Port51	Push-pull	KB_OUT	Input WPU	
P2.5	50	-	63	-	VCC	2KV	I/O	1	A13	Port51	Push-pull	KB_OUT	Input WPU	
P2.6	49	-	62	-	VCC	2KV	I/O	1	A14	Port51	Push-pull	KB_OUT	Input WPU	
P2.7	46	-	57	-	VCC	2KV	I/O	1	A15	Port51	Push-pull	KB_OUT	Input WPU	

Table 1. Pinout Description (Continued)

Port	VQFP64	LQFP32	PLCC68	PLCC28	Internal Power Supply	ESD	I/O	Reset Level	Alt	Reset Config	Conf 1	Conf 2	Conf 3	Led
P3.0	45	22	56	24	VCC	2KV	I/O	1	RxD	Port51	Push-pull	KB_OUT	Input WPU	
P3.1	48	24	59	25	VCC	2KV	I/O	1	TxD	Port51	Push-pull	KB_OUT	Input WPU	
P3.2	43	20	54	23	VCC	2KV	I/O	1	INT0	Port51				LED0
P3.3	41	19	52	22	VCC	2KV	I/O	1	INT1	Port51	Push-pull	KB_OUT	Input WPU	
P3.4	39	18	50	21	VCC	2KV	I/O	1	T0	Port51	Push-pull	KB_OUT	Input WPU	LED1
P3.5	44	21	55	-	VCC	2KV	I/O	1	T1	Port51				
P3.6	36	17	47	20	VCC	2KV	I/O	1	WR	Port51				LED2
P3.7	26	13	37	16	VCC	2KV	I/O	1	RD	Port51				LED3
P4.0	42	-	53	-	VCC	2KV	I/O	1	MISO	Port51				
P4.1	40	-	51	-	VCC	2KV	I/O	1	MOSI	Port51				
P4.2	38	-	49	-	VCC	2KV	I/O	1	SCK	Port51				
P4.3	37	-	48	-	VCC	2KV	I/O	1		Port51	Push-pull	KB_OUT	Input MPU	LED4
P4.4	35	-	46	-	VCC	2KV	I/O	1		Port51	Push-pull	KB_OUT	Input MPU	LED5
P4.5	33	-	44	-	VCC	2KV	I/O	1		Port51	Push-pull	KB_OUT	Input MPU	LED6
P4.6	-	-	61	-	VCC	2KV	I/O	1	**Reserved**					
P4.7	-	-	60	-	VCC	2KV	I/O	1	**Reserved**					
P5.0	14	7	23	-	VCC	2KV	I/O	1	KB0	Port51	Push-pull	Input MPU	Input WPU	
P5.1	13	-	22	-	VCC	2KV	I/O	1	KB1	Port51	Push-pull	Input MPU	Input WPU	
P5.2	11	-	20	-	VCC	2KV	I/O	1	KB2	Port51	Push-pull	Input MPU	Input WPU	
P5.3	10	-	19	-	VCC	2KV	I/O	1	KB3	Port51	Push-pull	Input WPD	Input WPU	
P5.4	8	-	17	-	VCC	2KV	I/O	1	KB4	Port51	Push-pull	Input WPD	Input WPU	
P5.5	7	-	16	-	VCC	2KV	I/O	1	KB5	Port51	Push-pull	Input WPD	Input WPU	

Table 1. Pinout Description (Continued)

Port	VQFP64	LQFP32	PLCC68	PLCC28	Internal Power Supply	ESD	I/O	Reset Level	Alt	Reset Config	Conf 1	Conf 2	Conf 3	Led
P5.6	5	-	14	-	VCC	2KV	I/O	1	KB6	Port51	Push-pull	Input WPD	Input WPU	
P5.7	4	-	13	-	VCC	2KV	I/O	1	KB7	Port51	Push-pull	Input WPD	Input WPU	
$\overline{\text{RST}}$	34	16	45	19	VCC		I/0		**Reset Input** Holding this pin low for 64 oscillator periods while the oscillator is running resets the device. The Port pins are driven to their reset conditions when a voltage lower than V_{IL} is applied, whether or not the oscillator is running. This pin has an internal pull-up resistor which allows the device to be reset by connecting a capacitor between this pin and VSS. Asserting $\overline{\text{RST}}$ when the chip is in Idle mode or Power-Down mode returns the chip to normal operation. The output is active for at least 12 oscillator periods when an internal reset occurs.					
D+	60	29	5	2	DVCC		I/O		**USB Positive Data Upstream Port** This pin requires an external 1.5 kΩ pull-up to V_{REF} for full speed					
D-	59	28	4	1	DVCC		I/O		**USB Negative Data Upstream Port**					
V_{REF}	61	30	6	3	AVCC		O		**USB Voltage Reference**: $3.0 < V_{REF} < 3.6$ V V_{REF} can be connected to D+ with a 1.5 kΩ resistor. The V_{REF} voltage is controlled by software.					
XTAL 1	31	14	42	17	VCC		I		**Input to the on-chip inverting oscillator amplifier** To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, its output is connected to this pin.					
XTAL 2	32	15	43	18	VCC		O		**Output of the on-chip inverting oscillator amplifier** To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, leave XTAL2 unconnected.					
$\overline{\text{EA}}$	63	-	8	-	VCC		I		**External Access Enable** $\overline{\text{EA}}$ must be strapped to ground in order to enable the device to fetch code from external memory locations 0000h to FFFFh. If security level 1 is programmed, $\overline{\text{EA}}$ will be latched on reset.					
ALE	21	-	32	-	VCC		O		**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					
$\overline{\text{PSEN}}$	15	-	24	-	VCC		O		**Program Strobe 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.					
PLLF	54	26	67	27	AVCC		O		**PLL Low Pass Filter input** Receives the RC network of the PLL low pass filter.					
AVCC	55	27	68	28			PWR		**Analog Supply Voltage** AVCC is used to supply the on-chip PLL and the USB drivers					

Table 1. Pinout Description (Continued)

Port	VQFP64	LQFP32	PLCC68	PLCC28	Internal Power Supply	ESD	I/O	Reset Level	Alt	Reset Config	Conf 1	Conf 2	Conf 3	Led
VCC	20	12	31	15			PWR		**Supply Voltage** VCC is used to power the internal voltage regulators and internal I/O's					
LI	18	10	29	13			PWR		**DC/DC Input** LI must be tied to VCC through an external coil and provide the current for the pump charge of the DC/DC converter					
CVCC	17	9	28	12			PWR		**Card Supply Voltage** CVCC is the programmable voltage output for the Card interface. It must be connected to an external decoupling capacitor					
DVCC	1	1	10	5			PWR		**Digital Supply Voltage** DVCC is used to supply the digital core and internal I/O's. It is internally connected to the output of a 3.3V voltage regulator and must be connected to an external decoupling capacitor					
CVSS	19	11	30	14			GND		**DC/DC Ground** CVSS is used to sink high shunt currents from the external coil					
VSS	16	8	25	11			GND		**Digital Ground** VSS is used to supply the buffer ring and the digital core					
AVSS	53	25	66	26			GND		**Analog Ground** AVSS is used to supply the on-chip PLL and the USB drivers					

I/O Port Definition

Ports vs packages

Table 2. IO number vs packages

	P0	P1	P2	P3	P4	P5	Total
VQFP64	8	8	8	8	8	6	46
LQFP32		8		8		1	17
PLCC28		6		6		1	13
PLCC68	8	8	8	8	8	8	48

Port 0

Port 0 has the following functions:

– Default function: Port 0 is an 8-bit I/O port.

– Alternate function: Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application, it uses strong internal pull-ups when emitting 1's and it can drive CMOS inputs without external pull-ups.

Port 0 has the following configurations:

– Default configuration: open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in this state they can be used as high-impedance inputs.

– Configuration 2: Low speed output, "KB_OUT"

– Configuration 3: Push-pull output

Port 1

Port 1 has the following functions:

– Default function: Only Port 1.2, P1.6 and P1.7 are standard I/O's; the other ports can only be activated with the SCIB function.

– Alternate function and configuration: see Table 3.

Table 3. Port 1 description.

	Alternate Function		**Configuration**	
Port	**Signal**	**Description**	**Mode**	**Description**
P1.0	CIO	**Smart card interface function** Card I/O	Port51	CVCC supply: inactive at reset
P1.1	CC8	**Smart card interface function** Card contact 8	Push-pull	CVCC supply: zero level at reset
P1.2	CPRES	**Smart card interface function** Card presence	Port51	Weak & medium pull-up can be deconnected by software
P1.3	CC4	**Smart card interface function** Card contact 4	Push-pull	CVCC supply: zero level at reset
P1.4	CCLK	**Smart card interface function** Card clock	Push-pull	CVCC supply: zero level at reset
P1.5	CRST	**Smart card interface function** Card reset	Push-pull	CVCC supply: zero level at reset
P1.6	$\overline{SS}$	$\overline{SS}$ pin of the SPI function	Port51	

P1.7	CCLK1	Alternate smart card clock output	Port51/Push-pull	

Note: P1.7 is switched automatically to Push-pull when the alternate clock is selected (see Table 42 on page 48)

Port 2

Port 2 has the following functions:

– Default function: Port 2 is an 8-bit I/O port.

– Alternate function 1: Port 2 is also the multiplexed high-order address during accesses to external Program and Data Memory. In this application, it uses strong internal pull-ups when emitting 1's and it can drive CMOS inputs without external pull-ups.

Port 2 has the following configurations:

– Default configuration: Pseudo bi-directional "Port51" digital input/output with internal pull-ups.

– Configuration 1: Push-pull output

– Configuration 2: Low speed output, "KB_OUT

– Configuration 3: Input with weak pull-up, "WPU input"

Port 3

Port 3 has the following functions:

– Default function: Port 3 is an 8-bit I/O port.

– Alternate functions: see table below

Port 3 has the following configurations:

– Default configuration: Pseudo bi-directional "Port51" digital input/output with internal pull-ups.

– Alternate configurations: See Table 4.

Table 4. Port 3 description

	Alternate Functions		Configurations			
Port	**Signal**	**Description**	**Mode 1**	**Mode 2**	**Mode 3**	**Mode 4**
P3.0	RxD	Receiver data input (asynchronous) or data input/output (synchronous) of the serial interface	Push-pull	KB_OUT	Input WPU	
P3.1	TxD	Transmitter data output (asynchronous) or clock output (synchronous) of the serial interface	Push-pull	KB_OUT	Input WPU	
P3.2	INT0	External interrupt 0 input/timer 0 gate control input				LED0
P3.3	INT1	External interrupt 1input/timer 1 gate control input	Push-pull	KB_OUT	Input WPU	
P3.4	T0	Timer 0 counter input	Push-pull	KB_OUT	Input WPU	LED1
P3.5	T1	Timer 1 counter input				
P3.6	WR	External Data Memory write strobe; latches the data byte from port 0 into the external data memory				LED2
P3.7	RD	External Data Memory read strobe; Enables the external data memory. Port 3 can drive CMOS inputs without external pull-ups				LED3

Port 4

Port 4 has the following functions:

- – Default function: Port 4 is an 6-bit I/O port.
- – Alternate functions: see table below

Port 4 has the following configurations:

- – Default configuration: Pseudo bi-directional “Port51” digital input/output with internal pull-ups.
- – Alternate configurations: See Table 5..

Table 5. Port 4 description

	Alternate Functions		Configurations		
Port	Signal	Description	Mode 1	Mode 2	Mode 3
P4.0	MISO	SPI Master In Slave Out I/O			
P4.1	MOSI	SPI Master Out Slave In I/O			
P4.2	SCK	SPI clock			
P4.3			Push-pull	KB_OUT	Input MPU
P4.4			Push-pull	KB_OUT	Input MPU
P4.5			Push-pull	KB_OUT	Input MPU

Port 5

Port 5 has the following functions:

- – Default function: Port 5 is an 8-bit I/O port.
- – Alternate function 1: Port 5 is an 8-bit keyboard port KB0 to KB7.

Port 5 has the following configurations:

- – Default configuration: Pseudo bi-directional “Port51” digital input/output with internal pull-ups.
- – Alternate configuration: see Table 6..

Table 6. Port 5 description

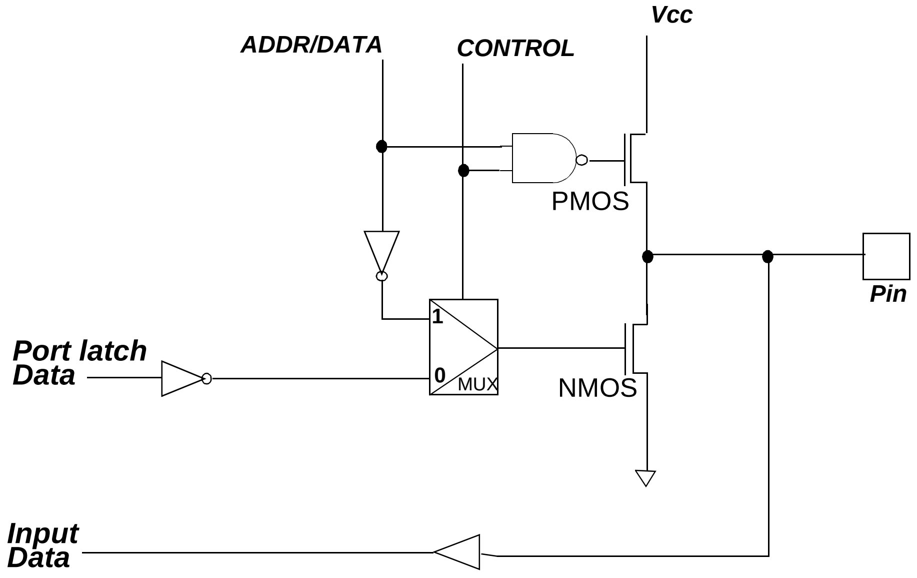
	Configurations			
Port	Mode 1	Mode 2	Mode 3	
P5.0	Push-pull	Input MPU	Input WPU	First cluster
P5.1	Push-pull	Input MPU	Input WPU	
P5.2	Push-pull	Input MPU	Input WPU	
P5.3	Push-pull	Input WPD	Input WPU	Second cluster
P5.4	Push-pull	Input WPD	Input WPU	
P5.5	Push-pull	Input WPD	Input WPU	
P5.6	Push-pull	Input WPD	Input WPU	Third cluster
P5.7	Push-pull	Input WPD	Input WPU	

Port Configuration

Standard I/O P0

The P0 port is described in Figure 5.

Figure 5. Standard Input/Output P0



Quasi-Bi-directional Output Configuration

The default port output configuration for standard I/O ports is the quasi-bi-directional output that is common on the 80C51 and most of its derivatives. The “Port51” output type can be used as both an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low.

When the port outputs a logic low state, it is driven strongly and is able to sink a fairly large current.

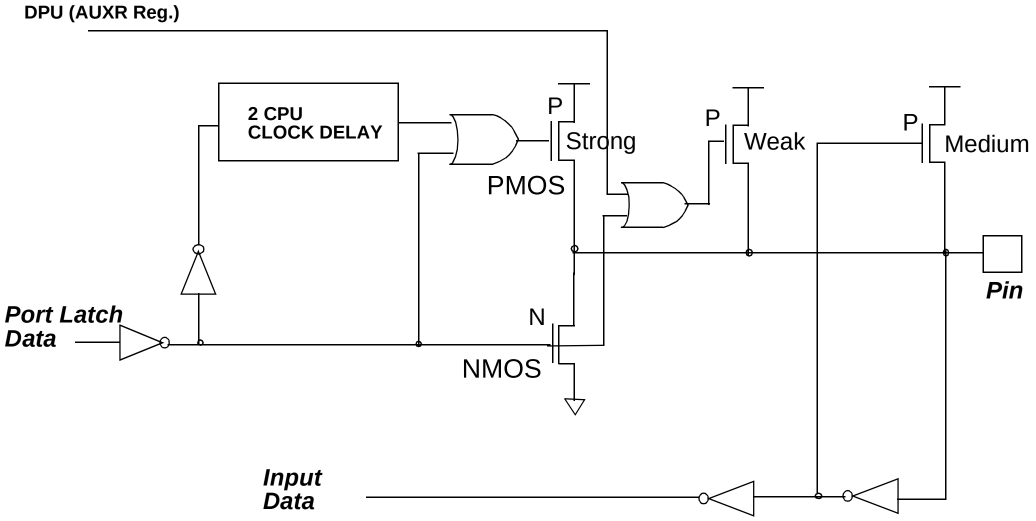
These features are somewhat similar to an open-drain output except that there are three pull-up transistors in the quasi-bi-directional output that serve different purposes.

One of these pull-ups, called the weak pull-up, is turned on whenever the port latch for the pin contains a logic 1. The weak pull-up sources a very small current that will pull the pin high if it is left floating. The weak pull-up can be turned off by the DPU bit in AUXR register.

A second pull-up, called the medium pull-up, is turned on when the port latch for the pin contains a logic 1 and the pin itself is also at a logic 1 level. This pull-up provides the primary source current for a quasi-bi-directional pin that is outputting a 1. If a pin that has a logic 1 on it is pulled low by an external device, the medium pull-up turns off, and only the weak pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to overpower the medium pull-up and take the voltage on the port pin below its input threshold.

The “Port51” is described in Figure 6.

Figure 6. Quasi-Bi-directional Output

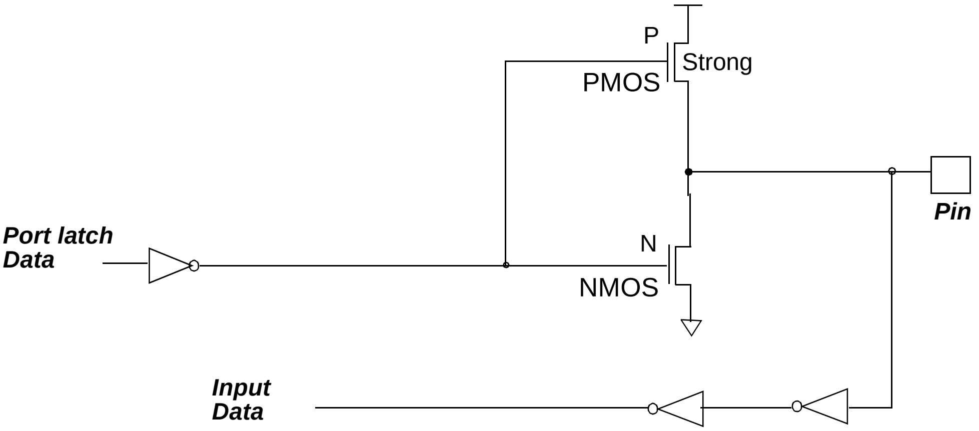


Push-pull Output Configuration

The push-pull output configuration has the same pull-down structure as both the open drain and the quasi-bi-directional output modes, but provides a continuous strong pull-up when the port latch contains a logic 1. The push-pull mode may be used when more source current is needed from a port output.

The Push-pull port configuration is shown in Figure 7.

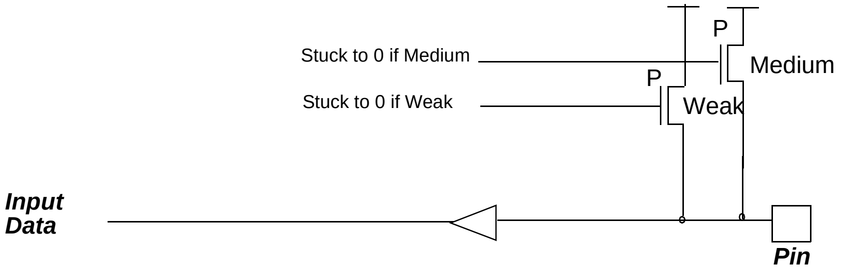
Figure 7. Push-pull Output



Input with Medium or Weak Pull-up Configuration

The input with pull-up (Input MPU and Input WPU) configuration is shown in Figure 8.

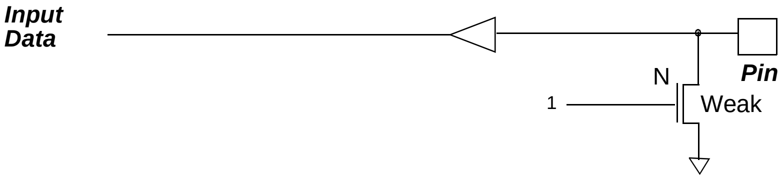
Figure 8. Input with Pull-up



Input with Weak Pull-down Configuration

The input with pull-down (input WPD) configuration is shown in Figure 9

Figure 9. Input with Pull-down



Low Speed Output Configuration

The low speed output with low speed t_{FALL} and t_{RISE} can drive keyboard.

The current limitation of the LED2CTRL block requires a polarisation current of about 250 µA. This block is automatically disabled in power-down mode.

The low speed output configuration (KB_OUT) is shown in Figure 10.

Figure 10. Low-speed Output

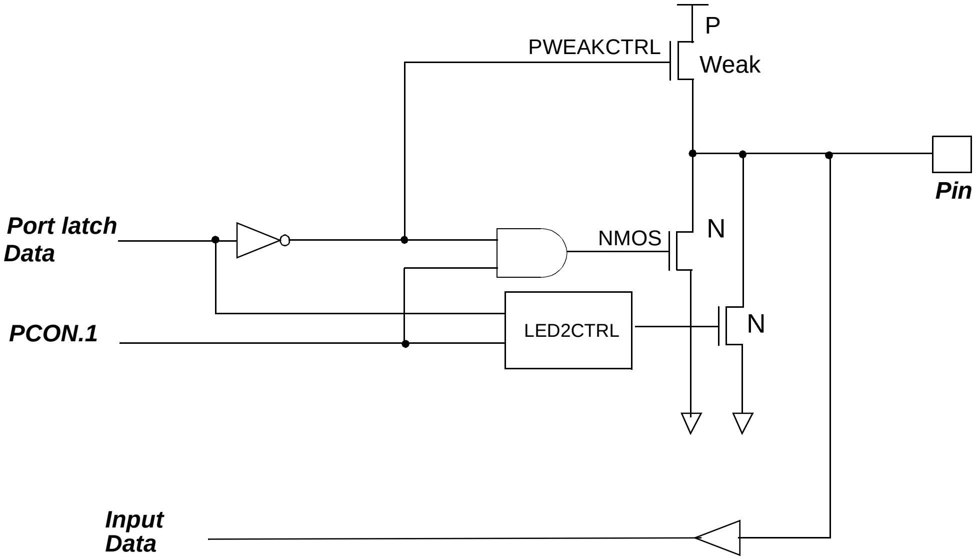


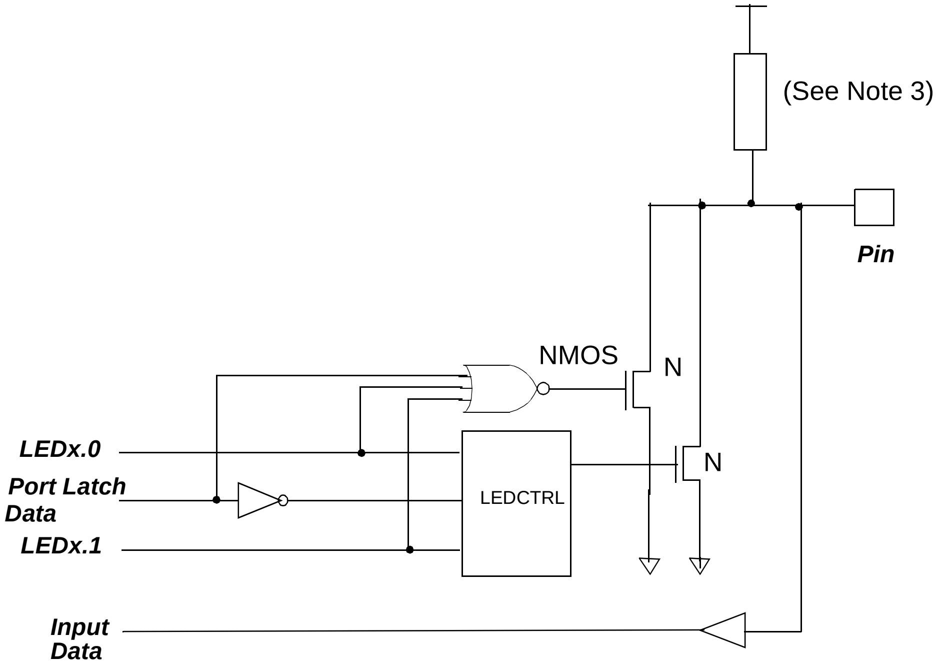
Table 7. Low Speed Output Configuration

Input Signals		Outputs Signals				
PCON.1	Port Latch Data	NMOS	LED2CTRL	PWEAKCTRL	PIN	Comments
0	0	0	1	1	0	Operating mode
0	1	0	0	0	1	
1	0	1	0	1	0	Power down mode
1	1	0	0	0	1	

LED Source Current

The LED configuration is shown in Figure 11.

Figure 11. LED Source Current



Notes:
1. When switching a low level, LEDCTRL device has a permanent current of about N mA/15 (N is 2, 4 or 8).
2. The port must be configured to be used as output by means of PMOD0 and PMOD1 registers and the level of current must be programmed by means of LEDCON0 and LEDCON1 registers before switching the led on.
3. The value of the pull-up depends on the mode that is selected to configure the port as output.

Table 8. LED Source Current

LEDx.1	LEDx.0	Port Latch Data	NMOS	PIN	Comments
0	0	0	1	0	LED control disabled
0	0	1	0	1	
0	1	0	0	0	LED mode 2 mA
0	1	1	0	1	
1	0	0	0	0	LED mode 4 mA
1	0	1	0	1	
1	1	0	0	0	LED mode 10 mA
1	1	1	0	1	

Registers

Table 9. Port Mode Register 0 - PMOD0 (91h) for AT8xC5122

7	6	5	4	3	2	1	0
P3C1	P3C0	P2C1	P2C0	CPRESRES	-	P0C1	P0C0

Bit Number	Bit Mnemonic	Description
7 - 6	P3C1-P3C0	**Port 3 Configuration bits (Applicable to P3.0, P3.1, P3.3, P3.4 only)** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with weak pull-up
5-4	P2C1-P2C0	**Port 2 Configuration bits** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with weak pull-down
3	CPRESRES	**Card Presence Pull-up resistor** Cleared to connect the internal pull-up Set to disconnect the internal pull-up
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1-0	P0C1-P0C0	**Port 0 Configuration bits** **00** C51 Standard P0 **01** Reserved **10** Output Low Speed **11** Push-pull

Reset Value = 0000 0x00b

Table 10. Port Mode Register 0 - PMOD0 (91h) for AT8xC5123

7	6	5	4	3	2	1	0
P3C1	P3C0	-	-	CPRESRES	-	-	-

Bit Number	Bit Mnemonic	Description
7 - 6	P3C1-P3C0	**Port 3 Configuration bits (Applicable to P3.0, P3.1, P3.3, P3.4 only)** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with weak pull-up
5-4		**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	CPRESRES	**Card Presence Pull-up resistor** Cleared to connect the internal pull-up Set to disconnect the internal pull-up
2-0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value = 00xx 0xxxb

Table 11. Port Mode Register 1 - PMOD1 (84h) for AT8xC5122

7	6	5	4	3	2	1	0
P5HC1	P5HC0	P5MC1	P5MC0	P5LC1	P5LC0	P4C1	P4C0

Bit Number	Bit Mnemonic	Description
7 - 6	P5HC1-P5HC0	**Port 5 High Configuration bits (Applicable from P5.6 to P5.7 only)** **00** Quasi bi-directional **01** Push-pull **10** Input with weak pull-down **11** Input with weak pull-up
5 - 4	P5MC1-P5MC0	**Port 5 Medium Configuration bits (Applicable from P5.3 to P5.5 only)** **00** Quasi bi-directional **01** Push-pull **10** Input with weak pull-down **11** Input with weak pull-up
3 - 2	P5LC1-P5LC0	**Port 5 Low Configuration bits (Applicable from P5.0 to P5.2 only)** **00** Quasi bi-directional **01** Push-pull **10** Input with medium pull-up **11** Input with weak pull-up
1 - 0	P4C1-P4C0	**Port 4 Configuration bits (Applicable from P4.3 to P4.5 only)** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with medium pull-up

Reset Value = 0000 0000b

Table 12. Port Mode Register 1 - PMOD1 (84h) for AT8xC5123

7	6	5	4	3	2	1	0
-	-	-	-	P5LC1	P5LC0	-	-

Bit Number	Bit Mnemonic	Description
7 - 4		**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3 - 2	P5LC1-P5LC0	**Port 5 Low Configuration bits (Applicable from P5.0 to P5.2 only)** **00** Quasi bi-directional **01** Push-pull **10** Input with medium pull-up **11** Input with weak pull-up
1 - 0		**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value = xxxx 00xxb

Table 13. LED Port Control Register 0 - LEDCON0 (F1h)

7	6	5	4	3	2	1	0
LED3.1	LED3.0	LED2.1	LED2.0	LED1.1	LED1.0	LED0.1	LED0.0

Bit Number	Bit Mnemonic	Description
7 - 6	LED3	**Port LED3 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.7 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.7 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.7 is configured as Quasi-bidirect. mode
5 - 4	LED2	**Port LED2 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.6 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.6 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.6 is configured as Quasi-bidirect. mode
3 - 2	LED1	**Port LED1 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.4 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.4 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.4 is configured as Quasi-bidirect. mode
1 - 0	LED0	**Port LED0 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.2 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.2 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.2 is configured as Quasi-bidirect. mode

Reset Value = 0000 0000b

Table 14. LED Port Control Register 1- LEDCON1 (F1h) only for AT8xC5122

7	6	5	4	3	2	1	0
-	-	LED6.1	LED6.0	LED5.1	LED5.0	LED4.1	LED4.0

Bit Number	Bit Mnemonic	Description
7 - 6		**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5 - 4	LED6	**Port LED6 Configuration bits** **00** LED control disabled **01** 2 mA current source when P4.5 is configured as Quasi-bi-directional mode **10** 4 mA current source when P4.5 is configured as Quasi-bi-directional mode **11** 10 mA current source when P4.5 is configured as Quasi-bidirect. mode
3 - 2	LED5	**Port LED5 Configuration bits** **00** LED control disabled **01** 2 mA current source when P4.4 is configured as Quasi-bi-directional mode **10** 4 mA current source when P4.4 is configured as Quasi-bi-directional mode **11** 10 mA current source when P4.4 is configured as Quasi-bidirect. mode
1 - 0	LED4	**Port LED0 Configuration bits** **00** LED control disabled **01** 2 mA current source when P4.3 is configured as Quasi-bi-directional mode **10** 4 mA current source when P4.3 is configured as Quasi-bi-directional mode **11** 10 mA current source when P4.3 is configured as Quasi-bidirect. mode

Reset Value = 0000 0000b

SFR Description

The Special Function Registers (SFRs) of the AT8xC5122/23 fall into the following categories:

- C51 Core Registers: ACC, B, DPH, DPL, PSW, SP
- System Configuration Registers: PCON, CKRL, CKCON0, CKCON1, CKSEL, PLLCON, PLLDIV, AUXR, AUXR1
- I/O Port Registers: P0, P1, P2, P3, P4, P5,PMOD1, PMOD2
- Timer Registers: TCON, TH0, TH1, TMOD, TL0, TL1
- Watchdog (WD) Registers: WDTRST, WDTPRG
- Serial I/O Port Registers: SADDR, SADEN, SBUF, SCON
- Baud Rate Generator (BRG) Registers: BRL, BDRCON
- System Interrupt Registers: IE0, IPL0, IPH0, IE1, IPL1, IPH1
- Smart Card Interface (SCI) Registers: SCSR, SCCON/SCETU0, SCISR/SCETU1, SCIER/SCIIR, SCTBUF/SCRBUF, SCGT0/SCWT0, SCGT1/SCWT1, SCICR/SCWT2, SCICLK
- DC/DC Converter Registers: DCCKPS
- Keyboard Interface Registers: KBE, KBF, KBLS
- Serial Port Interface (SPI) Registers: SPCON, SPSTA, SPDAT
- Universal Serial Bus (USB) Registers:USBCON, USBADDR, USBINT, USBIEN, UEPNUM, UEPCONX, UEPSTAX, UEPRST, UEPINT, UEPIEN, UEPDATX, UBYCTX, UFNUML, UFNUMH
- LED Controller Registers: LEDCON0, LEDCON1

Table 15. AT8xC5122 SFR Mapping

	Bit addressable	Not bit addressable							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	UEPINT 0000 0000								FFh
F0h	B 0000 0000	LEDCON0 0000 0000							F7h
E8h	P5 1111 1111								EFh
E0h	ACC 0000 0000	LEDCON1 XX00 0000	UBYCTX 0000 0000						E7h
D8h									DFh
D0h	PSW 0000 0000	RCON XXXX 0XXX			UEPCONX 1000 0000	UEPRST 0000 0000			D7h
C8h							UEPSTAX 0000 0000	UEPDATX 0000 0000	CFh
C0h	P4 1111 1111	SCWT3[1] 0000 0000 SCICLK[1] 0X10 1111	UEPIEN 0000 0000	SPCON 0001 0100	SPSTA 0000 0000	SPDAT 1111 1111	USBADDR 1000 0000	UEPNUM 0000 0000	C7h
B8h	IPL0 X000 000	SADEN 0000 0000	UFNUML 0000 0000	UFNUMH 0000 0000	USBCON 0000 0000	USBINT 0000 0000	USBIEN 0000 0000	DCCKPS 0000 0000	BFh
B0h	P3 1111 1111	IEN1 XXXX X000	IPL1 00XX 00X0	IPH1 00XX 00X0	SCWT0[1] 1000 0000 SCGT0 [1] 0000 1100	SCWT1 [1] 0010 0101 SCGT1[1] XXXX XXX0	SCWT2 [1] 0000 0000 SCICR [1] 0000 0000	IPH0 X000 0000	B7h
A8h	IEN0 0000 0000	SADDR 0000 0000	SCTBUF [1] 0000 0000 SCRBUF [1] 0000 0000	SCSR X000 1000	SCCON [1] 000 0000 SCETU0 [1] 0111 0100	SCISR [1] 10X0 0000 SCETU1 [1] XXXX X001	SCIIR [1] 0X00 0000 SCIER [1] 0X00 0000	CKCON1 XXXX XXX0	AFh
A0h	P2 1111 1111	ISEL 0000 0100	AUXR1 XX1X 0XX0	PLLCON XXXX X000	PLLDIV 0000 0000		WDTRST XXXX XXXX	WDTPRG XXXX X000	A7h
98h	SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000	KBLS 0000 0000	KBE 0000 0000	KBF 0000 0000		9Fh
90h	P1 1111 1111	PMOD0[2] 0000 0000						CKRL XXXX 1111	97h
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR 0XXX X000	CKCON0 X0X0 X000	8Fh
80h	P0 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000	PMOD1 0000 0000	CKSEL XXXX XXX0		PCON 00X1 0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Notes: 1. Mapping is done using SCRS bit in SCSR register: if SCRS = 0, upper cell, if SCRS = 1, lower cell.
2. Mapping is done using P/D# bit in PMOD0 register : if P/D# = 0, upper cell, if P/D# = 1, lower cell.

Note: Blank: Reserved, no write or read is allowed.

Table 16. AT8xC5123 SFR Mapping

		Bit addressable	Not bit addressable								
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F		
F8h		UEPINT 0000 0000									FFh
F0h		B 0000 0000	LEDCON0 0000 0000								F7h
E8h		P5 XXXX XXX1									EFh
E0h		ACC 0000 0000		UBYCTX 0000 0000							E7h
D8h											DFh
D0h		PSW 0000 0000				UEPCONX 1000 0000	UEPRST 0000 0000				D7h
C8h								UEPSTAX 0000 0000	UEPDATX 0000 0000		CFh
C0h	0	P4 11XX XXXX	SCWT3 0000 0000	UEPIEN 0000 0000				USBADDR 1000 0000	UEPNUM 0000 0000	0	C7h
	1		SCICLK 0X10 1111							1	
B8h		IPL0 X000 000	SADEN 0000 0000	UFNUML 0000 0000	UFNUMH 0000 0000	USBCON 0000 0000	USBINT 0000 0000	USBIEN 0000 0000	DCCKPS 0000 0000		BFh
B0h	0	P3 1111 1111	IEN1 X0XX 0XXX	IPL1 X0XX 0XXX	IPH1 X0XX 0XXX	SCWT0[1] 1000 0000	SCWT1 [1] 0010 0101	SCWT2 [1] 0000 0000	IPH0 X000 0000	0	B7h
	1					SCGT0 [1] 0000 1100	SCGT1[1] XXXX XXX0	SCICR [1] 0000 0000		1	
A8h	0	IEN0 0000 0000	SADDR 0000 0000	SCTBUF [1] 0000 0000	SCSR X000 1000	SCCON [1] 000 0000	SCISR [1] 10X0 0000	SCIIR [1] 0X00 0000	CKCON1 XXXX XXX0	0	AFh
	1			SCRBUF [1] 0000 0000		SCETU0 [1] 0111 0100	SCETU1 [1] XXXX X001	SCIER [1] 0X00 0000		1	
A0h			ISEL 0000 0100	AUXR1 XXXX 0XX0	PLLCON XXXX X000	PLLDIV 0000 0000		WDTRST XXXX XXXX	WDTPRG XXXX X000		A7h
98h		SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000						9Fh
90h		P1 1111 1111	PMOD0[2] 00XX 0XXX						CKRL XXXX 1111		97h
88h		TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR 0XXX X000	CKCON0 X0X0 X000		8Fh
80h			SP 0000 0111	DPL 0000 0000	DPH 0000 0000	PMOD1 XXXX 00XX	CKSEL XXXX XXX0		PCON 00X1 0000		87h
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F		

Notes: 1. Mapping is done using SCRS bit in SCSR register: if SCRS = 0, upper cell, if SCRS = 1, lower cell.
2. Mapping is done using P/D# bit in PMOD0 register : if P/D# = 0, upper cell, if P/D# = 1, lower cell.

Note: Blank: Reserved, no write or read is allowed.

Table 17. C51 Core Registers

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

Table 18. System Configuration Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
PCON	87h	Power Controller	SMOD1	SMOD0		POF	GF1	GF0	PD	IDL
CKCON0	8Fh	Clock Controller 0		WDX2		SIX2		T1X2	T0X2	X2
CKCON1	AFh	Clock Controller 1								SPIX2
CKSEL	85h	Clock Selection								CKS
CKRL	97h	Clock Reload Register					CKREL 3-0			
PLLCON	A3h	PLL Controller Register						EXT48	PLLEN	PLOCK
PLLDIV	A4h	PLL Divider register	R3-0				N3-0			
RCON [1]	D1h	Data Memory Configuration					RPS			
AUXR	8Eh	Auxiliary Register 0	DPU					XRS0	EXTRAM	A0
AUXR1	A2h	Auxiliary Register 1			ENBOOT[1]		GF3			DPS

Note: 1. Only for AT8xC5122

Table 19. I/O Ports Register

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P0[1]	80h	Port 0	P0							
P1	90h	Port 1	P1							
P2[1]	A0h	Port 2	P2							
P3	B0h	Port 3	P3							
P4[1]	C0h	Port 4	P4							
P5	E8h	Port 5	P5(only P5.0 for AT8xC5122)							
PMOD0	91h	Port Mode Register 0	P3C1	P3C0	P2C1[1]	P2C0[1]	CPRESRES	-	P0C1[1]	P0C0[1]
PMOD1	84h	Port Mode Register 1	P5HC1[1]	P5HC0[1]	P5MC1[1]	P5MC0[1]	P5LC1	P5LC0	P4C1[1]	P4C0[1]

Note: 1. Only for AT8xC5122

Table 20. Timer Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
TH0	8Ch	Timer/Counter 0 High byte	TH0							
TL0	8Ah	Timer/Counter 0 Low byte	TL0							
TH1	8Dh	Timer/Counter 1 High byte	TH1							
TL1	8Bh	Timer/Counter 1 Low byte	TL1							
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

Table 21. Watchdog Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
WDTRST	A6h	Watchdog Timer Reset	WDTRST							
WDTPRG	A7h	Watchdog Timer Program						S2-0		

Table 22. Serial I/O Port Registers

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

Table 23. Baud Rate Generator (BRG) Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
BRL	9Ah	Baud Rate Reload	BRL							
BDRCON	9Bh	Baud Rate Control				BRR	TBCK	RBCK	SPD	SRC

Table 24. System Interrupt Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
IEN0	A8h	Interrupt Enable Control 0	EA			ES	ET1	EX1	ET0	EX0
IEN1	B1h	Interrupt Enable Control 1		EUSB			ESCI	ESPI(1)		EKB(1)
IPL0	B8h	Interrupt Priority Control Low 0				PSL	PT1L	PX1L	PT0L	PX0L
IPH0	B7h	Interrupt Priority Control High 0				PSH	PT1H	PX1H	PT0H	PX0H
IPL1	B2h	Interrupt Priority Control Low 1		PUSBL			PSCIL	PSPIL(1)		PKBL(1)
IPH1	B3h	Interrupt Priority Control High 1		PUSBH			PSCIH	PSPIH(1)		PKBH(1)
ISEL	A1h	Interrupt Enable Register	CPLEV		PRESIT	RXIT	OELEV	OEEN	PRESEN	RXEN

Note: 1. Only for AT8xC5122

Table 25. Smart Card Interface (SCI) Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SCGT0	B4h	Smart Card Transmit Guard Time Register 0	GT7 - 0							
SCGT1	B5h	Smart Card Transmit Guard Time Register 1								GT8
SCWT0	B4h	Smart Card Character/ Block Wait Time Register 0	WT7 - 0							
SCWT1	B5h	Smart Card Character/ Block Wait Time Register 1	WT15-8							
SCWT2	B6h	Smart Card Character/ Block Wait Time Register 2	WT23-16							
SCWT3	C1h	Smart Card Character/ Block Wait Time Register 3	WT31-24							
SCICR	B6h	Smart Card Interface Control Register	RESET	CARDDET	VCARD1-0		UART	WTEN	CREP	CONV
SCCON	ACh	Smart Card Interface Contacts Register	CLK		CARDC8	CARDC4	CARDIO	CARDCLK	CARDRST	CARDVCC
SCETU0	ACh	Smart Card ETU Register 0	ETU7 - 0							
SCETU1	ADh	Smart Card ETU Register 1	COMP					ETU10-8		
SCISR	ADh	Smart Card UART Interface Status Register (Read only)	SCTBE	CARDIN	ICARDOVF	VCARDOK	SCWTO	SCTC	SCRC	SCPE
SCIIR	AEh	Smart Card UART Interrupt Identification Register (Read only)	SCTBI		ICARDERR	VCARDERR	SCWTI	SCTI	SCRI	SCPI
SCIER	AEh	Smart Card UART Interrupt Enable Register	ESCTBI		ICARDER	EVCARDER	ESCWTI	ESCTI	ESCRI	ESCPI
SCSR	ABh	Smart Card Selection Register		BGTEN		CREPSEL	ALTKPS1-0		SCCLK1	SCRS
SCTBUF	AAh	Smart Card Transmit Buffer Register (Write only)	Can store a new byte to be transmitted on the I/O pin when SCTBE is set. Bit ordering on the I/O pin depends on the convention							
SCRBUF	AAh	Smart Card Receive Buffer Register (Read Only)	Provides the byte received from the I/O pin when SCRI is set. Bit ordering on the I/O pin depends on the convention.							
SCICLK	C1h	Smart Card Frequency Prescaler Register	XTSCS[1]		SCICLK5-0					

Note: 1. Only for AT8xC5122

Table 26. DC/DC Converter Register

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
DCCKPS	BFh	DC/DC Converter Reload Register	MODE	OVFADJ	BOOST[1-0]		DCCKPS3-0			

Table 27. Keyboard Interface Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
KBF[1]	9Eh	Keyboard Flag Register	KBE7 - 0							
KBE[1]	9Dh	Keyboard Input Enable Register	KBF7 - 0							

Table 27. Keyboard Interface Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
KBLS[1]	9Ch	Keyboard Level Selector Register	KBLS7 - 0							

Note: 1. Only for AT8xC5122

Table 28. Serial Port Interface (SPI) Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SPCON[1]	C3h	Serial Peripheral Control	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
SPSTA[1]	C4h	Serial Peripheral Status-Control	SPIF	WCOL		MODF				
SPDAT[1]	C5h	Serial Peripheral Data	R7 - 0							

Notes: 1. Only for AT8xC5122

Table 29. Universal Serial Bus (USB) Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
USBCON	BCh	USB Global Control	USBE	SUSPCLK	SDRMWUP	DETACH	UPRSM	RMWUPE	CONFG	FADDEN
USBADDR	C6h	USB Address	FEN	UADD6-0						
USBINT	BDh	USB Global Interrupt			WUPCPU	EORINT	SOFINT			SPINT
USBIEN	BEh	USB Global Interrupt Enable			EWUPCPU	EEORINT	ESOFINT			ESPINT
UEPNUM	C7h	USB Endpoint Number					EPNUM3-0			
UEPCONX	D4h	USB Endpoint X Control	EPEN	NAKIEN	NAKOUT	NAKIN	DTGL	EPDIR	EPTYPE1	EPTYPE0
UEPSTAX	CEh	USB Endpoint X Status	DIR	RXOUTB1	STALLRQ	TXRDY	STL/CRC	RXSETUP	RXOUTB0	TXCMP
UEPRST	D5h	USB Endpoint Reset		EP6RST	EP5RST	EP4RST	EP3RST	EP2RST	EP1RST	EP0RST
UEPINT	F8h	USB Endpoint Interrupt		EP6INT	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT
UEPIEN	C2h	USB Endpoint Interrupt Enable		EP6INTE	EP5INTE	EP4INTE	EP3INTE	EP2INTE	EP1INTE	EP0INTE
UEPDATX	CFh	USB Endpoint X Fifo Data	FDAT7 - 0							
UBYCTX	E2h	USB Byte Counter Low (EPX)		BYCT6-0						
UFNUML	BAh	USB Frame Number Low	FNUM7 - 0							
UFNUMH	BBh	USB Frame Number High			CRCOK	CRCERR		FNUM10-8		

Table 30. LED Controller Registers

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
LEDCON0	F1h	LED Control 0	LED3		LED2		LED1		LED0	
LEDCON1[1]	E1h	LED Control 1			LED6		LED5		LED4	

Note: 1. Only for AT8xC5122

Clock Controller

The clock controller is based on an on-chip oscillator feeding an on-chip Phase Lock Loop (PLL). All the internal clocks to the peripherals and CPU core are generated by this controller.

The AT8xC5122/23 XTAL1 and XTAL2 pins are the input and the output of a single-stage on-chip inverter (see Figure 12), which can be configured with off-chip components as a Pierce oscillator (see Figure 14). Value of capacitors and crystal characteristics are detailed in the Section “DC Characteristics” of the AT8xC5122/23 datasheet.

The XTAL1 pin can also be used as input for an external 48 MHz clock.

The clock controller outputs several different clocks as shown in Figure 12:

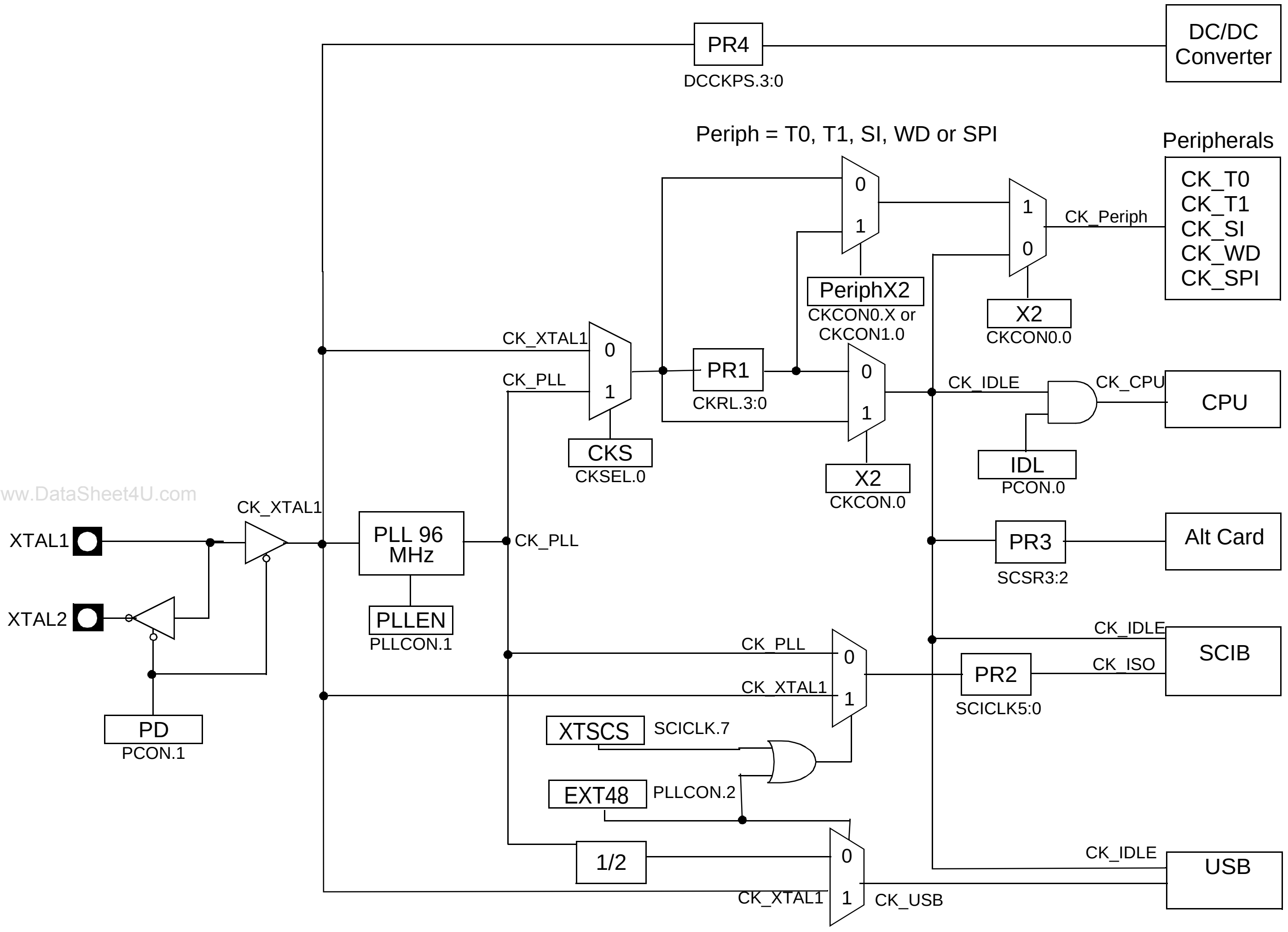
- a clock for the CPU core
- a clock for the peripherals which is used to generate the timers, watchdog, SPI, UART, and ports sampling clocks. This divided clock will be used to generate the alternate card clock.
- a clock for the USB
- a clock for the SCIB controller
- a clock for the DC/DC converter

These clocks are enabled or not depending on the power reduction mode as detailed in Section “Power Management”, page 142.

These clocks are generated using four presacalers defined in the table below:

Prescaler	Register	Reload Factor	Function
PR1	CKRL	CKRL0-3	CPU & Peripheral clocks
PR2	SCICLK	SCICLK0-5	Smart card
PR3	SCSR	ALTKPS0-1	Alternate card
PR4	DCCKPS	DCCKPS3:0	DC/DC

Figure 12. Oscillator Block Diagram



CPU and Peripheral Clock

Two clocks sources are available for CPU and peripherals:

- Crystal oscillator on XTAL1 and XTAL2 pins and a 96 MHz PLL (Set by SFR to this value).
- External 48 MHz clock on XTAL1 pin

These clock sources are adapted by the PR1 prescaler to generate the CPU core CK_CPU and the peripheral clocks:

- CK_IDLE for alternate card and peripherals registers access
- CK_T0 for Timer 0
- CK_T1 for Timer 1
- CK_SI for the UART
- CK_WD for the Watchdog
- CK_SPI for SPI

The CPU and peripherals clocks frequencies are defined in the table below.

CKS	X2	F_{CK_CPU} and F_{CK_IDLE}
0	0	F_{CK_XTAL1}/(2*(16-CKRL))
0	1	F_{CK_XTAL1}
1	0	F_{CK_PLL}/(2*(16-CKRL))
1	1	Not allowed

CK_ISO and CK_CPU selection

Two conditions must be present for an optimal work of the SCIB:

- CK_CPU > 4/3 * CK_ISO and
- CK_CPU < 6 * CK_ISO.

If the CK_CPU <= 4/3 * CK_ISO, the SCIB doesn’t work.

If the CK_CPU >= 6* CK_ISO, the programmer must take care in three cases:

- Read (or write) operation on a SCIB register followed immediatly with an other Read (or write) operation on the same register.
- Read (or write) operation on a SCIB register followed immediatly with an other Read (or write) operation on a linked register. The list of linked registers is in the table below.

Linked registers
Write in SCICR and after read of SCETU0-1
Write in SCTBUF and after read of SCISR

- Write operation on a register of the list below followed immediatly with a read operation on a SCIB register.

Wait after Write operation on this registers
SCICR, SCIER, SCETU0-1,SCGT0-1, SCWT0-3,SCCON

To avoid any trouble, a delay must be added between the two accesses on the SCIB register. The SCIB must complete the first read (or write) operation before to receive the second. A solution is to add NOP (no operation) instructions. The number of NOP to add depends of the rate between CK_CPU and CK_ISO (see table below).

min CLK_CPU	max CLK_CPU	Number of CPU cycles to add
CLK_CPU >= 6 * CLK_ISO	CLK_CPU <= 12 * CLK_ISO	6 (example1 NOP)
CLK_CPU >= 12* CLK_ISO	CLK_CPU <= 16 * CLK_ISO	12 (example 2 NOP)

Smart Card Interface Block

The Smart Card Interface Block (SCIB) uses two clock trees:

- The first one, CK_IDLE, is the peripheral clock used for the interface with the microcontroller.
- The second one, CK_ISO, is independant from the CPU clock and is generated from the PLL output. PR2, a 6-bit prescaler, will be used to generate: 12/9.6/8/6.85/6/5.33/4.8/4.36/ ..../1MHz frequencies. SCIB clock must be lower than CPU clock.

During SCI Reset, the CK_ISO input must be in the range 1 - 5 MHz according to ISO 7816. The SCIB clocks frequency is defined in Figure 27: Prescaler 2 Description and Table 37 on page 43.

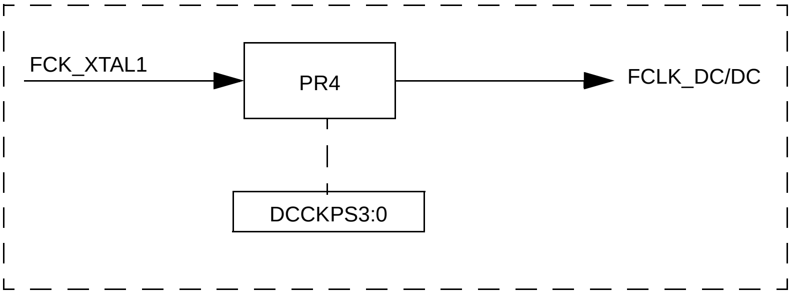
Alternate Card Clock

The alternate Card uses the peripheral clock divided by the PR3 prescaler. (1; 1/2; 1/4; 1/8 division ratio). See Section "Alternate Card", page 43 for the definition of the alternate clock.

DC/DC Clock

The DC/DC block needs a clock with a 50% duty cycle. The frequency must also respect a value between 3.68 MHz and 6MHz. The PR4 prescaler is used to comply with the DC/DC frequency requirement.

Figure 13. Functional Block Diagram



Before supplying the DC/DC block, the oscillator clock is adapted to the clock needed by the DC/DC converter. This factor is controlled with the DCCKPS3:0 register.

Examples of factors are shown in the following table:

XTAL1 (MHz)	DCCKPS3:0 value	Prescaler Factor	DC/DC converter CLK (MHz)
8	0	2	4

USB Clock

The USB Interface Block use two clock trees:

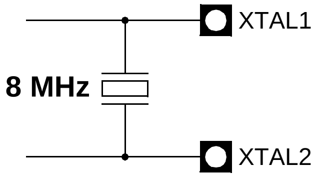
- The first one is the CPU clock used for the interface with the microcontroller, CK_IDLE.
- The second one is the USB clock, CK_USB. Since the USB frequency must be 48 MHz, a 96 MHz PLL with a by 2 divider has to be used. An external frequency can also be used.

Oscillator

Two clock sources are available for CPU:

- Crystal oscillator on XTAL1 and XTAL2 pins: Up to 8 MHz
- External 48 MHz clock on XTAL1 pin

Figure 14. Crystal Connection



PLL

PLL Description

The AT8xC5122 PLL is used to generate internal high frequency clock synchronized with an external low-frequency. Figure 15 shows the internal structure of the PLL.

The PFLD block is the Phase Frequency Comparator and Lock Detector. This block makes the comparison between the reference clock coming from the N divider and the reverse clock coming from the R divider and generates some pulses on the Up or Down signal depending on the edge position of the reverse clock. The PLLEN bit in PLLCON register is used to enable the clock generation. When the PLL is locked, the bit PLOCK in PLLCON register is set.

The CHP block is the Charge Pump that generates the voltage reference for the VCO by injecting or extracting charges from the external filter connected on PLLF pin (see Figure 16). Value of the filter components are detailed in the Section “DC Characteristics”.

The VCO block is the Voltage Controlled Oscillator controlled by the voltage V_{REF} produced by the charge pump. It generates a square wave signal: the PLL clock.The CK_PLL frequency is defined by the follwing formula:

$F_{CK\_PLL} = F_{CK\_XTAL1} * (R+1) / (N+1)$

Figure 15. PLL Block Diagram and Symbol

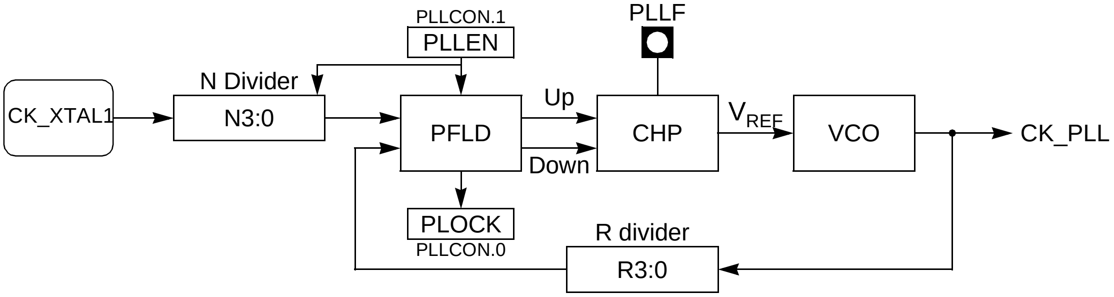
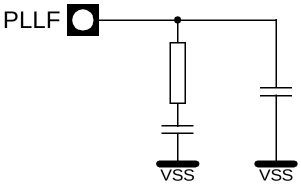


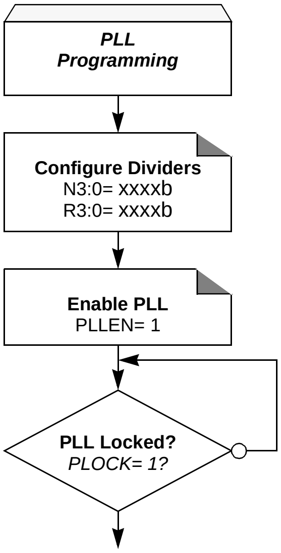
Figure 16. PLL Filter Connection



PLL Programming

The PLL is programmed V_{REF} using the flow showed in Figure 17. As soon as clock generation is enabled, user must wait until the lock indicator is set to ensure the clock output is stable.

Figure 17. PLL Programming Flow



Registers

Table 31. Clock Selection Register - CKSEL (S:85h)

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

Bit Number	Bit Mnemonic	Description
7:1	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	CKS	**CPU Oscillator Select Bit** Set this bit to connect CPU and Peripherals to PLL output. Clear this to to connect CPU and Peripherals to XTAL1 clock input.

Reset Value = XXXX XXX0b

Table 32. Clock Reload Register - CKRL (S:97h)

7	6	5	4	3	2	1	0
-	-	-	-	CKRL3	CKRL2	CKRL1	CKRL0

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3:0	CKRL3:0	**Clock Reload register** Prescaler1 value F_{ck_cpu} =[1 / 2*(16-CKRL)] * F_{ck_XTAL1}

Reset Value = XXXX 1111b

Table 33. Clock Configuration Register 1 - CKCON1 (S:AFh) only for AT8xC5122

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

Bit Number	Bit Mnemonic	Description
7 - 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.
0	SPIX2	**SPI clock** This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect. Cleared to bypass the PR1 prescaler. Set to select the PR1 output for this peripheral.

Reset Value = XXXX XXX0b

Table 34. Clock Configuration Register 0 - CKCON0 (S:8Fh)

7	6	5	4	3	2	1	0
-	WDX2	-	SIX2	-	T1X2	T0X2	X2

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
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 bypass the PR1 prescaler. Set to select the PR1 output for this peripheral.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	SIX2	**Enhanced UART 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 bypass the PR1 prescaler. Set to select the PR1 output for this peripheral.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	T1X2	**Timer 1 clock** This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect. Cleared to bypass the PR1 prescaler. Set to select the PR1 output for this peripheral.
1	T0X2	**Timer 0 clock** This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect. Cleared to bypass the PR1 prescaler. Set to select the PR1 output for this peripheral.
0	X2	**System clock Control bit** Cleared to select the PR1 output for CPU and all the peripherals . Set to bypass the PR1 prescaler and to enable the individual peripherals ‘X2’ bits.

Reset Value = X0X0 X000b

Table 35. PLL Control Register - PLLCON (S:A3h)

7	6	5	4	3	2	1	0
-	-	-	-	-	EXT48	PLLEN	PLOCK

Bit Number	Bit Mnemonic	Description
7 - 3	-	**Reserved** The value read from these bits is always 0. Do not set this bits.
2	EXT48	**External 48 MHz Enable Bit** Set this bit to select XTAL1 as USB clock. Clear this bit to select PLL as USB clock. SCIB clock is controlled by EXT48 bit and XTSCS bit.
1	PLLEN	**PLL Enable bit** Set to enable the PLL. Clear to disable the PLL.
0	PLOCK	**PLL Lock Indicator** Set by hardware when PLL is locked Clear by hardware when PLL is unlocked

Reset Value = 0000 0000b

Table 36. PLL Divider Register - PLLDIV (S:A4h)

7	6	5	4	3	2	1	0
R3	R2	R1	R0	N3	N2	N1	N0

Bit Number	Bit Mnemonic	Description
7 - 4	R3:0	PLL R Divider Bits
3 - 0	N3:0	PLL N Divider Bits

Reset Value = 0000 0000b

Smart Card Interface Block (SCIB)

The SCIB provides all signals to interface directly with a smart card. The compliance with the ISO7816, EMV'2000, GSM and WHQL standards has been certified.

Both synchronous (e.g. memory card) and asynchronous smart cards (e.g. microprocessor card) are supported. The component supplies the different voltages requested by the smart card. The power off sequence is directly managed by the SCIB.

The card presence switch of the smart card connector is used to detect card insertion or card removal. In case of card removal, the SCIB de-activates the smart card using the de-activation sequence. An interrupt can be generated when a card is inserted or removed.

Any malfunction is reported to the microcontroller (interrupt + control register).

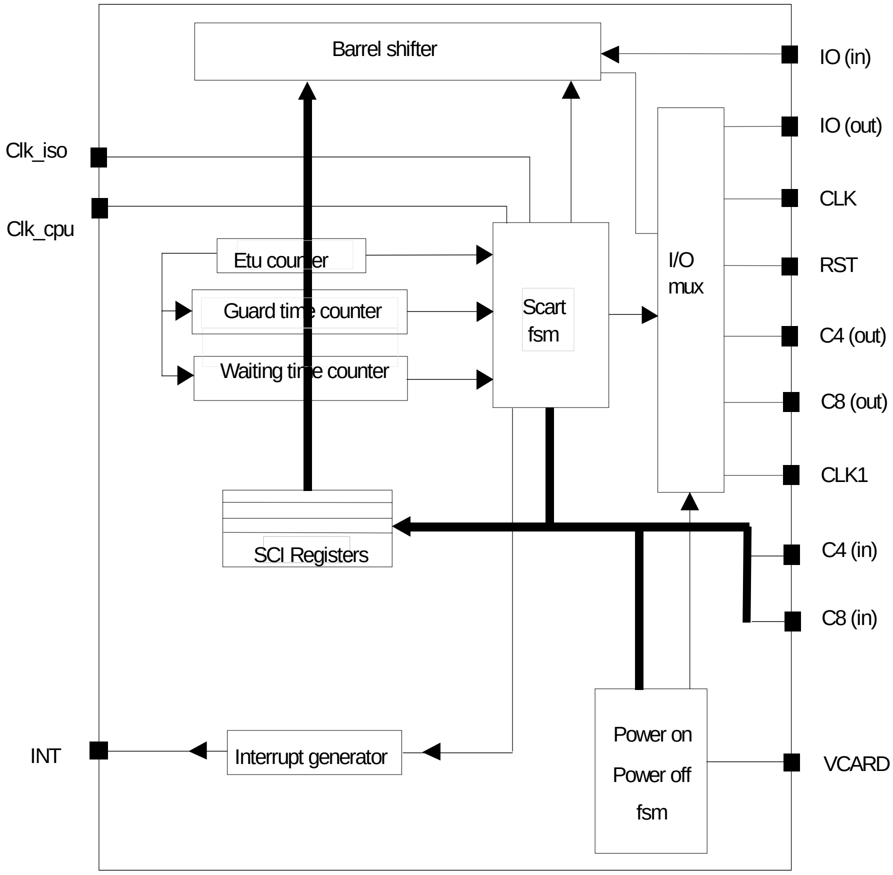
The different operating modes are configured by internal registers.

- Support of ISO/IEC 7816
- character mode
- one transmit buffer + one receive buffer
- 11 bits ETU counter
- 9 bits guard time counter
- 32 bits waiting time counter
- Auto character repetition on error signal detection in transmit mode
- Auto error signal generation on parity error detection in receive mode
- Power on and power off sequence generation
- Manual mode to drive directly the card I/O

Block Diagram

The Smart Card Interface Block diagram is shown Figure 18:

Figure 18. SCIB Block Diagram



Functional Description

The architecture of the Smart Card Interface Block can be detailed as follows:

Barrel Shifter

The Barrel Shifter allows the translation between 1 bit serial data and 8 bits parallel data

The barrel function is useful for character repetition since the character is still present in the shifter at the end of the character transmission.

This shifter is able to shift the data in both directions and to invert the input or output value in order to manage both direct and inverse ISO7816-3 convention.

Coupled with the barrel shifter there is a parity checker and generator.

There are 2 registers connected to this barrel shifter, one for the transmission and one for the reception. They act as buffers to relieve the CPU of timing constraints.

SCART FSM

(Smart Card Asynchronous Receiver Transmitter Finite State Machine)

This is the core of the design. Its purpose is to control the barrel shifter. To sequence correctly the barrel shifter for a reception or a transmission, it uses the signals issued by the different counters. One of the most important counters is the guard time counter that gives time slots corresponding to the character frame.

The SCART FSM is enabled only in UART mode.

The transition from the receipt mode to the transmit mode is done automatically. Priority is given to the transmission.

ETU Counter

The ETU (Elementary Timing Unit) counter controls the working frequency of the barrel shifter, in fact it generates the enable signal of the barrel shifter.

The ETU is 11 bits wide and there is a special compensation mode activated with the most significant bit that allows non integer ETU value with a working clock equal to the card clock (CK_ISO). But the decimal value is limited to a half clock cycle. In fact the bit duration is not fixed. It takes turns in n clock cycles and n-1 clock cycles.The character duration (10 bits) is also equal to 10*(n + 1/2) clock cycles.

This allows to reach the required precision of the character duration specified by the ISO7816 standard.

example: F=372 D=32 => ETU=11.625 clock cycles.

ETU = (ETU[10-0] -0.5 * COMP) / f iso with ETU[10-0] = 12, COMP = 1 (bit 7 of SCETU1)

To achieve this clock rate, we activated the compensation mode and we programmed the ETU duration to 12 clock cycles.

The result will be a full character duration (10 bits) equal to 11.5 clock cycles.

Guard Time and Block Guard Time Counters

The minimum time between the leading edge of the start bit of a character and the leading edge of the start bit of the following character transmitted (Guard time) is controlled by one counter, as described in Figure 19.

The minimum time between the leading edge of the start bit of the last received character and the first character transmitted (Block guard time) is controlled by another counter. The bit BGTEN in SCSR register must be set to use this functionality. The transfer of GT[8-0] value to the BGT counter is done on the rising edge of the BGTEN. They are 9 bits wide and are incremented at the ETU rate.

Figure 19. Guard Time and Block Guard Time.

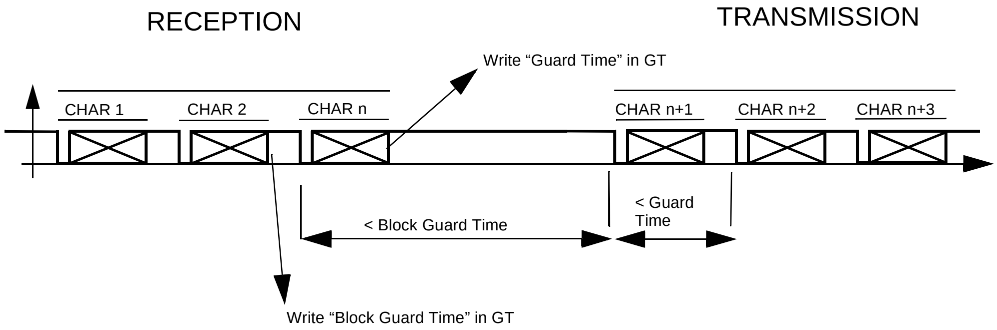
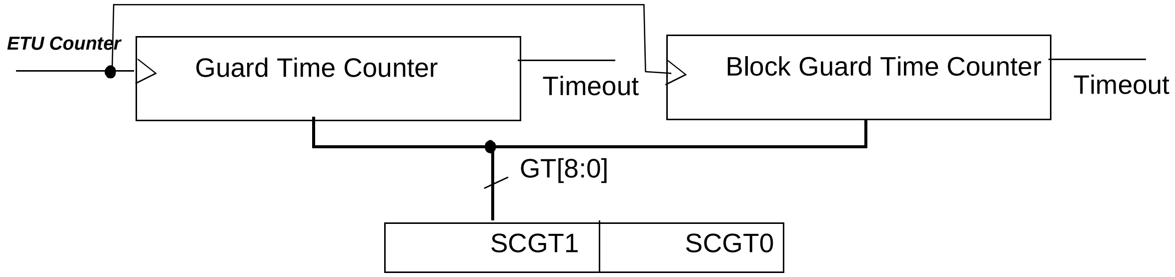


Figure 20. Guard Time and Block Guard Time counters

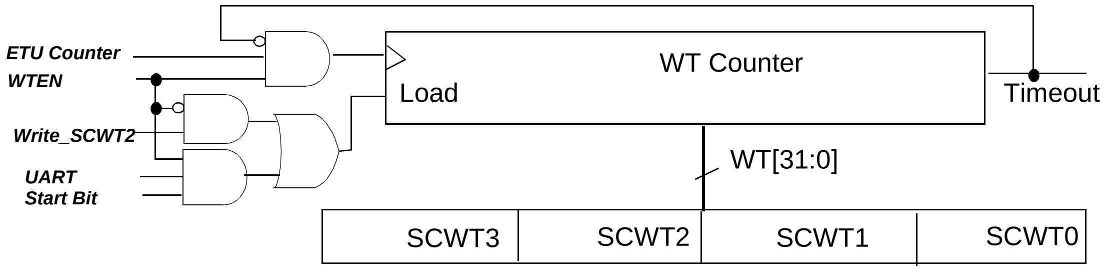


Waiting Time (WT) Counter

The WT counter is a 32 bits down counter which can be loaded with the value contained in the SCWT3, SCWT2, SCWT1, SCWT0 registers. Its main purpose is timeout signal generation. It is 32 bits wide and is decremented at the ETU rate. The ETU counter acts as a prescaler: see Figure 21.

When the WT counter times out, an interrupt is generated and the SCIB function is locked: reception and emission are disabled. It can be enabled by resetting the macro or reloading the counter.

Figure 21. Waiting Time Counter



The counter is loaded, if WTEN = 0, during the write of SCWT2 register.

This counter is available in both UART and manual modes. But the behavior depends on the selected mode.

In manual mode, the WTEN signal controls the start of the counter (rising edge) and the stop of the counter (falling edge). After a timeout of the counter, a falling edge on WTEN, a reload of SCWT2 and a rising edge of WTEN are necessary to start again the counter and to release the SCIB macro. The reload of SCWT2 transfers all SCWT0, SCWT1, SCWT2 and SCWT3 registers to the WT counter.

In UART mode there is an automatic load on the start bit detection. This automatic load is very useful for changing on-the-fly the timeout value since there is a register to hold the load value. That is the case, for example when in T = 1 we have to launch the BWT timeout on the start bit of the last transmitted character. But on the receipt of the first character an other timeout value (CWT) must be used. For this, the new load value of the waiting time counter must be loaded with CWT value before the transmission of the last character. The reload of SCWT[3-0] with the new value occurs with WTEN = 1.

After a timeout of the counter in UART mode, the restart is done as in manual mode.

The maximum interval between the start leading edge of a character and the start leading edge of the next character is loaded in the SCWT3, SCWT2, SCWT1, SCWT0 registers (see Figure 21).

In T=1 mode, the CWT (character waiting time) or the BWT (block waiting time) are loaded in the same registers.

The maximum time between two consecutive start bit is WT[31:0] * ETU.

When used to check BWT according to ISO 7816, WT can be set between 971 and 15728651.

The WT counter is 32 bits wide in order to handle the BWT extension. In this case, WT must be loaded with the value BWT * WTX.

Figure 22. T=0 mode

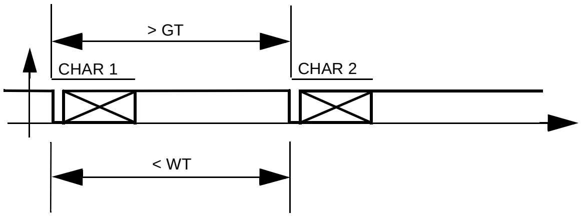
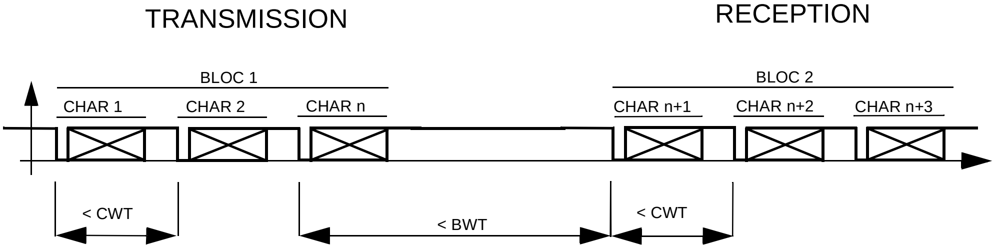


Figure 23. T=1 Mode



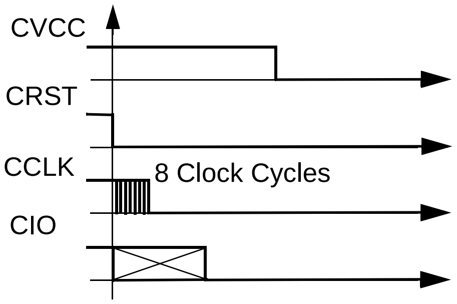
Power-on and Power-off FSM

In this state, the machine applies the signals on the smart card in accordance with ISO7816 standard.

To be able to power on the SCIB, the card presence is mandatory. Removal of the smart card will automatically start the power off sequence as described in Figure 24.

The SCI deactivation sequence after a reset of the CPU or after a lost of power supply is ISO7816 compliant. The switching order of the signals is the same as in Figure 24 but the delay between signals is analog and not clock dependant.

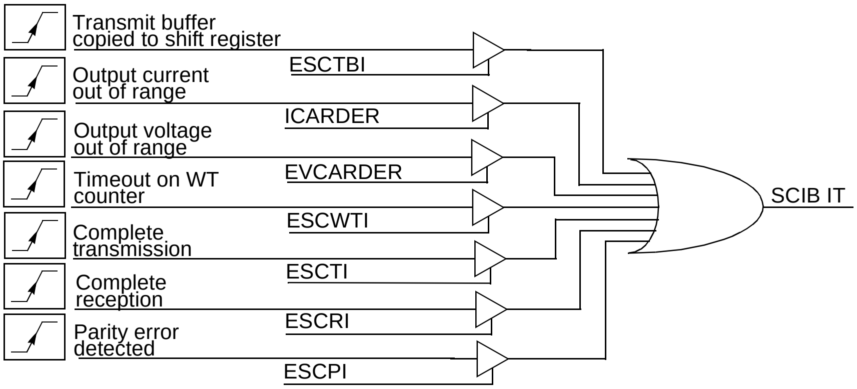
Figure 24. SCI Deactivation Sequence after a Card Extraction



Interrupt Generator

There are several sources of interruption but the SCIB macro-cell issues only one interrupt signal: SCIBIT.

Figure 25. SCIB Interrupt Sources



This signal is high level active. One of the sources is able to set up the INT signal and this is the read of the Smart Card Interrupt register by the CPU that clears this signal.

If during the read of the Smart Card Interrupt register an interrupt occurs, the set of the corresponding bit into the Smart Card Interrupt register and the set of the INT signal will be delayed after the read access.

Registers

There are fifteen registers to control the SCIB macro-cell. They are described in Table 54 to Table 45.

Some of the register widths are greater than a byte. Despite the 8 bits access provided by the BIU, the address mapping of this kind of register respects the following rule:

The Low significant byte register is implemented at the higher address.

This implementation makes access to these registers easier when using high level programming language (C,C++).

Additional Features

Clock

The CK_ISO input must be in the range 1 - 5 MHz according to ISO 7816.

The CK_ISO can be programmed up to 12 MHz. In this case, the timing specification of the output buffer will not comply to ISO 7816.

Figure 26. Clock Diagram of the SCIB Block

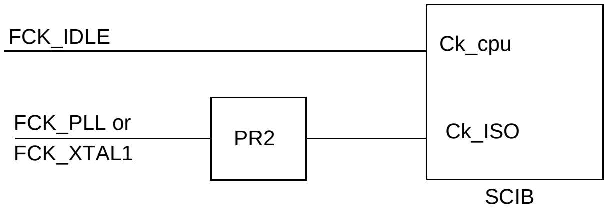
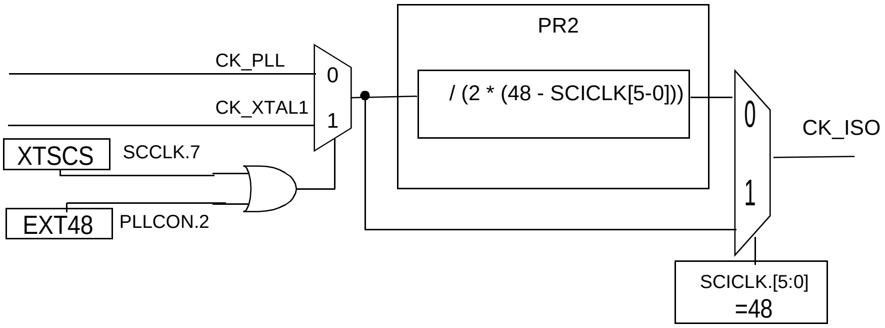


Figure 27. Prescaler 2 Description



The division factor SCICLK must be smaller than 49. If it is greater or equal to 49, the PR2 prescaler is locked.

Table 37. Examples of Settings Ffor Clocks

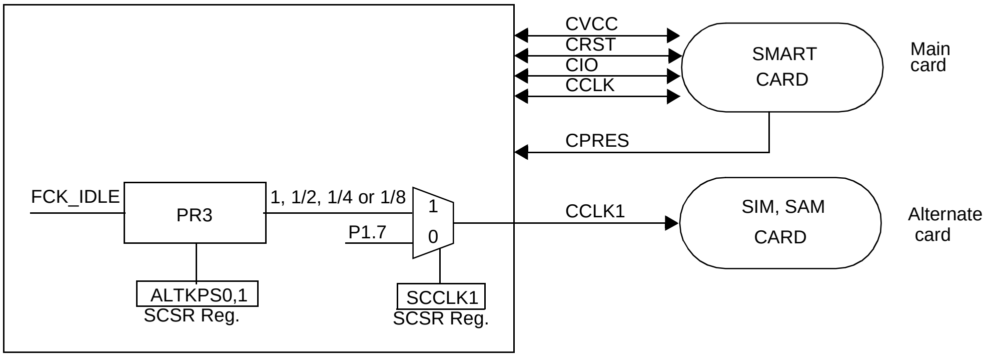
XTAL1 (MHz)	EXT48	SCICLK	CK_ ISO
48	1	42	4
8	0	36	4
8	0	44	12
8	0	42	8
8	0	40	6
8	0	24	2
8	0	0	1

Alternate Card

A second card named ‘Alternate Card’ can be controlled.

The Clock signal CCLK1 can be adapted to the XTAL frequency. Thanks to the clock prescaler which can divide the frequency by 1, 2, 4 or 8. The bits ALTKPS0 and ALTKPS1 in SCSR Register are used to set this factor.

Figure 28. Alternate Card



Card Presence Input

The internal pull-up (weak pull-up) on Card Presence input can be disconnected in order to reduce the consumption (CPRESRES, bit 3 in PMOD0).

In this case, an external resistor (typically 1 MΩ) must be externally tied to Vcc.

CPRES input can generate an interrupt (see Interrupt system section).

The detection level can be selected.

SCIB Reset

The SCICR register contains a reset bit. If set, this bit generates a reset of the SCI and its registers. Table 38 defines the SCIB registers that are reset and their reset values.

Table 38. Reset Values for SCI Registers

Register Name	SCIB Reset Value (Binary)
SCICR	0000 0000
SCCON	0X00 0000
SCISR	1000 0000
SCIIR	0X00 0000
SCIER	0X00 0000
SCSR	X000 1000
SCTBUF	0000 0000
SCRBUF	0000 0000
SCETU1, SCETU0	XXXX X001, 0111 0100 (372)
SCGT1, SCGT0	0000 0000, 0000 1100 (12)
SCWT3, SCWT2, SCWT1, SCWT0	0000 0000, 0000 0000, 0010 0101, 1000 0000 (9600)
SCICLK	0X10 1111

Registers

Table 39. Smart Card Interface Control Register - SCICR (S:B6h, SCRS = 1)

7	6	5	4	3	2	1	0
RESET	CARDDET	VCARD1	VCARD0	UART	WTEN	CREP	CONV

Bit Number	Bit Mnemonic	Description
7	RESET	**Reset** Set this bit to reset and deactivate the Smart Card Interface. Clear this bit to activate the Smart Card Interface. This bit acts as an active high software reset.
6	CARDDET	**Card Presence Detector Sense** Clear this bit to indicate the card presence detector is open when no card is inserted (CPRES is high). Set this bit to indicate the card presence detector is closed when no card is inserted (CPRES is low).
5-4	VCARD[1:0]	**Card Voltage Selection:** VCARD[1] VCARD[0] CVCC 0 0 0.0V 0 1 1.8V 1 0 3.0V 1 1 5.0V
3	UART	**Card UART Selection** Clear this bit to use the Card I/O bit to drive the Card I/O pin. Set this bit to use the Smart Card UART to drive the Card I/O pin. Controls also the Wait Time Counter as described in Section “Waiting Time (WT) Counter”, page 40
2	WTEN	**Wait Time counter Enable** Clear this bit to stop the counter and enable the load of the Wait Time counter hold registers. The hold registers are loaded with SCWT0, SCWT1, SCWT2 and SCWT3 values when SCWT2 is written. Set this bit to start the Wait Time counter. The counters stop when it reaches the timeout value. If the UART bit is set, the Wait Time counter automatically reloads with the hold registers whenever a start bit is sent or received.
1	CREP	**Character Repetition** Clear this bit to disable parity error detection and indication on the Card I/O pin in receive mode and to disable character repetition in transmit mode. Set this bit to enable parity error indication on the Card I/O pin in receive mode and to set automatic character repetition when a parity error is indicated in transmit mode. In receive mode, three times error indication is performed and the parity error flag is set after four times parity error detection. In transmit mode, up to three times character repetition is allowed and the parity error flag is set after five times (reset configuration, can be set at 4 using CREPSET bit in SCSR Register) consecutive parity error indication.
0	CONV	**ISO Convention** Clear this bit to use the direct convention: b0 bit (LSB) is sent first, the parity bit is added after b7 bit and a low level on the Card I/O pin represents a’0’. Set this bit to use the inverse convention: b7 bit (LSB) is sent first, the parity bit is added after b0 bit and a low level on the Card I/O pin represents a’1’.

Reset Value = 0000 0000b

Table 40. Smart Card Contacts Register - SCCON (S:ACh, SCRS=0)

7	6	5	4	3	2	1	0
CLK	-	CARDC8	CARDC4	CARDIO	CARDCLK	CARDRST	CARDVCC

Bit Number	Bit Mnemonic	Description
7	CLK	**Card Clock Selection** Clear this bit to use the Card CLK bit (CARDCLK) to drive Card CLK pin. Set this bit to use XTAL or PLL signal to drive the Card CLK pin. **Note:** internal synchronization avoids glitches on the CLK pin when switching this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
5	CARDC8	**Card C8** Clear this bit to drive a low level on the Card C8 pin. Set this bit to set a high level on the Card C8 pin. The CC8 pin can be used as a pseudo bi-directional I/O when this bit is set.
4	CARDC4	**Card C4** Clear this bit to drive a low level on the Card C4 pin. Set this bit to set a high level on the Card C4 pin. The CC4 pin can be used as a pseudo bi-directional I/O when this bit is set.
3	CARDIO	**Card I/O** When the UART bit is cleared in Registers, the value of this bit is driven to the Card I/O pin. Then this pin can be used as a pseudo bi-directional I/O when this bit is set. To be used as an input, this bit must contain a 1.
2	CARDCLK	**Card CLK** When the CLK bit is cleared in SCCON Register, the value of this bit is driven to the Card CLK pin.
1	CARDRST	**Card RST** Clear this bit to drive a low level on the Card RST pin. Set this bit to set a high level on the Card RST pin.
0	CARDVCC	**Card VCC Control** Clear this bit to desactivate the Card interface and set its power-off. The other bits of SCCON register have no effect while this bit is cleared. Set this bit to power-on the Card interface. The activation sequence should be handled by software.

Reset Value = 0X00 0000b

Table 41. Smart Card UART Interface Status Register - SCISR (S:ADh, SCRS=0)

7	6	5	4	3	2	1	0
SCTBE	CARDIN	ICARDOVF	VCARDOK	SCWTO	SCTC	SCRC	SCPE

Bit Number	Bit Mnemonic	Description
7	SCTBE	**SCIB Transmit Buffer Empty** This bit is set by hardware when the Transmit Buffer is copied to the transmit shift register of the Smart Card UART. It is cleared by hardware when SCTBUF register is written.
6	CARDIN	**Card Presence Status** This bit is set by hardware if there is a card presence (debouncing filter has to be done by software). This bit is cleared by hardware if there is no card presence.
5	ICARDOVF	**ICC Overflow on card** This bit is set when the current on card is above the limit specified by bit OVFADJ in DCCKPS register (Table 56 on page 56) It is cleared by hardware.
4	VCARDOK	**Card Voltage Status** This bit is set when the output voltage is within the voltage range specified by VCARD field. It is cleared otherwise.
3	SCWTO	**Smart Card Wait Timeout** This bit is set by hardware when the Smart Card Waiting Time Counter expires. It is cleared by the reload of the counter or by the reset of the SCIB.
2	SCTC	**Smart Card Transmitted Character** This bit is set by hardware when the Smart Card UART has transmitted a character. It shall be cleared by software after this register is read.
1	SCRC	**Smart Card Received Character** This bit is set by hardware when the Smart Card UART has received a character It is cleared by hardware when SCBUF register is read.
0	SCPE	**Smart Card Parity Error** This bit is set at the same time as SCTI or SCRI if a parity error is detected. It shall be cleared by software after this register is read.

Reset Value = 1000 0000b

Table 42. Smart Card UART Interrupt Identification Register (Read Only)
SCIIR (S:AEh, SCRS=0)

7	6	5	4	3	2	1	0
SCTBI	-	ICARDERR	VCARDERR	SCWTI	SCTI	SCRI	SCPI

Bit Number	Bit Mnemonic	Description
7	SCTBI	**SCIB Transmit Buffer Interrupt** This bit is set by hardware when the Transmit Buffer is copied to the transmit shift register of the Smart Card UART. It is cleared by hardware when this register is read.
6	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
5	ICARDERR	**Card Current Status** This bit is set when the output current goes out of the current range. It is cleared by hardware when this register is read.
4	VCARDERR	**Card Voltage Status** This bit is set when the output voltage goes out of the voltage range specified by VCARD field. It is cleared by hardware when this register is read.
3	SCWTI	**Smart Card Wait Timeout Interrupt** This bit is set by hardware when the Smart Card Timer times out. It is cleared by hardware when this register is read.
2	SCTI	**Smart Card Transmit Interrupt** This bit is set by hardware when the Smart Card UART completes a character transmission. It is cleared by hardware when this register is read.
1	SCRI	**Smart Card Receive Interrupt** This bit is set by hardware when the Smart Card UART completes a character reception. It is cleared by hardware when this register is read.
0	SCPI	**Smart Card Parity Error Interrupt** This bit is set at the same time as SCTI or SCRI if a parity error is detected. It is cleared by hardware when this register is read.

Reset Value = 0X00 0000b

Table 43. Smart Card UART Interrupt Enable Register - SCIER (S:AEh, SCRS=1)

7	6	5	4	3	2	1	0
ESCTBI	-	ICARDER	EVCARDER	ESCWTI	ESCTI	ESCRI	ESCPI

Bit Number	Bit Mnemonic	Description
7	ESCTBI	**Smart Card UART Transmit Buffer Empty Interrupt Enable** Clear this bit to disable the Smart Card UART Transmit Buffer Empty interrupt. Set this bit to enable the Smart Card UART Transmit Buffer Empty interrupt.
6	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
5	ICARDER	**Card Current Error Interrupt Enable** Clear this bit to disable the Card Current Error interrupt. Set this bit to enable the Card Current Error interrupt.
4	EVCARDER	**Card Voltage Error Interrupt Enable** Clear this bit to disable the Card Voltage Error interrupt. Set this bit to enable the Card Voltage Error interrupt.
3	ESCWTI	**Smart Card Wait Timeout Interrupt Enable** Clear this bit to disable the Smart Card Wait timeout interrupt. Set this bit to enable the Smart Card Wait timeout interrupt.
2	ESCTI	**Smart Card Transmit Interrupt Enable** Clear this bit to disable the Smart Card UART Transmit interrupt. Set this bit to enable the Smart Card UART Transmit interrupt.
1	ESCRI	**Smart Card Receive Interrupt Enable** Clear this bit to disable the Smart Card UART Receive interrupt. Set this bit to enable the Smart Card UART Receive interrupt.
0	ESCPI	**Smart Card Parity Error Interrupt Enable** Clear this bit to disable the Smart Card UART Parity Error interrupt. Set this bit to enable the Smart Card UART Parity Error interrupt.

Reset Value = 0X00 0000b

Table 44. Smart Card Selection Register - SCSR (S:ABh)

7	6	5	4	3	2	1	0
-	BGTEN	-	CREPSEL	ALTKPS1	ALTKPS0	SCCLK1	SCRS

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
6	BGTEN	**Block Guard Time Enable** Set this bit to select the minimum interval between the leading edge of the start bits of the last received character and the first character sent in the opposite direction. The transfer of GT[8-0] value to the BGT counter is done on the rising edge of the BGTEN. Clear this bit to suppress the minimum time between reception and transmission.
5	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
4	CREPSEL	**Character repetition selection** Clear this bit to select 5 times repetition before parity error indication Set this bit to select 4 times repetition before parity error indication
3-2	ALTKPS1:0	**Alternate Card Clock prescaler factor** 00 ALTKPS = 0: prescaler factor equals 1 01 ALTKPS = 1: prescaler factor equals 2 10 ALTKPS = 2: prescaler factor equals 4 (reset value) 11 ALTKPS = 3: prescaler factor equals 8
1	SCCLK1	**Alternate card clock selection** Set to select the prescaled clock (CCLK1) Clear to select the standard port configuration
0	SCRS	**Smart Card Register Selection** The SCRS bit selects which set of the SCIB registers is accessed.

Reset Value = X000 1000b

Table 45. Smart Card Transmit Buffer Register - SCTBUF (S:AA, write-only, SCRS=0)

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

Bit Number	Bit Mnemonic	Description
-	-	Can store a new byte to be transmitted on the I/O pin when SCTBE is set. Bit ordering on the I/O pin depends on the Convention.

Reset Value = 0000 0000b

Table 46. Smart Card Receive Buffer Register - SCRBUF (S:AA read-only, SCRS=1)

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

Bit Number	Bit Mnemonic	Description
-	-	Provides the byte received from the I/O pin when SCRI is set. Bit ordering on the I/O pin depends on the Convention.

Reset Value = 0000 0000b

Table 47. Smart Card ETU Register 1 - SCETU1 (S:ADh, SCRS=1)

7	6	5	4	3	2	1	0
COMP	-	-	-	-	ETU10	ETU9	ETU8

Bit Number	Bit Mnemonic	Description
7	COMP	**Compensation** Clear this bit when no time compensation is needed (i.e. when the ETU to Card CLK period ratio is close to an integer with an error less than 1/4 of Card CLK period). Set this bit otherwise and reduce the ETU period by 1 Card CLK cycle for even bits.
6-3	-	**Reserved** The value read from these bits is indeterminate. Do not change these bits.
2-0	ETU[10:8]	**ETU MSB** Used together with the ETU LSB in SCETU0 (Table 48)

Reset Value = 0XXX X001b

Table 48. Smart Card ETU Register 0 - SCETU0 (S:ACh, SCRS=1)

7	6	5	4	3	2	1	0
ETU7	ETU6	ETU5	ETU4	ETU3	ETU2	ETU1	ETU0

Bit Number	Bit Mnemonic	Description
7 - 0	ETU[7:0]	**ETU LSB** The Elementary Time Unit is (ETU[10:0] - 0.5*COMP)/f, where f is the Card CLK frequency. According to ISO 7816, ETU[10:0] can be set between 11 and 2047. The default reset value of ETU[10:0] is 372 (F=372, D=1).

Reset Value = 0111 0100b

Table 49. Smart Card Transmit Guard Time Register 0 - SCGT0 (S:B4h, SCRS=1)

7	6	5	4	3	2	1	0
GT7	GT6	GT5	GT4	GT3	GT2	GT1	GT0

Bit Number	Bit Mnemonic	Description
7 - 0	GT[7:0]	**Transmit Guard Time LSB** The minimum time between two consecutive start bits in transmit mode is GT[8:0] * ETU. According to ISO 7816, GT can be set between 11 and 266 (11 to 254+12 ETU).

Reset Value = 0000 1100b

Table 50. Smart Card Transmit Guard Time Register 1 - SCGT1 (S:B5h, SCRS=1)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	GT8

Bit Number	Bit Mnemonic	Description
7 - 1	-	**Reserved** The value read from these bits is indeterminate. Do not change these bits.
0	GT8	**Transmit Guard Time MSB** Used together with the Transmit Guard Time LSB in SCGT0 register (Table 49).

Reset Value = XXXX XXX0b

Table 51. Smart Card Character/Block Wait Time Register 3
SCWT3 (S:C1h, SCRS=0)

7	6	5	4	3	2	1	0
WT31	WT30	WT29	WT28	WT27	WT26	WT25	WT24

Bit Number	Bit Mnemonic	Description
7 - 0	WT[31:24]	**Wait Time Byte3** Used together with WT[23:0] in registers SCWT2,SCWT1, SCWT0 (see Table 52).

Reset Value = 0000 0000b

Table 52. Smart Card Character/Block Wait Time Register 2
SCWT2 (S:B6h, SCRS=0)

7	6	5	4	3	2	1	0
WT23	WT22	WT21	WT20	WT19	WT18	WT17	WT16

Bit Number	Bit Mnemonic	Description
7 - 0	WT[23:16]	**Wait Time Byte2** Used together with WT[31:24] and WT[15:0] in registers SCWT3,SCWT1, SCWT0 (see Table 54).

Reset Value = 0000 0000b

Table 53. Smart Card Character/Block Wait Time Register 1
SCWT1 (S:B5h, SCRS=0)

7	6	5	4	3	2	1	0
WT15	WT14	WT13	WT12	WT11	WT10	WT9	WT8

Bit Number	Bit Mnemonic	Description
7 - 0	WT[15:8]	**Wait Time Byte 1** Used together with WT[31:16] and WT[7:0] in registers SCWT3,SCWT2, SCWT0 (see Table 51).

Reset Value = 0010 0101b

Table 54. Smart Card Character/Block Wait Time Register 0
SCWT0 (S:B4h, SCRS=0)

7	6	5	4	3	2	1	0
WT7	WT6	WT5	WT4	WT3	WT2	WT1	WT0

Bit Number	Bit Mnemonic	Description
7 - 0	WT[7:0]	**Wait Time Byte 0** WT[31:0] is the reload value of the Wait Time counter WTC**.** The WTC is a general-purpose timer. It is using the ETU clock and is controlled by the WTEN bit (see Table 39 on page 45 and Section “Waiting Time (WT) Counter”, page 40). When UART bit of Registers is set, the WTC is automatically reloaded at each start bit of the UART. It is used to check the maximum time between to consecutive start bits.

Reset Value = 1000 0000b

Table 55. Smart Card Clock Reload Register - SCICLK (S:C1h, SCRS=1)

7	6	5	4	3	2	1	0
XTSCS	-	SCICLK5	SCICLK4	SCICLK3	SCICLK2	SCICLK1	SCICLK0

Bit Number	Bit Mnemonic	Description
7	XTSCS	**Smart Card Clock Selection Bit** If XTSCS bit is set, XTAL1 is SCIB clock. If XTSCS bit is cleared and EXT48 bit is set , XTAL1 is SCIB clock. If XTSCS bit is cleared and EXT48 bit is reset, PLL is SCIB clock.
6	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
5 - 0	SCICLK5:0	**SCIB clock reload register** Prescaler 2 reload value is used to defines the card clock frequency. If SCICLK5:0 is smaller than 48 Fck_iso = Fck_pll or Fck_XTAL1/ (2 * (48 - SCICLK5:0)) If SCICLK5:0 is equal to 48 Fck_iso = Fck_XTAL1 or Fck_XTAL1 SCICLK5:0 must be smaller than 49.

Reset Value = 0X10 1111b *(default value for a divider by two)*

DC/DC Converter

The Smart Card voltage (CVCC) is supplied by the integrated DC/DC converter which is controlled by several registers:

- The SCIIR register (Table 42 on page 48) controls the CVCC level by means of bits VCARD[1:0].
- The SCCON register (Table 40 on page 46) enables to switch the DC/DC converter on or off by means of bit CARDVCC.
- The DCCKPS register (Table 56 on page 56) controls the DC/DC clock and current.

The DC/DC converter cannot be switched on while the CPRES pin remains inactive. If CPRES pin becomes inactive while the DC/DC converter is operating an automatic shut down sequence of the DC/DC converter is initiated by the electronics.

It is mandatory to switch off the DC/DC Converter before entering in Power-down mode.

Configuration

The DC/DC Converter can work in two different modes which are selected by bit Mode in DCCKPS register:

- Pump Mode: an external inductance of 10 µH must be connected between pins LI and VCC. VCC can be higher or lower than CVCC.
- Regulator mode: no external inductance is required but VCC must be always higher than CVCC.

The DC/DC clock prescaler which is controlled by bits DCCKPS[3:0], in DCCKPS register must be configured to set the DC/DC clock to a working frequency of 4 MHz which depends on the value of the quartz. There is no need to change the default configuration set by the reset sequence if an 8 MHz quartz is used by the application.

The DC/DC Converter implements a current overflow controller which avoids permanent damage of the DC/DC converter in case of short circuit between CVCC and CVSS. The maximum limit is around 100 mA. It is possible to increase this limit in normal operating mode by 20% by means of bit OVFADJ in DCCKPS register. When the current overflow controller is operating, the ICARDOVF is set by the hardware in SCISR register.

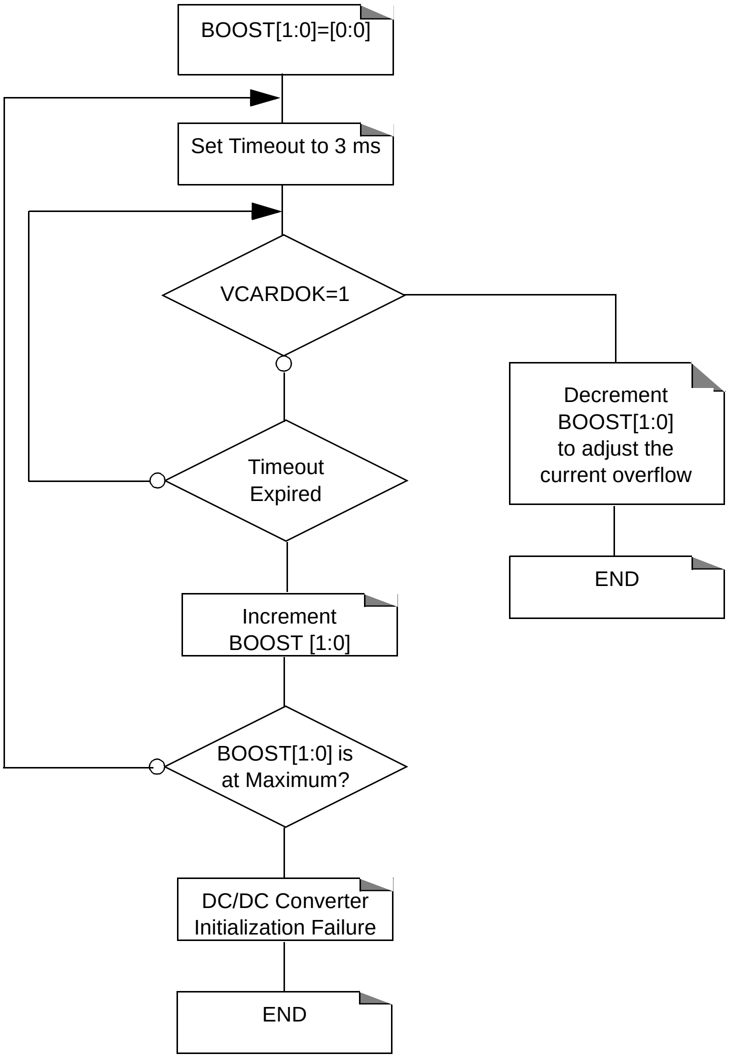
The current drawn from power supply by the DC/DC converter is controlled during the startup phase in order to avoid high transient current mainly in Pump Mode which could cause the power supply voltage to drop dramatically. This control is done by means of bits BOOST[1:0], which increases progressively the startup current level.

Initialization Procedure

The initialization procedure is described in flow chart of Figure 29.

- Select the CVCC level by means of bits VCARD[1:0] in SCIIR register,
- Set bits BOOST[1:0] in DCCKPS register following the current level control wanted.
- Switch the DC/DC on by means of bit CARDVCC in SCCON register,
- Monitor bit VCARDOK in SCISR register in order to know when the DC/DC Converter is ready (CVCC voltage has reached the expected level)

Figure 29. DC/DC Converter Initialization Procedure


BOOST[1:0]=[0:0]
Set Timeout to 3 ms
VCARDOK=1
Timeout Expired
Decrement BOOST[1:0] to adjust the current overflow
END
Increment BOOST [1:0]
BOOST[1:0] is at Maximum?
DC/DC Converter Initialization Failure
END

While VCC remains higher than 3.6V and startup current lower than 30 mA (depending on the load type), the DC/DC converter should be ready without having to increment

BOOST[1:0] bits beyond [0:0] level. If at least one of the two conditions are not met (VCC < 3.6V or startup current > 30 mA), it will be necessary to increment the BOOST[1:0] bits until the DC/DC converter is ready.

Incrementation of BOOST[1:0] bits increases at the same time the current overflow level in the same proportion as the startup current. So once the DC/DC converter is ready it advised to decrement the BOOST[1:0] bits to restore the overflow current to its normal or desired value.

Table 56. DC/DC Converter Control Register - DCCKPS (S:BFh)

7	6	5	4	3	2	1	0
MODE	OVFADJ	BOOST1	BOOST0	DCCKPS3	DCCKPS2	DCCKPS1	DCCKPS0

Bit Number	Bit Mnemonic	Description	
7	MODE	Regulation mode 0 DC/DC converter (External Inductance required) 1 Voltage Regulator (No External inductance required but VCC > CVCC+0.3V)	
6	OVFADJ	Current Overflow Adjustment on Smart Card terminal 0 normal: 100 mA average 1 normal + 20%	
5 - 4	BOOST[1:0]	**Maximum Startup Current drawn from power supply** 00 Normal: 30 mA average 01 Normal + 10% 10 Normal + 30% 11 Normal + 60%	**Current Overflow Level on Smart Card terminal** 00 Normal = OVFADJ 01 Normal + 10% 10 Normal + 30% 11 Normal + 60%
3 - 0	DCCKPS[3:0]	DC/DC Clock Prescaler Value 0000 Division factor: 2 (reset value) 0001 Division factor: 3 0010 Division factor: 4 0011 Division factor: 5 0100 Division factor: 6 0101 Division factor: 8 0110 Division factor: 10 0111 Division factor: 12 1000 Division factor: 24 Other values reserved	

Reset Value = 0000 0000b

USB Controller

The AT8xC5122 implements a USB device controller supporting Full Speed data transfer. In addition to the default control endpoint 0, it provides 6 other endpoints, which can be configured in Control, Bulk, Interrupt or Isochronous modes:

- Endpoint 0: 32-byte FIFO, default control endpoint
- Endpoint 1,2,3: 8-byte FIFO
- Endpoint 4,5: 64-byte FIFO
- Endpoint 6: 2 x 64-byte Ping-pong FIFO

This allows the firmware to be developed conforming to most USB device classes, for example:

- USB Mass Storage Class Control/Bulk/Interrupt (CBI) Transport, Revision 1.0 - December 14, 1998.
- USB Mass Storage Class Bulk-Only Transport, Revision 1.0 - September 31, 1999.
- USB Human Interface Device Class, Version 1.1 - April 7, 1999.
- USB Device Firmware Upgrade Class, Revision 1.0 - May 13, 1999.

USB Mass Storage Classes

USB Mass Storage Class CBI Transport

Within the CBI framework, the Control endpoint is used to transport command blocks as well as to transport standard USB requests. One Bulk-Out endpoint is used to transport data from the host to the device. One Bulk-In endpoint is used to transport data from the device to the host. And one interrupt endpoint may also be used to signal command completion (protocol 0); it is optional and may not be used (protocol 1).

The following configuration adheres to these requirements:

- Endpoint 0: 8 bytes, Control In-Out
- Endpoint 4: 64 bytes, Bulk-Out
- Endpoint 5: 64 bytes, Bulk-In
- Endpoint 1: 8 bytes, Interrupt In

USB Mass Storage Class Bulk-Only Transport

Within the Bulk-Only framework, the Control endpoint is only used to transport class-specific and standard USB requests for device set-up and configuration. One Bulk-Out endpoint is used to transport commands and data from the host to the device. One Bulk-In endpoint is used to transport status and data from the device to the host. No interrupt endpoint is needed.

The following configuration adheres to these requirements:

- Endpoint 0: 8 bytes, Control In-Out
- Endpoint 4: 64 bytes, Bulk-Out
- Endpoint 5: 64 bytes, Bulk-In

USB Device Firmware Upgrade (DFU)

The USB Device Firmware Update (DFU) protocol can be used to upgrade the on-chip program memory of the AT8xC5122. This allows the implementation of product enhancements and patches to devices that are already in the field. Two different configurations and description sets are used to support DFU functions. The Run-Time configuration co-exists with the usual functions of the device, which may be USB Mass Storage for the AT8xC5122. It is used to initiate DFU from the normal operating mode. The DFU configuration is used to perform the firmware update after device re-configuration and USB reset. It excludes any other function. Only the default control pipe (endpoint 0) is used to support DFU services in both configurations.

The only possible value for the wMaxPacketSize in the DFU configuration is 32 bytes, which is the size of the FIFO implemented for endpoint 0.

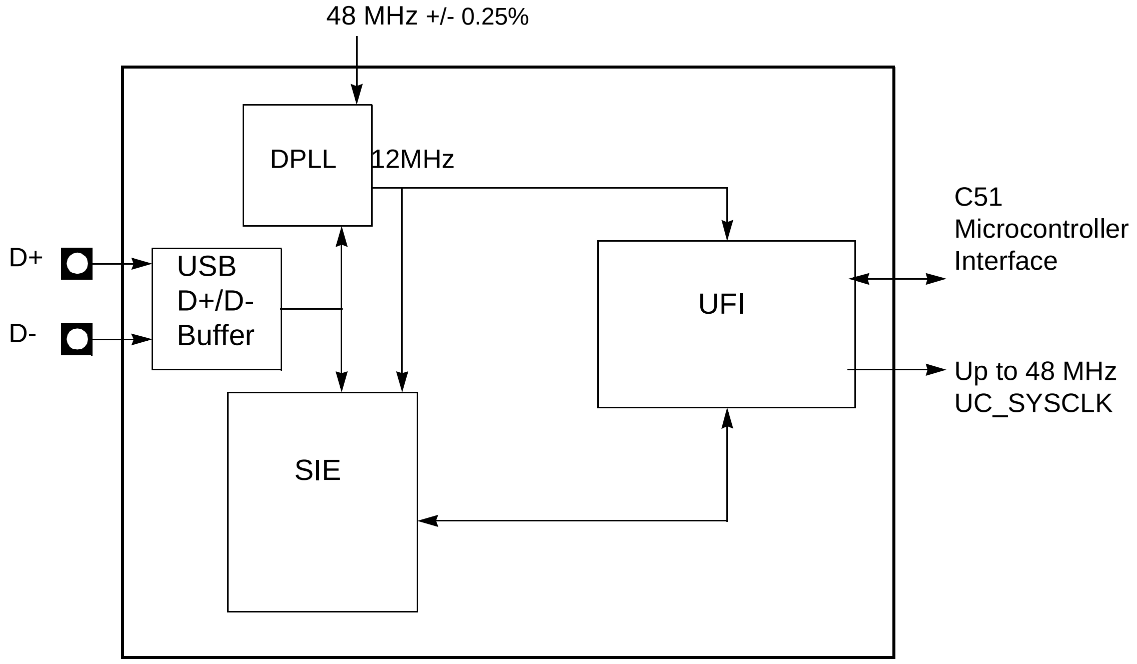
Description

The USB device controller provides the hardware that the AT8xC5122 and the AT8xC5123 need to interface a USB link to a data flow stored in a double port memory (DPRAM).

The USB controller requires a 48 MHz reference clock, which is the output of the AT8xC5122/23 PLL (see Section "PLL", page 32) divided by a clock prescaler. This clock is used to generate a 12 MHz full speed bit clock from the received USB differential data and to transmit data according to full speed USB device tolerance. Clock recovery is done by a Digital Phase Locked Loop (DPLL) block, which is compliant with the jitter specification of the USB bus.

The Interface Engine (SIE) block performs NRZI encoding and decoding, bit stuffing, CRC generation and checking, and the serial-parallel data conversion. The Universal Function Interface (UFI) performs the interface between the data flow and the Dual Port Ram

Figure 30. USB Device Controller Block Diagram

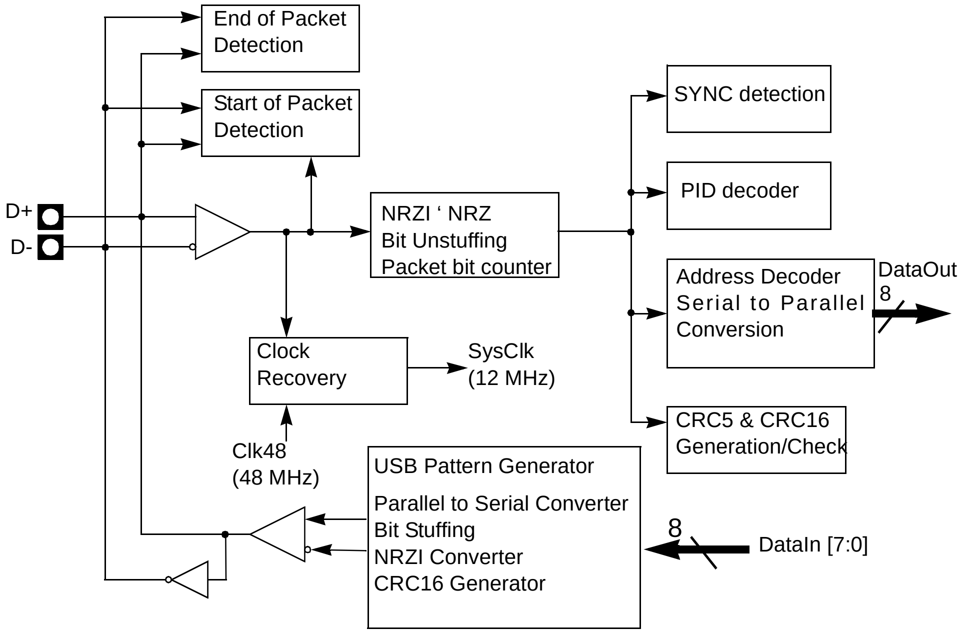


Serial Interface Engine (SIE)

The SIE performs the following functions:

- NRZI data encoding and decoding.
- Bit stuffing and unstuffing.
- CRC generation and checking.
- Handshakes.
- TOKEN type identifying.
- Address checking.
- Clock generation (via DPLL).

Figure 31. SIE Block Diagram



Function Interface Unit (UFI)

The Function Interface Unit provides the interface between the AT8xC5122 (or AT8xC5123) and the SIE. It manages transactions at the packet level with minimal intervention from the device firmware, which reads and writes the endpoint FIFOs.

Figure 32. UFI Block Diagram

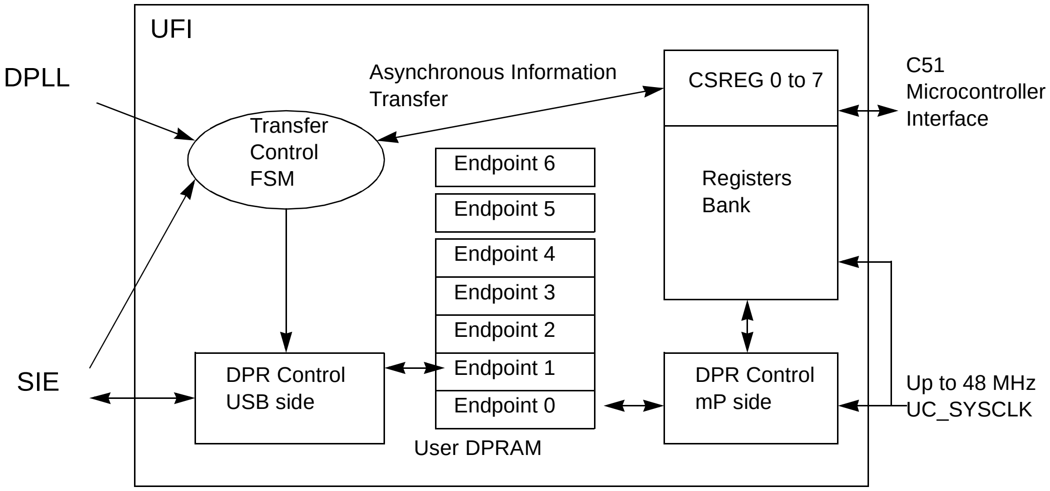
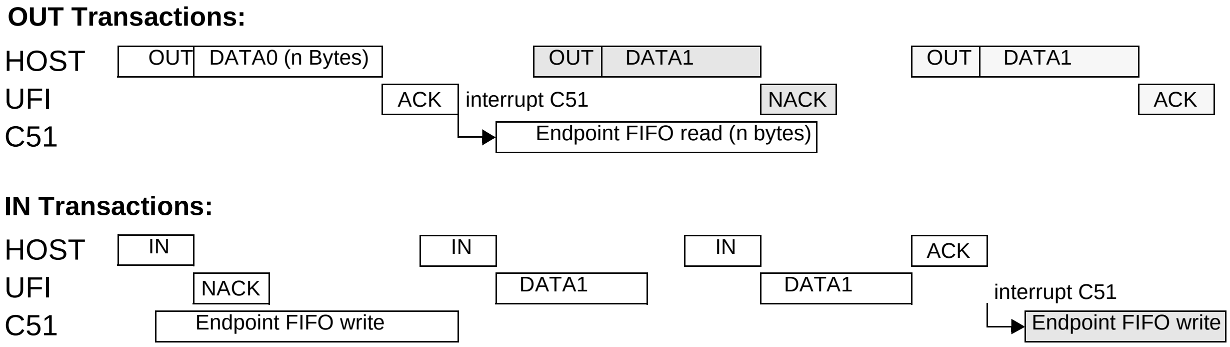


Figure 33. Minimum Intervention from the USB Device Firmware



Configuration

General Configuration

- USB controller enable

Before any USB transaction, the 48 MHz required by the USB controller must be correctly generated (Section "Clock Controller", page 28).

The USB controller should be then enabled by setting the USBE bit in the USBCON register.

- Set address

After a Reset or a USB reset, the software has to set the FEN (Function Enable) bit in the USBADDR register. This action will allow the USB controller to answer to the requests sent at the address 0.

When a SET_ADDRESS request has been received, the USB controller must only answer to the address defined by the request. The new address should be stored in the USBADDR register. The FEN bit and the FADDEN bit in the USBCON register should be set to allow the USB controller to answer only to requests sent at the new address.

- Set configuration

The CONFG bit in the USBCON register should be set after a SET_CONFIGURATION request with a non-zero value. Otherwise, this bit should be cleared.

Endpoint Configuration

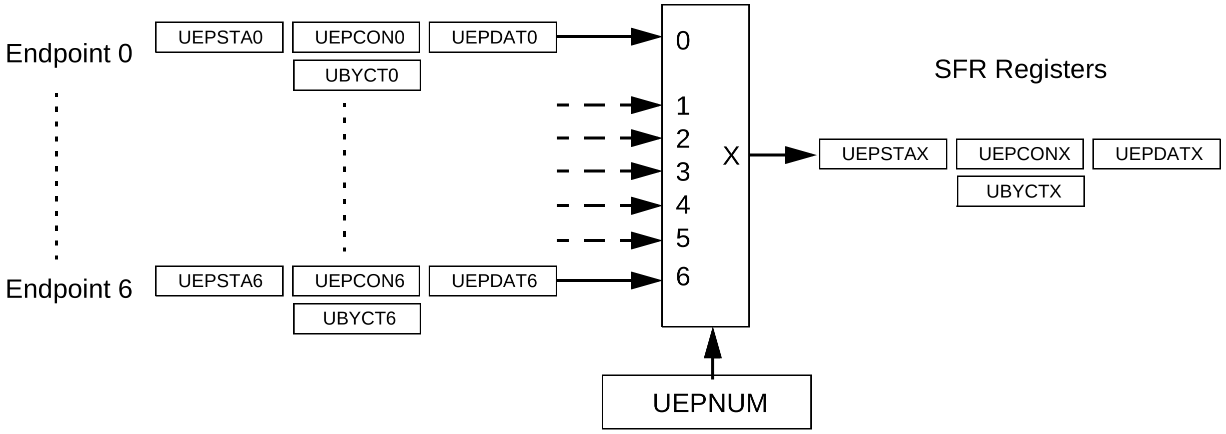
- Selection of an Endpoint

The endpoint register access is performed using the UEPNUM register. The following registers

correspond to the endpoint whose number is stored in the UEPNUM register. To select an Endpoint, the firmware has to write the endpoint number in the UEPNUM register.

  - UEPSTAX,
  - UEPCONX,
  - UEPDATX,
  - UBYCTX,

Figure 34. Endpoint Selection



- Endpoint enable

Before using an endpoint, this one should be enabled by setting the EPEN bit in the UEPCONX register.

An endpoint which is not enabled won't answer to any USB request. The Default Control Endpoint (Endpoint 0) should always be enabled in order to answer to USB standard requests.

- Endpoint type configuration

All Standard Endpoints can be configured in Control, Bulk, Interrupt or Isochronous mode. The Ping-pong Endpoints can be configured in Bulk, Interrupt or Isochronous mode. The configuration of an endpoint is performed by setting the field EPTYPE with the following values:

– Control: EPTYPE = 00b
– Isochronous: EPTYPE = 01b
– Bulk: EPTYPE = 10b
– Interrupt: EPTYPE = 11b

The Endpoint 0 is the Default Control Endpoint and should always be configured in Control type.

- Endpoint direction configuration

For Bulk, Interrupt and Isochronous endpoints, the direction is defined with the EPDIR bit of the UEPCONX register with the following values:

– IN:EPDIR = 1b
– OUT:EPDIR = 0b

For Control endpoints, the EPDIR bit has no effect.

- Summary of Endpoint Configuration:

Make sure to select the correct endpoint number in the UEPNUM register before accessing to endpoint specific registers.

Table 57. Summary of Endpoint Configuration

Endpoint configuration	EPEN	EPDIR	EPTYPE	UEPCONX
Disabled	0b	Xb	XXb	0XXX XXXb
Control	1b	Xb	00b	80h
Bulk-In	1b	1b	10b	86h
Bulk-Out	1b	0b	10b	82h
Interrupt-In	1b	1b	11b	87h
Interrupt-Out	1b	0b	11b	83h
Isochronous-In	1b	1b	01b	85h
Isochronous-Out	1b	0b	01b	81h

- Endpoint FIFO reset

Before using an endpoint, its FIFO should be reset. This action resets the FIFO pointer to its original value, resets the byte counter of the endpoint (UBYCTX register), and resets the data toggle bit (DTGL bit in UEPCONX).

The reset of an endpoint FIFO is performed by setting to 1 and resetting to 0 the corresponding bit in the UEPRST register.

For example, in order to reset the Endpoint number 2 FIFO, write 0000 0100b then 0000 0000b in the UEPRST register.

Read/Write Data FIFO

Read Data FIFO

The read access for each OUT endpoint is performed using the UEPDATX register.

After a new valid packet has been received on an Endpoint, the data are stored into the FIFO and the byte counter of the endpoint is updated (UBYCTX register). The firmware has to store the endpoint byte counter before any access to the endpoint FIFO. The byte counter is not updated when reading the FIFO.

To read data from an endpoint, select the correct endpoint number in UEPNUM and read the UEPDATX register. This action automatically decreases the corresponding address vector, and the next data is then available in the UEPDATX register.

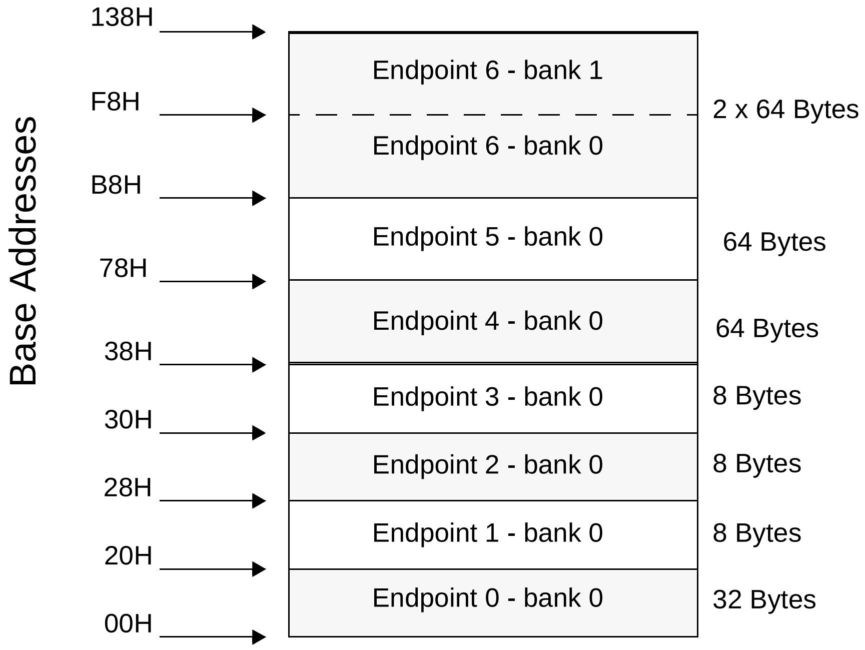
Write Data FIFO

The write access for each IN endpoint is performed using the UEPDATX register.

To write a byte into an IN endpoint FIFO, select the correct endpoint number in UEPNUM and write into the UEPDATX register. The corresponding address vector is automatically increased, and another write can be carried out.

Warning 1: The byte counter is not updated.
Warning 2: Do not write more bytes than supported by the corresponding endpoint.

Figure 35. Endpoint FIFO Configuration

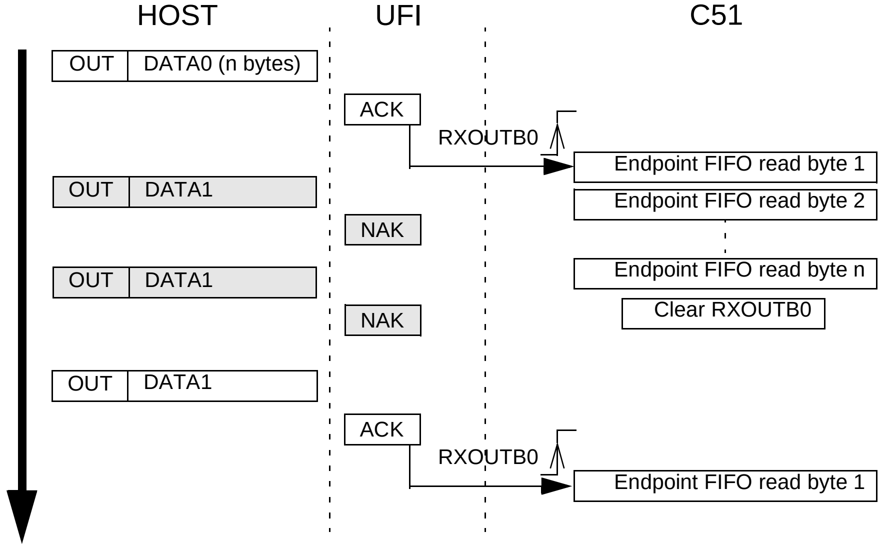


Bulk / Interrupt Transactions

Bulk and Interrupt transactions are managed in the same way.

Bulk/Interrupt OUT Transactions in Standard Mode

Figure 36. Bulk/Interrupt OUT transactions in Standard Mode


HOST
UFI
C51
OUT
DATA0 (n bytes)
ACK
RXOUTB0
Endpoint FIFO read byte 1
Endpoint FIFO read byte 2
OUT
DATA1
NAK
Endpoint FIFO read byte n
OUT
DATA1
Clear RXOUTB0
NAK
OUT
DATA1
ACK
RXOUTB0
Endpoint FIFO read byte 1

An endpoint should be first enabled and configured before being able to receive Bulk or Interrupt packets.

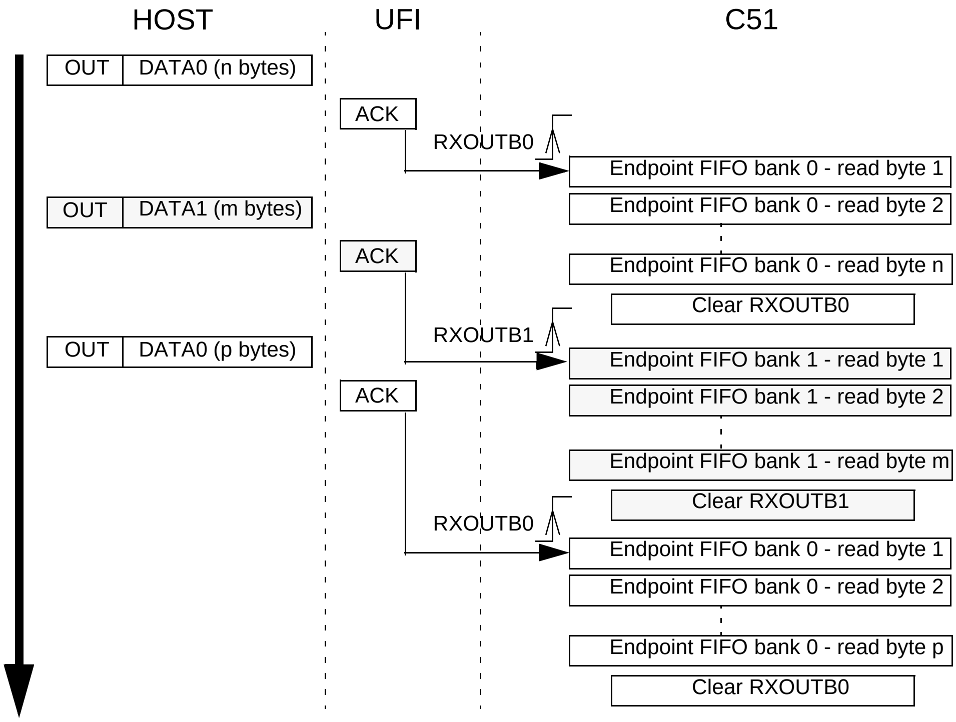
When a valid OUT packet is received on an endpoint, the RXOUTB0 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware has to select the corresponding endpoint, store the number of data bytes by reading the UBYCTX register. If the received packet is a ZLP (Zero Length Packet), the UBYCTX register value is equal to 0 and no data has to be read.

When all the endpoint FIFO bytes have been read, the firmware should clear the RXOUTB0 bit to allow the USB controller to accept the next OUT packet on this endpoint. Until the RXOUTB0 bit has been cleared by the firmware, the USB controller will answer a NAK handshake for each OUT requests.

If the Host sends more bytes than supported by the endpoint FIFO, the overflow data won’t be stored, but the USB controller will consider that the packet is valid if the CRC is correct and the endpoint byte counter contains the number of bytes sent by the Host.

Bulk/Interrupt OUT Transactions in Ping-Pong Mode (Endpoints 6)

Figure 37. Bulk / Interrupt OUT transactions in Ping-Pong mode


HOST
UFI
C51
OUT
DATA0 (n bytes)
ACK
RXOUTB0
Endpoint FIFO bank 0 - read byte 1
Endpoint FIFO bank 0 - read byte 2
OUT
DATA1 (m bytes)
ACK
Endpoint FIFO bank 0 - read byte n
Clear RXOUTB0
RXOUTB1
OUT
DATA0 (p bytes)
Endpoint FIFO bank 1 - read byte 1
ACK
Endpoint FIFO bank 1 - read byte 2
Endpoint FIFO bank 1 - read byte m
Clear RXOUTB1
RXOUTB0
Endpoint FIFO bank 0 - read byte 1
Endpoint FIFO bank 0 - read byte 2
Endpoint FIFO bank 0 - read byte p
Clear RXOUTB0

An endpoint should be first enabled and configured before being able to receive Bulk or Interrupt packets.

When a valid OUT packet is received on the endpoint bank 0, the RXOUTB0 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware has to select the corresponding endpoint, store the number of data bytes by reading the UBYCTX register. If the received packet is a ZLP (Zero Length Packet), the UBYCTX register value is equal to 0 and no data has to be read.

When all the endpoint FIFO bytes have been read, the firmware should clear the RXOUB0 bit to allow the USB controller to accept the next OUT packet on the endpoint bank 0. This action switches the endpoint bank 0 and 1. Until the RXOUTB0 bit has been cleared by the firmware, the USB controller will answer a NAK handshake for each OUT requests on the bank 0 endpoint FIFO.

When a new valid OUT packet is received on the endpoint bank 1, the RXOUTB1 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware empties the bank 1 endpoint FIFO before clearing the RXOUTB1 bit. Until the RXOUTB1 bit has been cleared by the firmware, the USB controller will answer a NAK handshake for each OUT requests on the bank 1 endpoint FIFO.

The RXOUTB0 and RXOUTB1 bits are alternatively set by the USB controller at each new valid packet receipt.

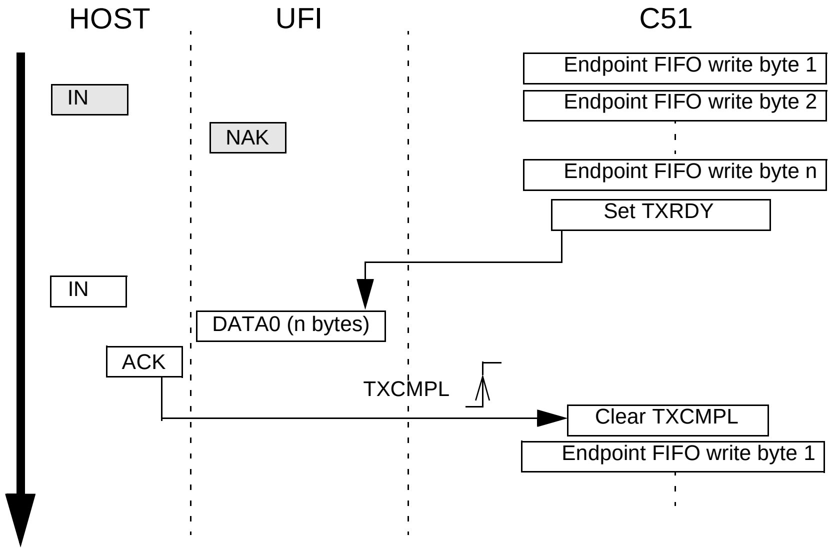
The firmware has to clear one of these two bits after having read all the data FIFO to allow a new valid packet to be stored in the corresponding bank.

A NAK handshake is sent by the USB controller only if the banks 0 and 1 has not been released by the firmware.

If the Host sends more bytes than supported by the endpoint FIFO, the overflow data won't be stored, but the USB controller will consider that the packet is valid if the CRC is correct.

Bulk/Interrupt IN Transactions In Standard Mode

Figure 38. Bulk/Interrupt IN Transactions in Standard Mode


HOST
UFI
C51
Endpoint FIFO write byte 1
IN
Endpoint FIFO write byte 2
NAK
Endpoint FIFO write byte n
Set TXRDY
IN
DATA0 (n bytes)
ACK
TXCMPL
Clear TXCMPL
Endpoint FIFO write byte 1

An endpoint should be first enabled and configured before being able to send Bulk or Interrupt packets.

The firmware should fill the FIFO with the data to be sent and set the TXRDY bit in the UEPSTAX register to allow the USB controller to send the data stored in FIFO at the next IN request concerning this endpoint. To send a Zero Length Packet, the firmware should set the TXRDY bit without writing any data into the endpoint FIFO.

Until the TXRDY bit has been set by the firmware, the USB controller will answer a NAK handshake for each IN requests.
To cancel the sending of this packet, the firmware has to reset the TXRDY bit. The packet stored in the endpoint FIFO is then cleared and a new packet can be written and sent.

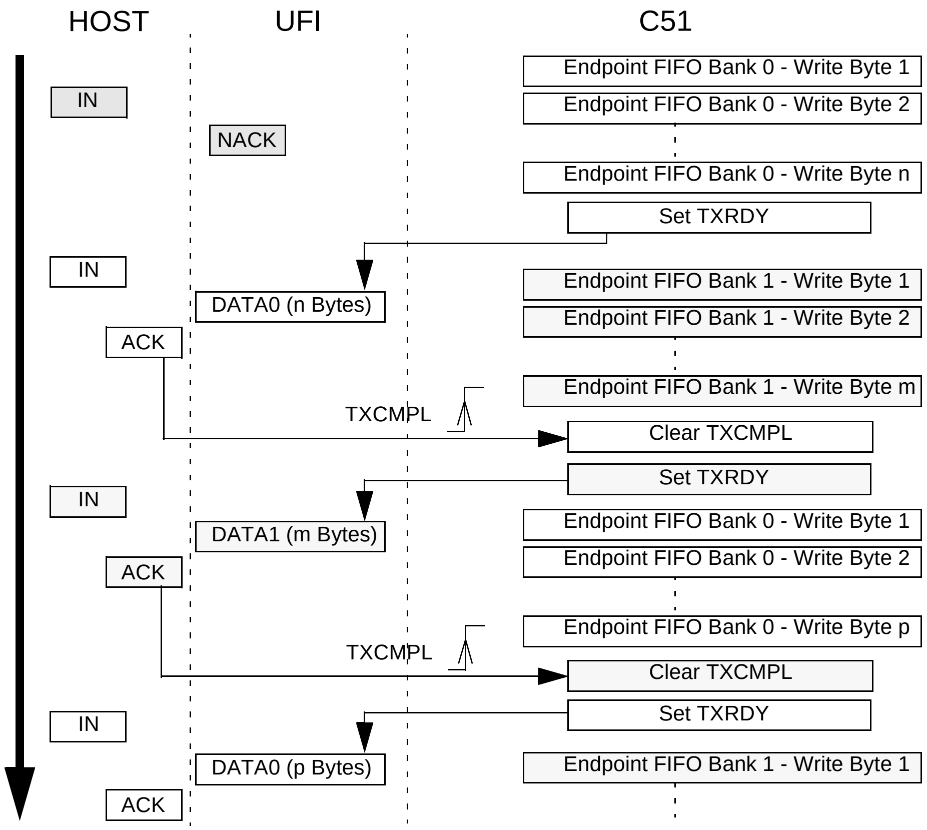
When the IN packet has been sent and acknowledged by the Host, the TXCMPL bit in the UEPSTAX register is set by the USB controller. This triggers a USB interrupt if enabled. The firmware should clear the TXCMPL bit before filling the endpoint FIFO with new data.

The firmware should never write more bytes than supported by the endpoint FIFO.

All USB retry mechanisms are automatically managed by the USB controller.

Bulk/Interrupt IN Transactions in Ping-Pong Mode

Figure 39. Bulk / Interrupt IN transactions in Ping-Pong mode


HOST
UFI
C51
Endpoint FIFO Bank 0 - Write Byte 1
IN
Endpoint FIFO Bank 0 - Write Byte 2
NACK
Endpoint FIFO Bank 0 - Write Byte n
Set TXRDY
IN
Endpoint FIFO Bank 1 - Write Byte 1
DATA0 (n Bytes)
Endpoint FIFO Bank 1 - Write Byte 2
ACK
Endpoint FIFO Bank 1 - Write Byte m
TXCMPL
Clear TXCMPL
Set TXRDY
IN
Endpoint FIFO Bank 0 - Write Byte 1
DATA1 (m Bytes)
Endpoint FIFO Bank 0 - Write Byte 2
ACK
Endpoint FIFO Bank 0 - Write Byte p
TXCMPL
Clear TXCMPL
Set TXRDY
IN
DATA0 (p Bytes)
Endpoint FIFO Bank 1 - Write Byte 1
ACK

An endpoint will be first enabled and configured before being able to send Bulk or Interrupt packets.

The firmware will fill the FIFO bank 0 with the data to be sent and set the TXRDY bit in the UEPSTAX register to allow the USB controller to send the data stored in FIFO at the next IN request concerning the endpoint. The FIFO banks are automatically switched, and the firmware can immediately write into the endpoint FIFO bank 1.

When the IN packet concerning the bank 0 has been sent and acknowledged by the Host, the TXCMPL bit is set by the USB controller. This triggers a USB interrupt if enabled. The firmware will clear the TXCMPL bit before filling the endpoint FIFO bank 0 with new data. The FIFO banks are then automatically switched.

When the IN packet concerning the bank 1 has been sent and acknowledged by the Host, the TXCMPL bit is set by the USB controller. This triggers a USB interrupt if enabled. The firmware will clear the TXCMPL bit before filling the endpoint FIFO bank 1 with new data.

The bank switch is performed by the USB controller each time the TXRDY bit is set by the firmware. Until the TXRDY bit has been set by the firmware for an endpoint bank, the USB controller will answer a NAK handshake for each IN requests concerning this bank.

Note that in the example above, the firmware clears the Transmit Complete bit (TXCMPL) before setting the Transmit Ready bit (TXRDY). This is done in order to avoid the firmware to clear at the same time the TXCMPL bit for bank 0 and the bank 1.

The firmware will never write more bytes than supported by the endpoint FIFO.

Control Transactions

Setup Stage

The DIR bit in the UEPSTAX register should be at 0.
Receiving Setup packets is the same as receiving Bulk Out packets, except that the Rxsetup bit in the UEPSTAX register is set by the USB controller instead of the RXOUTB0 bit to indicate that an Out packet with a Setup PID has been received on the Control endpoint. When the RXSETUP bit has been set, all the other bits of the UEPSTAX register are cleared and an interrupt is triggered if enabled.
The firmware has to read the Setup request stored in the Control endpoint FIFO before clearing the RXSETUP bit to free the endpoint FIFO for the next transaction.

Data Stage: Control Endpoint Direction

The data stage management is similar to Bulk management.

A Control endpoint is managed by the USB controller as a full-duplex endpoint: IN and OUT. All other endpoint types are managed as half-duplex endpoint: IN or OUT. The firmware has to specify the control endpoint direction for the data stage using the DIR bit in the UEPSTAX register.

- If the data stage consists of INs,
  the firmware has to set the DIR bit in the UEPSTAX register before writing into the FIFO and sending the data by setting to 1 the TXRDY bit in the UEPSTAX register. The IN transaction is complete when the TXCMPL has been set by the hardware. The firmware should clear the TXCMPL bit before any other transaction.
- If the data stage consists of OUTs,
  the firmware has to leave the DIR bit at 0. The RXOUTB0 bit is set by hardware when a new valid packet has been received on the endpoint. The firmware must read the data stored into the FIFO and then clear the RXOUTB0 bit to reset the FIFO and to allow the next transaction.

The bit DIR is used to send the correct data toggle in the data stage.

To send a STALL handshake, see “STALL Handshake” on page 72.

Status Stage

The DIR bit in the UEPSTAX register should be reset at 0 for IN and OUT status stage.

The status stage management is similar to Bulk management.

- For a Control Write transaction or a No-Data Control transaction, the status stage consists of a IN Zero Length Packet (see “Bulk/Interrupt IN Transactions In Standard Mode” on page 67). To send a STALL handshake, see “STALL Handshake” on page 72.
- For a Control Read transaction, the status stage consists of a OUT Zero Length Packet (see “Bulk/Interrupt OUT Transactions in Standard Mode” on page 65).

Isochronous Transactions

Isochronous OUT Transactions in Standard Mode

An endpoint should be first enabled and configured before being able to receive Isochronous packets.

When an OUT packet is received on an endpoint, the RXOUTB0 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware has to select the corresponding endpoint, store the number of data bytes by reading the UBYCTX register. If the received packet is a ZLP (Zero Length Packet), the UBYCTX register value is equal to 0 and no data has to be read.
The STLCRC bit in the UEPSTAX register is set by the USB controller if the packet stored in FIFO has a corrupted CRC. This bit is updated after each new packet receipt.

When all the endpoint FIFO bytes have been read, the firmware should clear the RXOUTB0 bit to allow the USB controller to store the next OUT packet data into the endpoint FIFO. Until the RXOUTB0 bit has been cleared by the firmware, the data sent by the Host at each OUT transaction will be lost.
If the RXOUTB0 bit is cleared while the Host is sending data, the USB controller will store only the remaining bytes into the FIFO.

If the Host sends more bytes than supported by the endpoint FIFO, the overflow data won’t be stored, but the USB controller will consider that the packet is valid if the CRC is correct.

Isochronous OUT Transactions in Ping-pong Mode

An endpoint should be first enabled and configured before being able to receive Isochronous packets.

When a OUT packet is received on the endpoint bank 0, the RXOUTB0 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware has to select the corresponding endpoint, store the number of data bytes by reading the UBYCTX register. If the received packet is a ZLP (Zero Length Packet), the UBYCTX register value is equal to 0 and no data has to be read.
The STLCRC bit in the UEPSTAX register is set by the USB controller if the packet stored in FIFO has a corrupted CRC. This bit is updated after each new packet receipt.

When all the endpoint FIFO bytes have been read, the firmware should clear the RXOUB0 bit to allow the USB controller to store the next OUT packet data into the endpoint FIFO bank 0. This action switches the endpoint bank 0 and 1. Until the RXOUTB0 bit has been cleared by the firmware, the data sent by the Host on the bank 0 endpoint FIFO will be lost.
If the RXOUTB0 bit is cleared while the Host is sending data on the endpoint bank 0, the USB controller will store only the remaining bytes into the FIFO.

When a new OUT packet is received on the endpoint bank 1, the RXOUTB1 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware empties the bank 1 endpoint FIFO before clearing the RXOUTB1 bit. Until the RXOUTB1 bit has been cleared by the firmware, the data sent by the Host on the bank 1 endpoint FIFO will be lost.

The RXOUTB0 and RXOUTB1 bits are alternatively set by the USB controller at each new packet receipt.
The firmware has to clear one of these two bits after having read all the data FIFO to allow a new packet to be stored in the corresponding bank.

If the Host sends more bytes than supported by the endpoint FIFO, the overflow data won't be stored, but the USB controller will consider that the packet is valid if the CRC is correct.

Isochronous IN Transactions in Standard Mode

An endpoint should be first enabled and configured before being able to send Isochronous packets.

The firmware should fill the FIFO with the data to be sent and set the TXRDY bit in the UEPSTAX register to allow the USB controller to send the data stored in FIFO at the next IN request concerning this endpoint.

If the TXRDY bit is not set when the IN request occurs, nothing will be sent by the USB controller.

When the IN packet has been sent, the TXCMPL bit in the UEPSTAX register is set by the USB controller. This triggers a USB interrupt if enabled. The firmware should clear the TXCMPL bit before filling the endpoint FIFO with new data. The firmware should never write more bytes than supported by the endpoint FIFO.

Isochronous IN Transactions in Ping-Pong Mode

An endpoint should be first enabled and configured before being able to send Isochronous packets.

The firmware should fill the FIFO bank 0 with the data to be sent and set the TXRDY bit in the UEPSTAX register to allow the USB controller to send the data stored in FIFO at the next IN request concerning the endpoint. The FIFO banks are automatically switched, and the firmware can immediately write into the endpoint FIFO bank 1.
If the TXRDY bit is not set when the IN request occurs, nothing will be sent by the USB controller.

When the IN packet concerning the bank 0 has been sent, the TXCMPL bit is set by the USB controller. This triggers a USB interrupt if enabled. The firmware should clear the TXCMPL bit before filling the endpoint FIFO bank 0 with new data. The FIFO banks are then automatically switched.

When the IN packet concerning the bank 1 has been sent, the TXCMPL bit is set by the USB controller. This triggers a USB interrupt if enabled. The firmware should clear the TXCMPL bit before filling the endpoint FIFO bank 1 with new data.

The bank switch is performed by the USB controller each time the TXRDY bit is set by the firmware. Until the TXRDY bit has been set by the firmware for an endpoint bank, the USB controller won't send anything at each IN requests concerning this bank.

The firmware should never write more bytes than supported by the endpoint FIFO.

Miscellaneous

USB Reset

The EORINT bit in the USBINT register is set by hardware when a End of Reset has been detected on the USB bus. This triggers a USB interrupt if enabled. The USB controller is still enabled, but all the USB registers are reset by hardware. The firmware should clear the EORINT bit to allow the next USB reset detection.

STALL Handshake

This function is only available for Control, Bulk, and Interrupt endpoints.

The firmware has to set the STALLRQ bit in the UEPSTAX register to send a STALL handshake at the next request of the Host on the endpoint selected with the UEPNUM register. The RXSETUP, TXRDY, TXCMPL, RXOUTB0 and RXOUTB1 bits must be first reset to 0. The bit STLCRC is set at 1 by the USB controller when a STALL has been sent. This triggers an interrupt if enabled.

The firmware should clear the STALLRQ and STLCRC bits after each STALL sent.
The STALLRQ bit is cleared automatically by hardware when a valid SETUP PID is received on a CONTROL type endpoint.

Start of Frame Detection

The SOFINT bit in the USBINT register is set when the USB controller detects a Start Of Frame PID. This triggers an interrupt if enabled. The firmware should clear the SOFINT bit to allow the next Start of Frame detection.

Frame Number

When receiving a Start of Frame, the frame number is automatically stored in the UFNUML and UFNUMH registers. The CRCOK and CRCERR bits indicate if the CRC of the last Start Of Frame is valid (CRCOK set at 1) or corrupt (CRCERR set at 1). The UFNUML and UFNUMH registers are automatically updated when receiving a new Start of Frame.

Data Toggle Bit

The Data Toggle bit is set by hardware when a DATA 0 packet is received and accepted by the USB controller and cleared by hardware when a DATA 1 packet is received and accepted by the USB controller. This bit is reset when the firmware resets the endpoint FIFO using the UEPRST register.

For Control endpoints, each SETUP transaction starts with a DATA 0 and data toggling is then used as for Bulk endpoints until the end of the Data stage (for a control write transfer). The Status stage completes the data transfer with a DATA 1 (for a control read transfer).

For Isochronous endpoints, the device firmware should ignore the data-toggle.

NAK handshakes

When a NAK handshake is sent by the USB controller to a IN or OUT request from the Host, the NAKIN or NAKOUT bit is set by hardware. This information can be used to determine the direction of the communication during a Control transfer.
These bits are cleared by software.

Suspend/Resume Management

Suspend

The Suspend state can be detected by the USB controller if all the clocks are enabled and if the USB controller is enabled. The bit SPINT is set by hardware when an idle state is detected for more than 3 ms. This triggers a USB interrupt if enabled.

In order to reduce current consumption, the firmware can put the USB PAD in idle mode, stop the clocks and put the C51 in Idle or Power-down mode. The Resume detection is still active.
The USB PAD is put in idle mode when the firmware clear the SPINT bit. In order to avoid a new suspend detection 3ms later, the firmware has to disable the USB clock input using the SUSPCLK bit in the USBCON Register. The USB PAD automatically exits of idle mode when a wake-up event is detected.

The stop of the 48 MHz clock from the PLL should be done in the following order:

1. Disable of the 48 MHz clock input of the USB controller by setting to 1 the SUSPCLK bit in the USBCON register.
2. Disable the PLL by clearing the PLLEN bit in the PLLCON register.

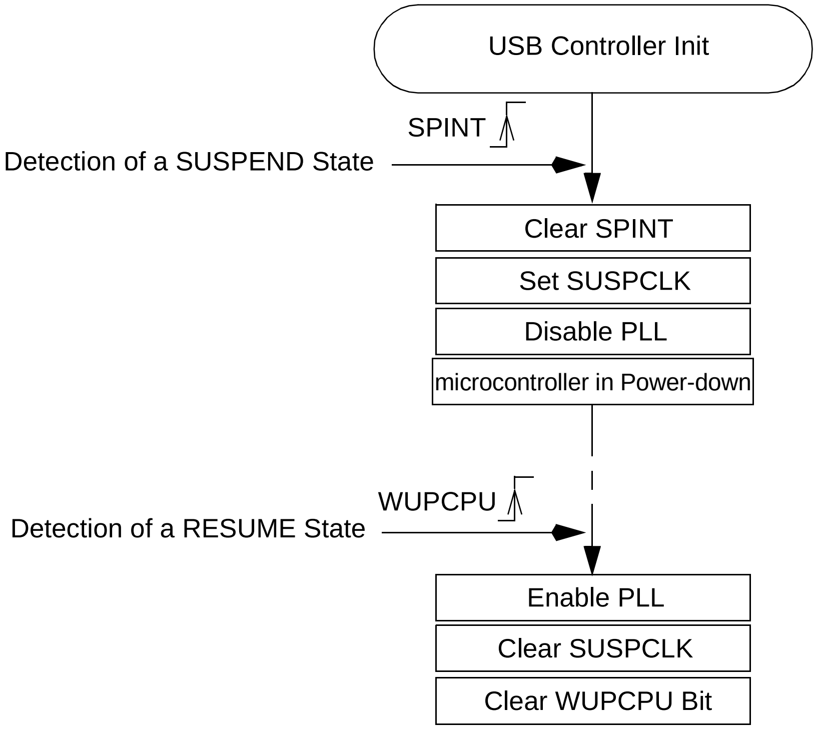
Resume

When the USB controller is in Suspend state, the Resume detection is active even if all the clocks are disabled and if the C51 is in Idle or Power-down mode. The WUPCPU bit is set by hardware when a non-idle state occurs on the USB bus. This triggers an interrupt if enabled. This interrupt wakes up the CPU from its Idle or Power-down state and the interrupt function is then executed. The firmware will first enable the 48 MHz generation and then reset to 0 the SUSPCLK bit in the USBCON register if needed.

The firmware has to clear the SPINT bit in the USBINT register before any other USB operation in order to wake up the USB controller from its Suspend mode.

The USB controller is then re-activated.

Figure 40. Example of a Suspend/Resume Management



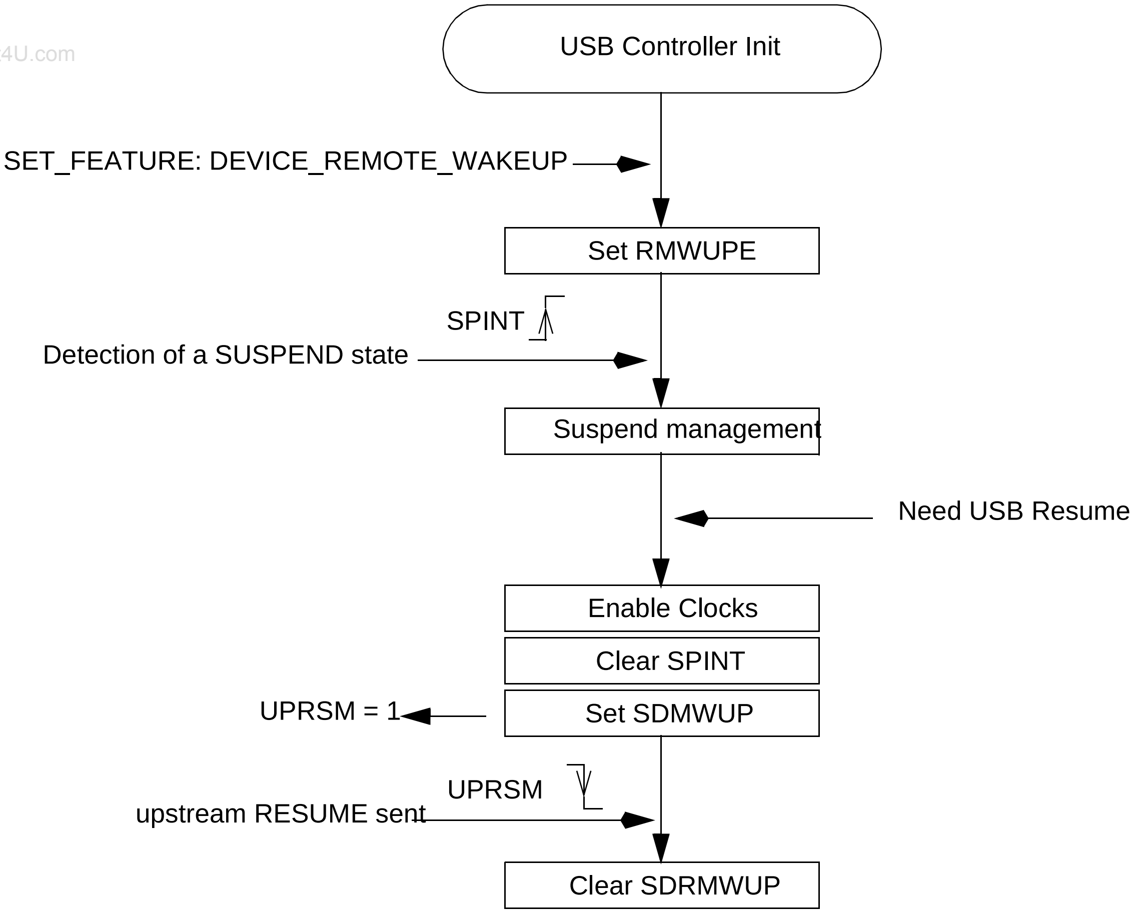
Upstream Resume

A USB device can be allowed by the Host to send an upstream resume for Remote Wake-up purpose.

When the USB controller receives the SET_FEATURE request: DEVICE_REMOTE_WAKEUP, the firmware should set to 1 the RMWUPE bit in the USBCON register to enable this function. RMWUPE value should be 0 in the other cases.

If the device is in SUSPEND mode, the USB controller can send an upstream resume by clearing first the SPINT bit in the USBINT register and by setting then to 1 the SDRMWUP bit in the USBCON register. The USB controller sets to 1 the UPRSM bit in the USBCON register. All clocks must be enabled first. The Remote Wake is sent only if the USB bus was in Suspend state for at least 5 ms. When the upstream resume is completed, the UPRSM bit is reset to 0 by hardware. The firmware should then clear the SDRMWUP bit.

Figure 41. Example of REMOTE WAKEUP Management



Detach Simulation

In order to be re-enumerated by the Host, the AT8xC5122/23 has the possibility to simulate a DETACH-ATTACH of the USB bus.

The V_{REF} output voltage is between 3.0V and 3.6V. This output can be connected to the D+ pull-up as shown in Figure 42. This output can be put in high-impedance when the DETACH bit is set to 1 in the USBCON register. Maintaining this output in high impedance for more than 3 µs will simulate the disconnection of the device. When resetting the DETACH bit, an ATTACH is then simulated. The USB controller should be enabled to use this feature.

Figure 42. Example of V_{REF} Connection

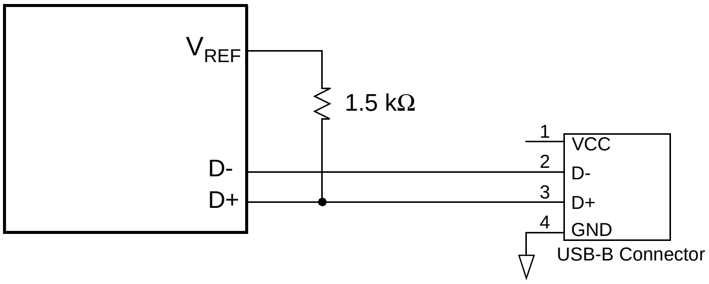
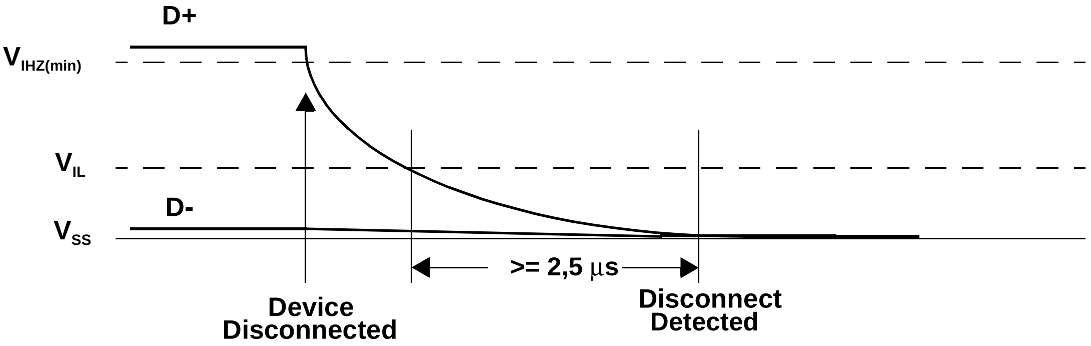


Figure 43. Disconnect Timing



USB Interrupt System

Interrupt System Priorities

Figure 44. USB Interrupt Control System

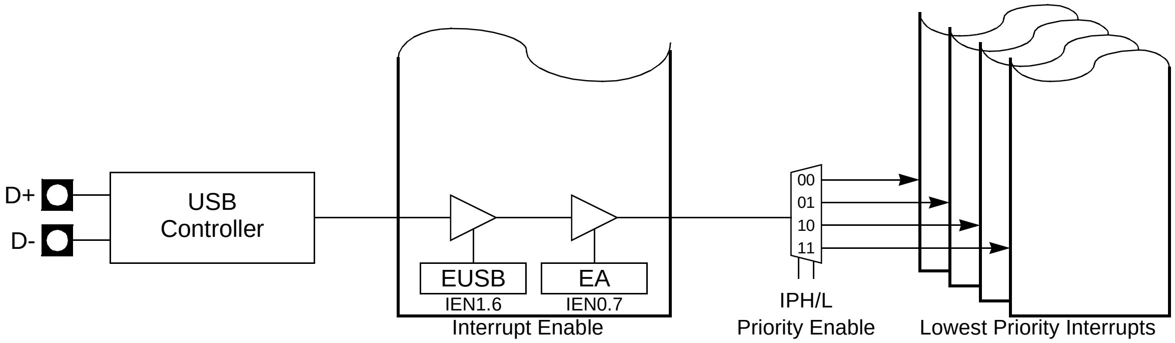


Table 58. Priority Levels

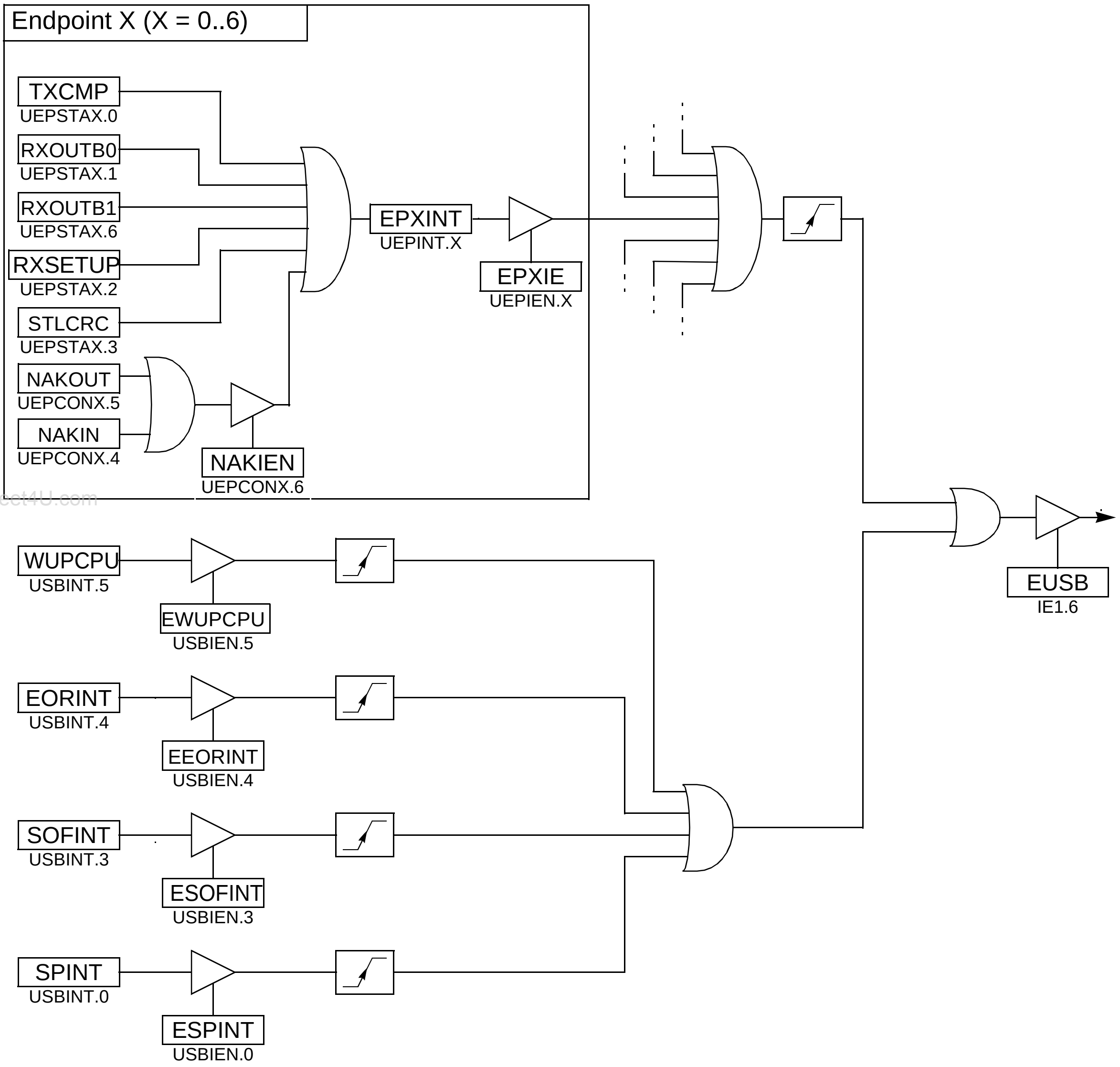
IPHUSB	IPLUSB	USB Priority Level
0	0	0 Lowest
0	1	1
1	0	2
1	1	3 Highest

Interrupt Control System

As shown in Figure 45, many events can produce a USB interrupt:

- TXCMPL: Transmitted In Data (Table 65 on page 83). This bit is set by hardware when the Host accept a In packet.
- RXOUTB0: Received Out Data Bank 0 (Table 65 on page 83). This bit is set by hardware when an Out packet is accepted by the endpoint and stored in bank 0.
- RXOUTB1: Received Out Data Bank 1 (only for Ping-Pong endpoints) (Table 65 on page 83). This bit is set by hardware when an Out packet is accepted by the endpoint and stored in bank 1.
- RXSETUP: Received Setup (Table 65 on page 83). This bit is set by hardware when an SETUP packet is accepted by the endpoint.
- NAKIN and NAKOUT: These bits are set by hardware when a Nak Handshake has been received on the corresponding endpoint. These bits are cleared by software.
- STLCRC: STALLED (only for Control, Bulk and Interrupt endpoints) (Table on page 84). This bit is set by hardware when a STALL handshake has been sent as requested by STALLRQ, and is reset by hardware when a SETUP packet is received.
- SOFINT: Start Of Frame Interrupt (Table 60 on page 79). This bit is set by hardware when a USB start of frame packet has been received.
- WUPCPU: Wake-Up CPU Interrupt (Table 60 on page 79). This bit is set by hardware when a USB resume is detected on the USB bus, after a SUSPEND state.
- SPINT: Suspend Interrupt (Table 60 on page 79). This bit is set by hardware when a USB suspend is detected on the USB bus.

Figure 45. USB Interrupt Control Block Diagram



Registers

Table 59. USB Global Control Register - USBCON (S:BCh)

7	6	5	4	3	2	1	0
USBE	SUSPCLK	SDRMWUP	DETACH	UPRSM	RMWUPE	CONFG	FADDEN

Bit Number	Bit Mnemonic	Description
7	USBE	**USB Enable** Set this bit to enable the USB controller. Clear this bit to disable and reset the USB controller, to disable the USB transceiver an to disable the USB controller clock inputs.
6	SUSPCLK	**Suspend USB Clock** Set this bit to disable the 48MHz clock input (Resume Detection is still active). Clear this bit to enable the 48MHz clock input.
5	SDRMWUP	**Send Remote Wake-up** Set this bit to force an external interrupt on the USB controller for Remote Wake UP purpose. An upstream resume is send only if the bit RMWUPE is set, all USB clocks are enabled AND the USB bus was in SUSPEND state for at least 5 ms. See UPRSM below. This bit is cleared by software.
4	DETACH	**Detach Command** Set this bit to simulate a Detach on the USB line. The V_{REF} pin is then in a floating state. Clear this bit to maintain V_{REF} at 3.3V.
3	UPRSM	**Upstream Resume (read only)** This bit is set by hardware when SDRMWUP has been set and if RMWUPE is enabled. This bit is cleared by hardware after the upstream resume has been sent.
2	RMWUPE	**Remote Wake-Up Enable** Set this bit to enabled request an upstream resume signaling to the host. Clear this bit otherwise. Note: Do not set this bit if the host has not set the DEVICE_REMOTE_WAKEUP feature for the device.
1	CONFG	**Configured** This bit should be set by the device firmware after a SET_CONFIGURATION request with a non-zero value has been correctly processed. It should be cleared by the device firmware when a SET_CONFIGURATION request with a zero value is received. It is cleared by hardware on hardware reset or when an USB reset is detected on the bus (SE0 state for at least 32 Full Speed bit times: typically 2.7 µs).
0	FADDEN	**Function Address Enable** This bit should be set by the device firmware after a successful status phase of a SET_ADDRESS transaction. It should not be cleared afterwards by the device firmware. It is cleared by hardware on hardware reset or when an USB reset is received (see above). When this bit is cleared, the default function address is used (0).

Reset Value = 0000 0000b

Table 60. USB Global Interrupt Register - USBINT (S:BDh)

7	6	5	4	3	2	1	0
-	-	WUPCPU	EORINT	SOFINT	-	-	SPINT

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The value read from these bits is always 0. Do not change these bits.
5	WUPCPU	**Wake-up CPU Interrupt** This bit is set by hardware when the USB controller is in SUSPEND state and is re-activated by a non-idle signal FROM USB line (not by an upstream resume). This triggers a USB interrupt when EWUPCPU is set in the Table on page 80. When receiving this interrupt, user has to enable all USB clock inputs. This bit should be cleared by software (USB clocks must be enabled before).
4	EORINT	**End of Reset Interrupt** This bit is set by hardware when a End of Reset has been detected by the USB controller. This triggers a USB interrupt when EEORINT is set in the Table on page 80. This bit should be cleared by software.
3	SOFINT	**Start Of Frame Interrupt** This bit is set by hardware when an USB Start Of Frame PID (SOF) has been detected. This triggers a USB interrupt when ESOFINT is set in the Table on page 80. This bit should be cleared by software.
2-1	-	**Reserved** The value read from these bits is always 0. Do not change these bits.
0	SPINT	**Suspend Interrupt** This bit is set by hardware when a USB Suspend (Idle bus for three frame periods: a J state for 3 ms) is detected. This triggers a USB interrupt when ESPINT is set in the Table on page 80. This bit should be cleared by software BEFORE any other USB operation to re-activate the macro.

Reset Value = 0000 0000b

Table 61. USB Global Interrupt Enable Register - USBIEN (S:BEh)

7	6	5	4	3	2	1	0
-	-	EWUPCPU	EEORINT	ESOFINT	-	-	ESPINT

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The value read from these bits is always 0. Do not change these bits.
5	EWUPCPU	**Enable Wake-up CPU Interrupt** Set this bit to enable Wake-up CPU Interrupt. Clear this bit to disable Wake-up CPU Interrupt.
4	EEOFINT	**Enable End of Reset Interrupt** Set this bit to enable End of Reset Interrupt. This bit is set after reset. Clear this bit to disable End of Reset Interrupt.
3	ESOFINT	**Enable SOF Interrupt** Set this bit to enable SOF Interrupt. Clear this bit to disable SOF Interrupt.
2-1	-	**Reserved** The value read from these bits is always 0. Do not change these bits.
0	ESPINT	**Enable Suspend Interrupt** Set this bit to enable Suspend Interrupts (See Table 60 on page 79). Clear this bit to disable Suspend Interrupts.

Reset Value = 0001 0000b

Table 62. USB Address Register - USBADDR (S:C6h)

7	6	5	4	3	2	1	0
FEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0

Bit Number	Bit Mnemonic	Description
7	FEN	**Function Enable** Set this bit to enable the function. FADD is reset to 1. Cleared this bit to disable the function.
6-0	UADD[6:0]	**USB Address** This field contains the default address (0) after power-up or USB bus reset. It should be written with the value set by a SET_ADDRESS request received by the device firmware.

Reset Value = 1000 0000b

Table 63. USB Endpoint Number - UEPNUM (S:C7h)

7	6	5	4	3	2	1	0
-	-	-	-	EPNUM3	EPNUM2	EPNUM1	EPNUM0

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The value read from these bits is always 0. Do not change these bits.
3 - 0	EPNUM[3:0]	**Endpoint Number** Set this field with the number of the endpoint which should be accessed when reading or writing to, USB Byte Count Register X (X=EPNUM set in UEPNUM Register) - UBYCTX (S:E2h) or USB Endpoint X Control Register - UEPCONX (S:D4h). This value can be 0, 1, 2, 3, 4, 5 or 6.

Reset Value = 0000 0000b

Table 64. USB Endpoint X Control Register - UEPCONX (S:D4h)

7	6	5	4	3	2	1	0
EPEN	NAKIEN	NAKOUT	NAKIN	DTGL	EPDIR	EPTYPE1	EPTYPE0

Bit Number	Bit Mnemonic	Description
7	EPEN	**Endpoint Enable** Set this bit to enable the endpoint according to the device configuration. Endpoint 0 will always be enabled after a hardware or USB bus reset and participate in the device configuration. Clear this bit to disable the endpoint according to the device configuration.
6	NAKIEN	**NAK Interrupt Enable** Set this bit to enable NAKIN and NAKOUT Interrupt. Clear this bit to disable NAKIN and NAKOUT Interrupt.
5	NAKOUT	**NAK OUT Sent** This bit is set by hardware when the a NAK handshake is sent by the USB controller to an OUT request from the Host. This generates an interrupt if the NAKIEN bit is set. This bit shall be cleared by software.
4	NAKIN	**NAK IN Sent** This bit is set by hardware when the a NAK handshake is sent by the USB controller to an IN request from the Host. This generates an interrupt if the NAKIEN bit is set. This bit shall be cleared by software.
3	DTGL	**Data Toggle (Read-only)** This bit is set by hardware when a valid DATA0 packet is received and accepted. This bit is cleared by hardware when a valid DATA1 packet is received and accepted.
2	EPDIR	**Endpoint Direction** Set this bit to configure IN direction for Bulk, Interrupt and Isochronous endpoints. Clear this bit to configure OUT direction for Bulk, Interrupt and Isochronous endpoints. This bit has no effect for Control endpoints.
1-0	EPTYPE[1:0]	**Endpoint Type** Set this field according to the endpoint configuration (Endpoint 0 will always be configured as control): 00Control endpoint 01Isochronous endpoint 10Bulk endpoint 11Interrupt endpoint

Reset Value = 1000 0000b when UEPNUM = 0
Reset Value = 0000 0000b otherwise

Table 65. USB Endpoint Status and Control Register X - UEPSTAX (S:CEh) X=EPNUM set in UEPNUM Register)

7	6	5	4	3	2	1	0
DIR	RXOUTB1	STALLRQ	TXRDY	STL/CRC	RXSETUP	RXOUTB0	TXCMP

Bit Number	Bit Mnemonic	Description
7	DIR	**Control Endpoint Direction** This bit is used only if the endpoint is configured in the control type (see“USB Endpoint X Control Register - UEPCONX (S:D4h)” on page 82). This bit determines the Control data and status direction. The device firmware should set this bit ONLY for the IN data stage, before any other USB operation. Otherwise, the device firmware should clear this bit.
6	RXOUTB1	**Received OUT Data Bank 1 for Endpoint 6 (Ping-pong Mode)** This bit is set by hardware after a new packet has been stored in the endpoint FIFO Data bank 1 (only in Ping-pong mode). Then, the endpoint interrupt is triggered if enabled (see “USB Global Interrupt Register - USBINT (S:BDh)” on page 79) and all the following OUT packets to the endpoint bank 1 are rejected (NAK'ed) until this bit has been cleared, excepted for Isochronous Endpoints. This bit should be cleared by the device firmware after reading the OUT data from the endpoint FIFO.
5	STALLRQ	**Stall Handshake Request** Set this bit to request a STALL answer to the host for the next handshake. Clear this bit otherwise. For CONTROL endpoints: cleared by hardware when a valid SETUP PID is received.
4	TXRDY	**TX Packet Ready** Set this bit after a packet has been written into the endpoint FIFO for IN data transfers. Data should be written into the endpoint FIFO only after this bit has been cleared. Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet. This bit is cleared by hardware, as soon as the packet has been sent for Isochronous endpoints, or after the host has acknowledged the packet for Control, Bulk and Interrupt endpoints. When this bit is cleared, the endpoint interrupt is triggered if enabled (see Table 60 on page 79).
3	STLCRC	**Stall Sent / CRC error flag** - For Control, Bulk and Interrupt Endpoints: This bit is set by hardware after a STALL handshake has been sent as requested by STALLRQ. Then, the endpoint interrupt is triggered if enabled (see“” on page 79) It should be cleared by the device firmware. - For Isochronous Endpoints **(Read-Only)**: This bit is set by hardware if the last received data is corrupted (CRC error on data). This bit is updated by hardware when a new data is received.
2	RXSETUP	**Received SETUP** This bit is set by hardware when a valid SETUP packet has been received from the host. Then, all the other bits of the register are cleared by hardware and the endpoint interrupt is triggered if enabled (see Table 60 on page 79). It should be cleared by the device firmware after reading the SETUP data from the endpoint FIFO.
1	RXOUTB0	**Received OUT Data Bank 0** (see also RXOUTB1 bit for Ping-pong Endpoints) This bit is set by hardware after a new packet has been stored in the endpoint FIFO data bank 0. Then, the endpoint interrupt is triggered if enabled (see“” on page 79) and all the following OUT packets to the endpoint bank 0 are rejected (NAK'ed) until this bit has been cleared, excepted for Isochronous Endpoints. However, for control endpoints, an early SETUP transaction may overwrite the content of the endpoint FIFO, even if its Data packet is received while this bit is set. This bit should be cleared by the device firmware after reading the OUT data from the endpoint FIFO.
0	TXCMPL	**Transmitted IN Data Complete** This bit is set by hardware after an IN packet has been transmitted for Isochronous endpoints and after it has been accepted (ACK'ed) by the host for Control, Bulk and Interrupt endpoints. Then, the endpoint interrupt is triggered if enabled (see Table 60). This bit should be cleared by the device firmware before setting TXRDY.

Reset Value = 0000 0000b

Table 66. USB FIFO Data Endpoint X (X=EPNUM set in UEPNUM Register) - UEPDATX (S:CFh)

7	6	5	4	3	2	1	0
FDAT7	FDAT6	FDAT5	FDAT4	FDAT3	FDAT2	FDAT1	FDAT0

Bit Number	Bit Mnemonic	Description
7 - 0	FDAT[7:0]	**Endpoint X FIFO data** Data byte to be written to FIFO or data byte to be read from the FIFO, for the Endpoint X (see EPNUM).

Reset Value = XXXX XXXXb

Table 67. USB Byte Count Register X (X=EPNUM set in UEPNUM Register) - UBYCTX (S:E2h)

7	6	5	4	3	2	1	0
-	BYCT6	BYCT5	BYCT4	BYCT3	BYCT2	BYCT1	BYCT0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from these bits is always 0. Do not change this bit.
6 - 0	BYCT[6:0]	**Byte Count LSB** Least Significant Byte of the byte count of a received data packet. This byte count is equal to the number of data bytes received after the Data PID.

Reset Value = 0000 0000b

Table 68. USB Endpoint FIFO Reset Register - UEPRST (S:D5h)

7	6	5	4	3	2	1	0
-	EP6RST	EP5RST	EP4RST	EP3RST	EP2RST	EP1RST	EP0RST

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from these bits is always 0. Do not change this bit.
6	EP6RST	**Endpoint 6 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.
5	EP5RST	**Endpoint 5 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.
4	EP4RST	**Endpoint 4 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.
3	EP3RST	**Endpoint 3 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.
2	EP2RST	**Endpoint 2 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.
1	EP1RST	**Endpoint 1 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.
0	EP0RST	**Endpoint 0 FIFO Reset** Set this bit and reset the endpoint FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received. Then, clear this bit to complete the reset operation and start using the FIFO.

Reset Value = 0000 0000b

Table 69. USB Endpoint Interrupt Register - UEPINT (S:F8h read-only)

7	6	5	4	3	2	1	0
-	EP6INT	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from these bits is always 0. Do not change this bit.
6	EP6INT	**Endpoint 6 Interrupt** This bit is set by hardware when an interrupt is triggered by the (see Table 65 on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.
5	EP5INT	**Endpoint 5 Interrupt** This bit is set by hardware when an interrupt is triggered by the UEPSTAX Register (see Table 65 on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.
4	EP4INT	**Endpoint 4 Interrupt** This bit is set by hardware when an interrupt is triggered by the UEPSTAX Register (see Table 65 on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.
3	EP3INT	**Endpoint 3 Interrupt** This bit is set by hardware when an interrupt is triggered by the UEPSTAX Register (see Table 65 on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.
2	EP2INT	**Endpoint 2 Interrupt** This bit is set by hardware when an interrupt is triggered by the UEPSTAX Register (see Table 65 on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.
1	EP1INT	**Endpoint 1 Interrupt** This bit is set by hardware when an interrupt is triggered by the UEPSTAX Register (see Table 65 on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.
0	EP0INT	**Endpoint 0 Interrupt** This bit is set by hardware when an interrupt is triggered by the UEPSTAX Register (see Table on page 83) and this endpoint interrupt is enabled by the UEPIEN Register (see Table 70 on page 87). This bit is cleared by hardware.

Reset Value = 0000 0000b

Table 70. USB Endpoint Interrupt Enable Register - UEPIEN (S:C2h)

7	6	5	4	3	2	1	0
-	EP6INTE	EP5INTE	EP4INTE	EP3INTE	EP2INTE	EP1INTE	EP0INTE

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from these bits is always 0. Do not change this bit.
6	EP6INTE	**Endpoint 6 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
5	EP5INTE	**Endpoint 5 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
4	EP4INTE	**Endpoint 4 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
3	EP3INTE	**Endpoint 3 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
2	EP2INTE	**Endpoint 2 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
1	EP1INTE	**Endpoint 1 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
0	EP0INTE	**Endpoint 0 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.

Reset Value = 0000 0000b

Serial I/O Port

The serial I/O port in the AT8xC5122/23 is compatible with the serial I/O port in the 80C52.

The I/O port provides both synchronous and asynchronous communication modes. It operates as an 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

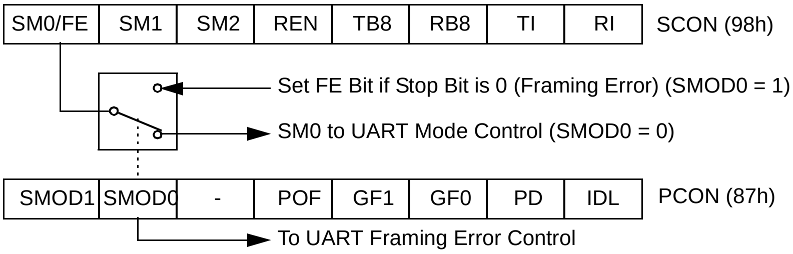
Serial I/O port includes the following enhancements:

- Framing error detection
- Automatic address recognition

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 46).

Figure 46. Framing Error Block Diagram


SM0/FE
SM1
SM2
REN
TB8
RB8
TI
RI
SCON (98h)
Set FE Bit if Stop Bit is 0 (Framing Error) (SMOD0 = 1)
SM0 to UART Mode Control (SMOD0 = 0)
SMOD1
SMOD0
-
POF
GF1
GF0
PD
IDL
PCON (87h)
To UART Framing Error Control

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 Figure 51 on page 92) 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 47 and Figure 48).

Figure 47. UART Timings in Mode 1

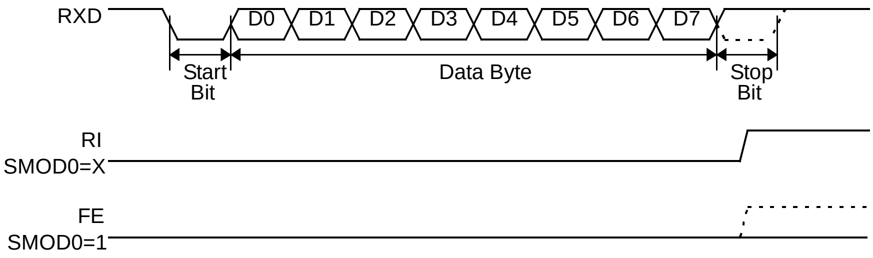
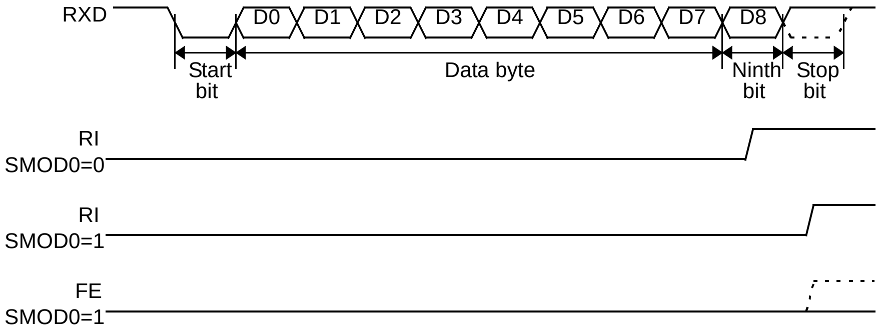


Figure 48. UART Timings in Modes 2 and 3



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, you 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 0010b
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.

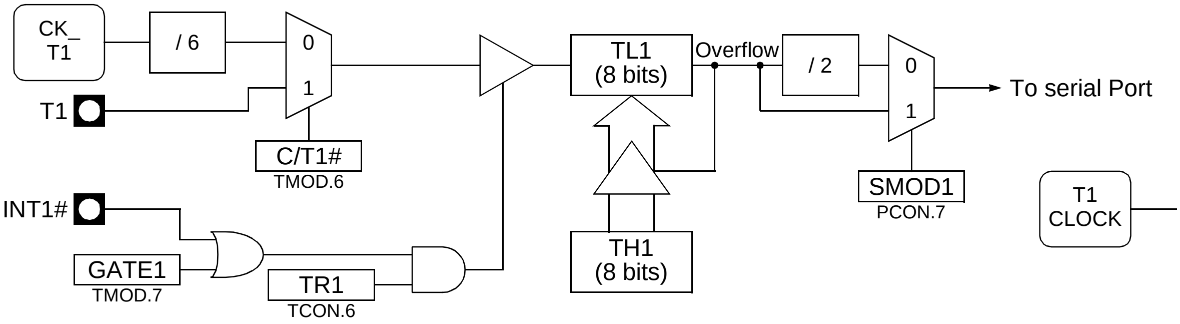
Reset Addresses

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 does not support automatic address recognition.

Timer 1

When using the Timer 1, the Baud Rate is derived from the overflow of the timer. As shown in Figure 49 the Timer 1 is used in its 8-bit auto-reload mode). SMOD1 bit in PCON register allows doubling of the generated baud rate.

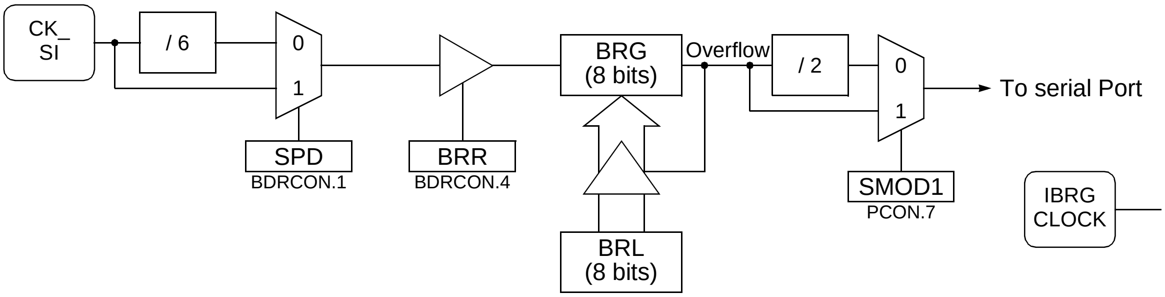
Figure 49. Timer 1 Baud Rate Generator Block Diagram



Internal Baud Rate Generator

When using the Internal Baud Rate Generator, the Baud Rate is derived from the overflow of the timer. As shown in Figure 50 the Internal Baud Rate Generator is an 8-bit auto-reload timer feed by the peripheral clock or by the peripheral clock divided by 6 depending on the SPD bit in BDRCON register (see Figure 76 on page 98). The Internal Baud Rate Generator is enabled by setting BBR bit in BDRCON register. SMOD1 bit in PCON register allows doubling of the generated baud rate.

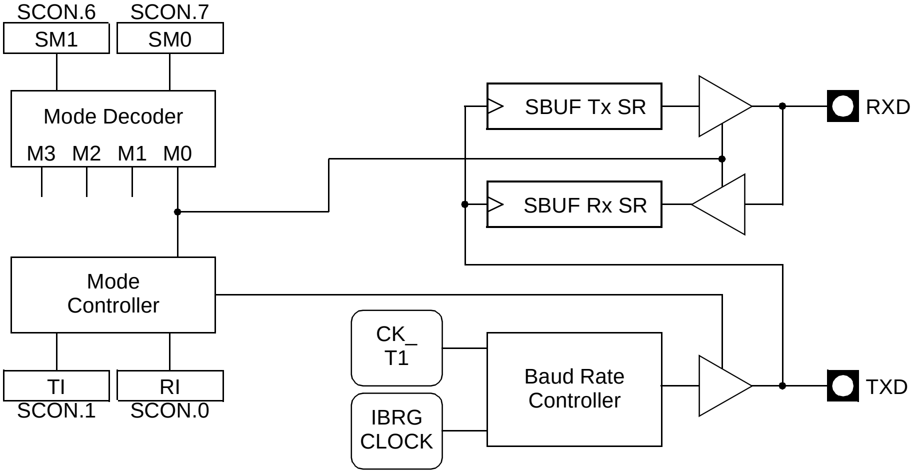
Figure 50. Internal Baud Rate Generator Block Diagram



Synchronous Mode (Mode 0)

Mode 0 is a half-duplex, synchronous mode, which is commonly used to expand the I/0 capabilities of a device with shift registers. The transmit data (TXD) pin outputs a set of eight clock pulses while the receive data (RXD) pin transmits or receives a byte of data. The 8-bit data are transmitted and received least-significant bit (LSB) first. Shifts occur at a fixed Baud Rate (see Section “Baud Rate Selection (Mode 0)”). Figure 51 shows the serial port block diagram in Mode 0.

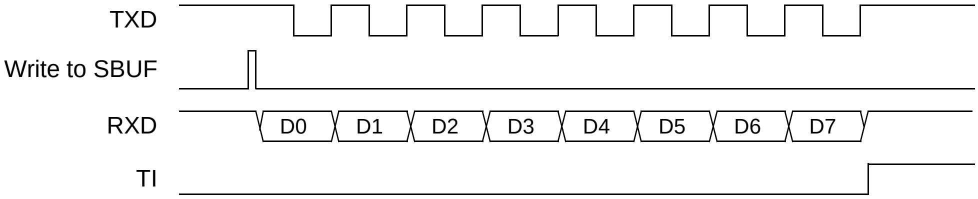
Figure 51. Serial I/O Port Block Diagram (Mode 0)



Transmission (Mode 0)

To start a transmission mode 0, write to SCON register clearing bits SM0, SM1.
As shown in Figure 52, writing the byte to transmit to SBUF register starts the transmission. Hardware shifts the LSB (D0) onto the RXD pin during the first clock cycle composed of a high level then low level signal on TXD. During the eighth clock cycle the MSB (D7) is on the RXD pin. Then, hardware drives the RXD pin high and asserts TI to indicate the end of the transmission.

Figure 52. Transmission Waveforms (Mode 0)

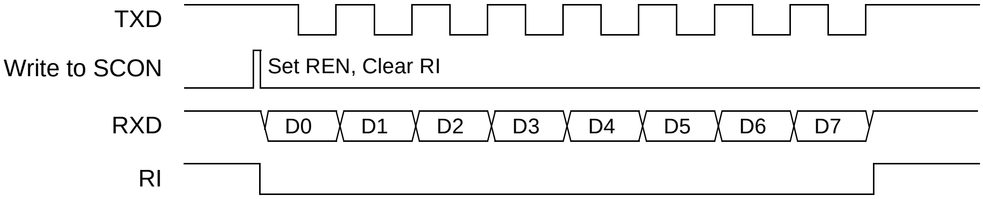


Reception (Mode 0)

To start a reception in mode 0, write to SCON register clearing SM0, SM1 and RI bits and setting the REN bit.

As shown in Figure 53, Clock is pulsed and the LSB (D0) is sampled on the RXD pin. The D0 bit is then shifted into the shift register. After eight sampling, the MSB (D7) is shifted into the shift register, and hardware asserts RI bit to indicate a completed reception. Software can then read the received byte from SBUF register.

Figure 53. Reception Waveforms (Mode 0)



Baud Rate Selection (Mode 0)

In mode 0, baud rate can be either fixed or variable.

As shown in Figure 54, the selection is done using M0SRC bit in BDRCON register.

Figure 55 gives the baud rate calculation formulas for each baud rate source.

Figure 54. Baud Rate Source Selection (Mode 0)

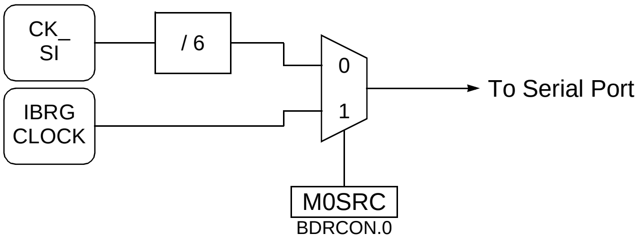


Figure 55. Baud Rate Formulas (Mode 0)

$$\text{Baud\_Rate} = \frac{F_{CK\_SI}}{6}$$

a. Fixed Formula

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

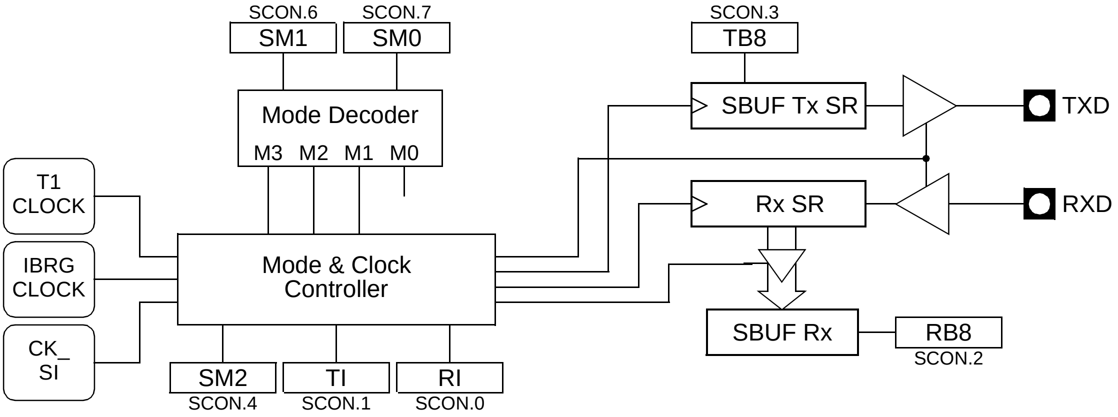
$$BRL = 256 - \frac{2^{SMOD1} \cdot F_{CK\_SI}}{6^{(1-SPD)} \cdot 32 \cdot \text{Baud\_Rate}}$$

b. Variable Formula

Asynchronous Modes (Modes 1, 2 and 3)

The Serial Port has one 8-bit and two 9-bit asynchronous modes of operation. Figure 56 shows the Serial Port block diagram in such asynchronous modes.

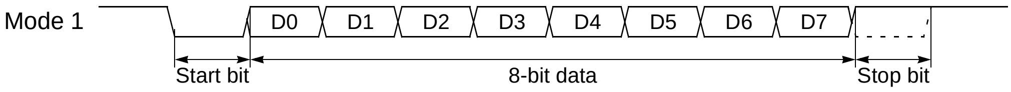
Figure 56. Serial I/O Port Block Diagram (Modes 1, 2 and 3)



Mode 1

Mode 1 is a full-duplex, asynchronous mode. The data frame (see Figure 57) consists of 10 bits: one start, eight data bits and one stop bit. Serial data is transmitted on the TXD pin and received on the RXD pin. When a data is received, the stop bit is read in the RB8 bit in SCON register.

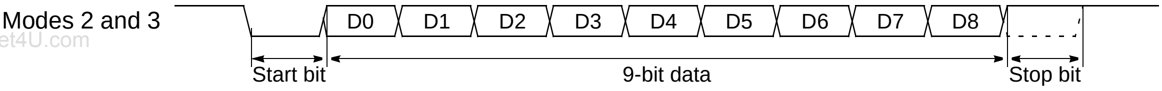
Figure 57. Data Frame Format (Mode 1)



Modes 2 and 3

Modes 2 and 3 are full-duplex, asynchronous modes. The data frame (see Figure 58) consists of 11 bits: one start bit, eight data bits (transmitted and received LSB first), one programmable ninth data bit and one stop bit. Serial data is transmitted on the TXD pin and received on the RXD pin. On receive, the ninth bit is read from RB8 bit in SCON register. On transmit, the ninth data bit is written to TB8 bit in SCON register. Alternatively, you can use the ninth bit as a command/data flag.

Figure 58. Data Frame Format (Modes 2 and 3)



Transmission (Modes 1, 2 and 3)

To initiate a transmission, write to SCON register, setting SM0 and SM1 bits according to Figure 51 on page 92, and setting the ninth bit by writing to TB8 bit. Then, writing the byte to be transmitted to SBUF register starts the transmission.

Reception (Modes 1, 2 and 3)

To prepare for a reception, write to SCON register, setting SM0 and SM1 bits according to Figure 51 on page 92, and setting REN bit. The actual reception is then initiated by a detected high-to-low transition on the RXD pin.

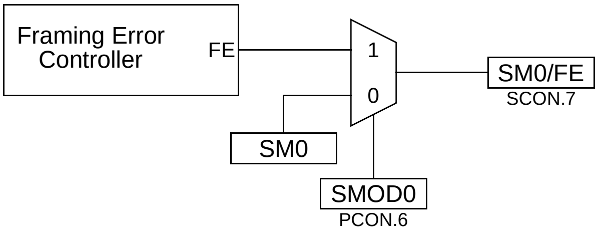
Framing Error Detection (Modes 1, 2 and 3)

Framing error detection is provided for the three asynchronous modes. To enable the framing bit error detection feature, set SMOD0 bit in PCON register as shown in Figure 59.

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 devices. If a valid stop bit is not found, the software sets FE bit in SCON register.

Software may examine FE bit after each reception to check for data errors. Once set, only software or a chip reset clear FE bit. Subsequently received frames with valid stop bits cannot clear FE bit. When the framing error detection feature is enabled, RI rises on stop bit instead of the last data bit as detailed in Figure 57 and Figure 58.

Figure 59. Framing Error Block Diagram



Baud Rate Selection (Modes 1 and 3)

In modes 1 and 3, the Baud Rate is derived either from the Timer 1 or the Internal Baud Rate Generator and allows different baud rate in reception and transmission.

As shown in Figure 60 the selection is done using RBCK and TBCK bits in BDRCON register.

Figure 61 gives the baud rate calculation formulas for each baud rate source while Table 71 details Internal Baud Rate Generator configuration for different peripheral clock frequencies and giving baud rates closer to the standard baud rates.

Figure 60. Baud Rate Source Selection (Modes 1 and 3)

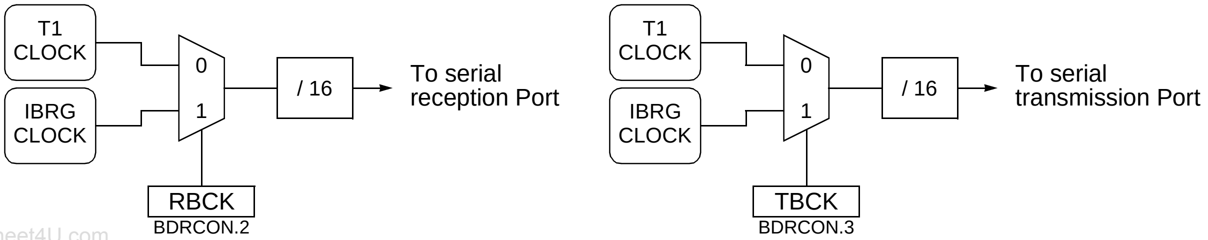


Figure 61. Baud Rate Formulas (Modes 1 and 3)

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

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

a. IBRG Formula

$$\text{Baud\_Rate} = \frac{2^{\text{SMOD1}} \cdot \mathbf{F_{CK\_T1}}}{6 \cdot 32 \cdot (256\ \text{-TH1})}$$

$$\text{TH1} = 256 - \frac{2^{\text{SMOD1}} \cdot \mathbf{F_{CK\_T1}}}{192 \cdot \text{Baud\_Rate}}$$

b. T1 Formula

Table 71. Internal Baud Rate Generator Value

Baud Rate	F_{CK_IDLE}= 6 MHz[1]				F_{CK_IDLE}= 8 MHz[1]				F_{CK_IDLE}= 10 MHz[1]			
	SPD	SMOD1	BRL	Error%	SPD	SMOD1	BRL	Error%	SPD	SMOD1	BRL	Error%
115200	-	-	-	-	-	-	-	-	-	-	-	-
57600	-	-	-	-	1	1	247	3.55	1	1	245	1.36
38400	1	1	246	2.34	1	1	243	0.16	1	1	240	1.73
19200	1	1	236	2.34	1	1	230	0.16	1	1	223	1.36
9600	1	1	217	0.16	1	1	204	0.16	1	1	191	0.16
4800	1	1	178	0.16	1	1	152	0.16	1	1	126	0.16

Baud Rate	F_{CK_IDLE}= 12 MHz[2]				F_{CK_IDLE}= 16 MHz[2]				F_{CK_IDLE}= 20 MHz[2]			
	SPD	SMOD1	BRL	Error%	SPD	SMOD1	BRL	Error%	SPD	SMOD1	BRL	Error%
115200	-	-	-	-	1	1	247	3.55	1	1	245	1.36
57600	1	1	243	0.16	1	1	239	2.12	1	1	234	1.36
38400	1	1	236	2.34	1	1	230	0.16	1	1	223	1.36
19200	1	1	217	0.16	1	1	204	0.16	1	1	191	0.16
9600	1	1	178	0.16	1	1	152	0.16	1	1	126	0.16
4800	1	1	100	0.16	1	1	48	0.16	1	0	126	0.16

Notes: 1. These frequencies are achieved in X1 mode, $F_{CK\_IDLE} = F_{OSC} \div 2$.
2. These frequencies are achieved in X2 mode, $F_{CK\_IDLE} = F_{OSC}$.

Baud Rate Selection (Mode 2)

In mode 2, the baud rate can only be programmed to two fixed values: 1/16 or 1/32 of the peripheral clock frequency.

As shown in Figure 62 the selection is done using SMOD1 bit in PCON register.

Figure 63 gives the baud rate calculation formula depending on the selection.

Figure 62. Baud Rate Generator Selection (Mode 2)

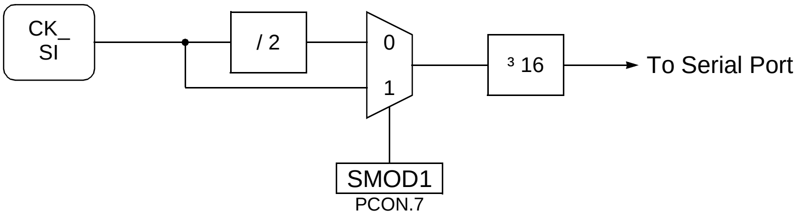


Figure 63. Baud Rate Formula (Mode 2)

$$\text{Baud\_Rate} = \frac{2^{\text{SMOD1}} \cdot F_{CK\_SI}}{32}$$

For mode 0 for UART, thanks to the bit SRC located in BDRCON register (Table 76)

Registers

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** SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register F_{Ck_IDLE}/6 0 1 1 8-bit UART Variable 1 0 2 9-bit UART F_{CK_IDLE} /32 or /16 1 1 3 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 47 and Figure 48 in the other modes.

Reset Value = 0000 0000b (Bit addressable)

Table 72. Slave Address Mask Register for UART - SADEN (B9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 73. Slave Address Register for UART - SADDR (A9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 74. Serial Buffer Register for UART - SBUF (99h)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXb

Table 75. Baud Rate Reload Register for the internal baud rate generator, UART - BRL (9Ah)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Table 76. Baud Rate Control Register - BDRCON - (9Bh)

7	6	5	4	3	2	1	0
-	-	-	BRR	TBCK	RBCK	SPD	SRC

Bit Number	Bit Mnemonic	Description
7 - 5	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
4	BRR	**Baud Rate Run Control bit** Cleared to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK	**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	**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	**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	**Baud Rate Source select bit in Mode 0 for UART** Cleared to select F_{OSC}/12 as the Baud Rate Generator (F_{CL_IDLE}/6 in X2 mode). Set to select the internal Baud Rate Generator for UARTs in mode 0.

Reset Value = XXX0 0000b (Not bit addressable)

Serial Port Interface (SPI)

Only for AT8xC5122.

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

Features

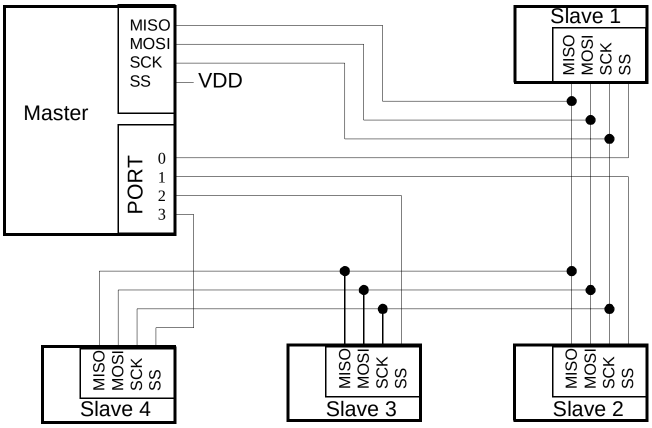
Features of the SPI module include the following:

- Full-duplex, three-wire synchronous transfers
- Master or Slave operation
- Eight programmable Master clock rates
- Serial clock with programmable polarity and phase
- Master Mode fault error flag with MCU interrupt capability
- Write collision flag protection

Signal Description

Figure 64 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 64. Typical SPI Bus



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 movement both in and out 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. Only one Master ($\overline{SS}$ high level) can drive the network. The Master may select each Slave device by software through port pins (Figure 64). 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 (SPSTA) to prevent multiple masters from driving MOSI and SCK (see Section “Error Conditions”, page 6).

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 will be pulled low. Therefore, the MODF flag in the SPSTA 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 comprises 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.

Baud Rate

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

Table 77 gives the different clock rates selected by SPR2:SPR1:SPR0

Table 77. SPI Master Baud Rate Selection

SPR2:SPR1:SPR0	Clock Rate	Baud Rate Divisor (BD)
000	Reserved	N/A
001	F_{CK_SPI} /4	4
010	F_{CK_SPI} / 8	8
011	F_{CK_SPI} /16	16
100	F_{CK_SPI} /32	32
101	F_{CK_SPI} /64	64
110	F_{CK_SPI} /128	128
111	Reserved	N/A

[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.

Functional Description

Figure 65 shows a detailed structure of the SPI module.

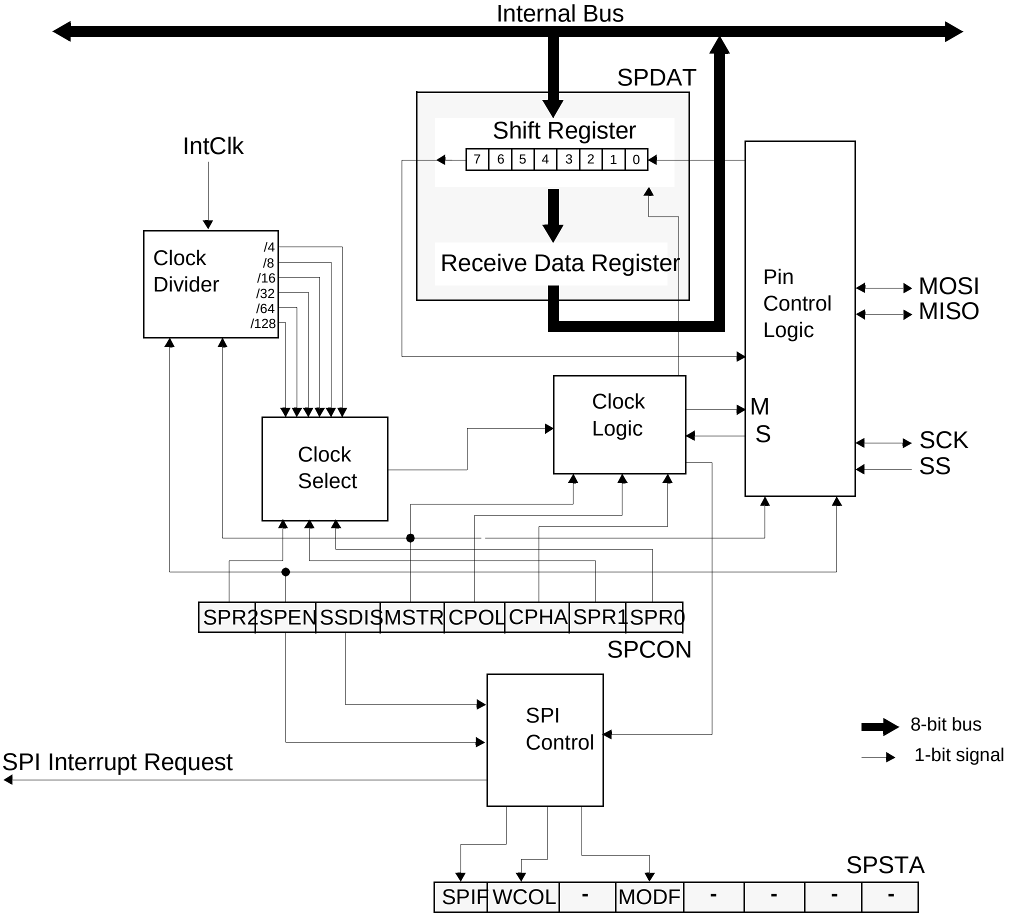


Figure 65. SPI Module Block Diagram

Operating Modes

The Serial Peripheral Interface can be configured as one of the two modes: Master mode or Salve mode. The configuration and initialization of the SPI module is made through one register:

- The Serial Peripheral Control register (SPCON)

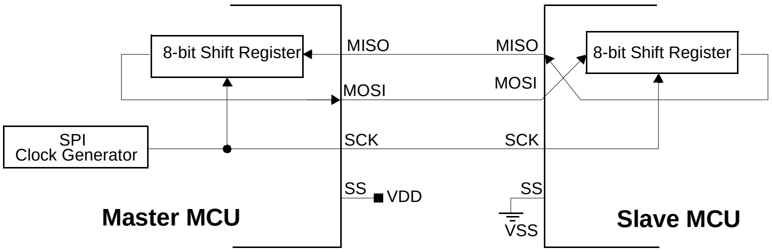
Once the SPI is configured, the data exchange is made using:

- SPCON
- The Serial Peripheral Status register (SPSTA)
- 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 66).

Figure 66. 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 SPSTA 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 (SPSTA) with the SPIF bit set, and then reading the SPDAT.

When the pin SS is pulled down during a transmission, the data is interrupted and when the transmission is established again, the data present in the SPDAT is resent.

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, $\overline{\text{SS}}$, of the Slave device must be set to '0'. $\overline{\text{SS}}$ 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 67 and Figure 68). 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 67. Data Transmission Format (CPHA = 0)

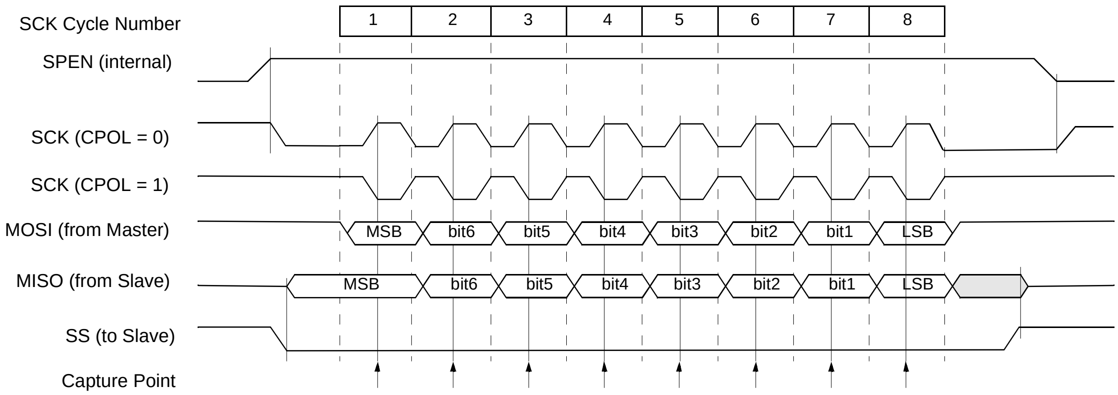
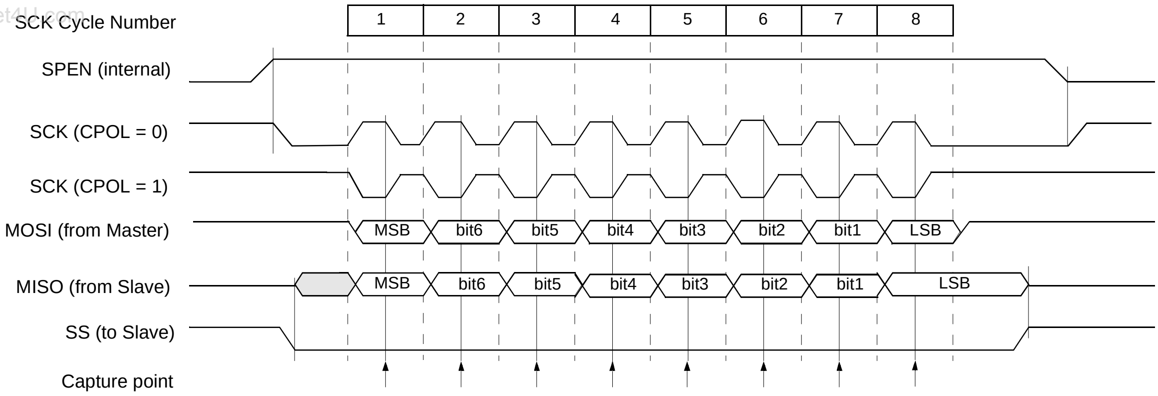


Figure 68. Data Transmission Format (CPHA = 1)



As shown in Figure 67, 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 $\overline{SS}$ pin is used to start the transmission. The $\overline{SS}$ pin must be toggled high and then low between each byte transmitted (Figure 69).

Figure 69. CPHA/$\overline{SS}$ Timing

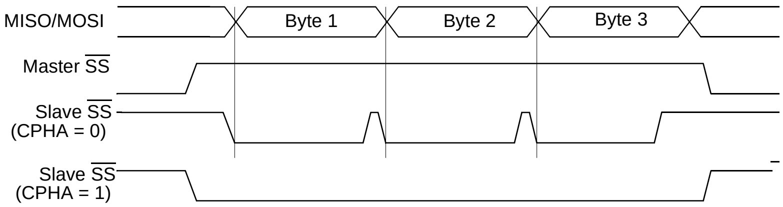


Figure 68 shows an SPI transmission in which CPHA is “1”. In this case, the Master begins driving its MOSI pin on the first SCK edge. Therefore, the Slave uses the first SCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions (Figure 69). This format may be preferable in systems having only one Master and only one Slave driving the MISO data line.

Error Conditions

The following flags in the SPSTA signal SPI error conditions.

Mode Fault (MODF)

MODF error bit in Master mode SPI indicates that the level on the Slave Select ($\overline{SS}$) pin is inconsistent with the actual mode of the device. MODF is set to warn that there may have 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 disable the SPI.
- The MSTR bit in SPCON is cleared.

When $\overline{SS}$ Disable (SSDIS) bit in the SPCON register is cleared, the MODF flag is set when the $\overline{SS}$ signal becomes ’0’.

However, as stated before, for a system with one Master, if the $\overline{SS}$ pin of the Master device is pulled low, there is no way that another Master is attempting to drive the network. In this case, to prevent the MODF flag from being set, software can set the SSDIS bit in the SPCON register and therefore making the $\overline{SS}$ pin as a general-purpose I/O pin.

Clearing the MODF bit is accomplished by a read of SPSTA 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.

Write Collision (WCOL)

A Write Collision (WCOL) flag in the SPSTA is set when a write to the SPDAT register is done during a transmit sequence.

WCOL does not cause an interruption, and the transfer continues uninterrupted.

Clearing the WCOL bit is done through a software sequence of an access to SPSTA and an access to SPDAT.

Overrun Condition

An overrun condition occurs when the Master device tries to send several data bytes and the Slave device has not cleared the SPIF bit issuing from the previous data byte transmitted. In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read of the SPDAT returns this byte. All others bytes are lost.

This condition is not detected by the SPI peripheral.

SS Error Flag (SSERR)

A Synchronous Serial Slave Error occurs when $\overline{SS}$ goes high before the end of a received data in slave mode. SSERR does not cause in interruption, this bit is cleared by writing 0 to SPEN bit (reset of the SPI state machine).

Interrupts

Two SPI status flags can generate a CPU interrupt requests:

Table 78. SPI Interrupts

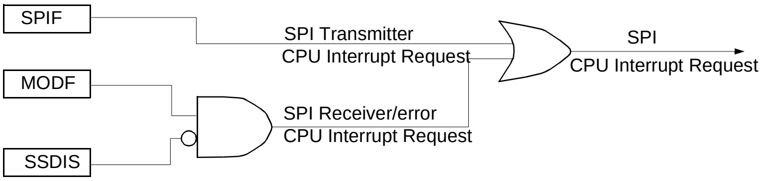
Flag	Request
SPIF (SP data transfer)	SPI Transmitter Interrupt request
MODF (Mode Fault)	SPI Receiver/Error Interrupt Request (if SSDIS = ’0’)

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

Mode Fault flag, MODF: This bit becomes set to indicate that the level on the $\overline{SS}$ is inconsistent with the mode of the SPI. MODF with SSDIS reset, generates receiver/error CPU interrupt requests.

Figure 70 gives a logical view of the above statements.

Figure 70. SPI Interrupt Requests Generation



Registers

There are three registers in the module that provide control, status and data storage functions. These registers are describes in the following paragraphs.

Serial Peripheral Control Register (SPCON)

The Serial Peripheral Control Register does the following:

- Selects one of the Master clock rates
- Configures the SPI module as Master or Slave
- Selects serial clock polarity and phase
- Enables the SPI module
- Frees the $\overline{\text{SS}}$ pin for a general-purpose

Table 79. Serial Peripheral Control Register - SPCON (C3h)

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

Bit Number	Bit Mnemonic	R/W Mode	Description
7	SPR2	RW	**Serial Peripheral Rate 2** Bit with SPR1 and SPR0 define the clock rate
6	SPEN	RW	**Serial Peripheral Enable** Clear to disable the SPI interface (internal reset of the SPI) Set to enable the SPI interface
5	SSDIS	RW	**$\overline{SS}$ Disable** Clear 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’
4	MSTR	RW	**Serial Peripheral Master** Clear to configure the SPI as a Slave Set to configure the SPI as a Master
3	CPOL	RW	**Clock Polarity** Clear to have the SCK set to ’0’ in idle state Set to have the SCK set to ’1’ in idle low
2	CPHA	RW	**Clock Phase** Clear to have the data sampled when the SPSCK leaves the idle state (see CPOL) Set to have the data sampled when the SPSCK returns to idle state (see CPOL)
1	SPR1	RW	**Serial Peripheral Rate** (SPR2:SPR1:SPR0) 000: Reserved 001: F_{CK_SPI} /4 010: F_{CK_SPI}/8 011: F_{CK_SPI}/16
0	SPR0	RW	100: F_{CK_SPI}/32 101: F_{CK_SPI}/64 110: F_{CK_SPI}/128 111: Reserved

Reset Value = 00010100b

Serial Peripheral Status Register (SPSTA)

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 80. Serial Peripheral Status and Control Register - SPSTA (C4h)

7	6	5	4	3	2	1	0
SPIF	WCOL	SSERR	MODF	-	-	-	-

Bit Number	Bit Mnemonic	R/W Mode	Description
7	SPIF	R	**Serial Peripheral data transfer flag** Clear 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.
6	WCOL	R	**Write Collision flag** Cleared by hardware to indicate that no collision has occurred or has been approved by a clearing sequence. Set by hardware to indicate that a collision has been detected.
5	SSERR	R	**Synchronous Serial Slave Error flag** Set by hardware when $\overline{SS}$ is modified before the end of a received data. Cleared by disabling the SPI (clearing SPEN bit in SPCON).
4	MODF	R	**Mode Fault** Cleared by hardware to indicate that the $\overline{SS}$ pin is at appropriate logic level, or has been approved by a clearing sequence. Set by hardware to indicate that the $\overline{SS}$ pin is at inappropriate logic level
3 - 0	-	RW	**Reserved** The value read from this bit is indeterminate. Do not change these bits.

Reset Value = 00X0XXXXb

Serial Peripheral DATa Register (SPDAT)

The Serial Peripheral Data Register (Table 81) 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 81. Serial Peripheral Data Register - SPDAT (C5h)

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

Bit Number	Bit Mnemonic	Description
7-0	R7:0	**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

Reset Value = XXXX XXXXb

Timers/Counters

The AT8xC5122 implements two general-purpose, 16-bit Timers/Counters. Although they are identified as Timer 0, Timer 1, you can independently configure each to operate in a variety of modes as a Timer or as an event Counter. When operating as a Timer, a Timer/Counter runs for a programmed length of time, then issues an interrupt request. When operating as a Counter, a Timer/Counter counts negative transitions on an external pin. After a preset number of counts, the Counter issues an interrupt request.

The Timer registers and associated control registers are implemented as addressable Special Function Registers (SFRs). Two of the SFRs provide programmable control of the Timers as follows:

- Timer/Counter mode control register (TMOD) and Timer/Counter control register (TCON) control respectively Timer 0 and Timer 1.

The various operating modes of each Timer/Counter are described below.

Timer/Counter Operations

For example, 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 the TCON register (see Table 82 on page 114) turns the Timer on by allowing the selected input to increment TLx. When TLx overflows, it increments THx and when THx overflows it sets the Timer overflow flag (TFx) in the 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 the 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 system clock or the external pin Tx as the source for the counted signal. The TRx bit must be cleared when changing the operating mode, otherwise the behavior of the Timer/Counter is unpredictable.

For Timer operation (C/Tx#= 0), the Timer register counts the divided-down system clock. The Timer register is incremented once every peripheral cycle.

Exceptions are the Timer 2 Baud Rate and Clock-Out modes in which the Timer register is incremented by the system clock divided by two.

For Counter operation (C/Tx#= 1), the Timer register counts the negative transitions on the Tx external input pin. The external input is sampled during every S5P2 state. The Programmer’s Guide describes the notation for the states in a peripheral cycle. When the sample is high in one cycle and low in the next one, the Counter is incremented. The new count value appears in the register during the next S3P1 state after the transition has been detected. Since it takes 12 states (24 oscillator periods) to recognize a negative transition, the maximum count rate is 1/24 of the oscillator frequency. 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 an event Counter in four operating modes. Figure 71 through Figure 77 show the logic configuration of each mode.

Timer 0 is controlled by the four lower bits of the TMOD register (see Table 83 on page 115) and bits 0, 1, 4 and 5 of the TCON register (see Table 82 on page 114). The TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (T/C0#) and the operating mode (M10 and M00). The 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 the TF0 flag and generates an interrupt request.

It is important to stop the Timer/Counter before changing modes.

Mode 0 (13-bit Timer)

Mode 0 configures Timer 0 as a 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 the TL0 register (see Figure 71). The upper three bits of the TL0 register are indeterminate and should be ignored. Prescaler overflow increments the TH0 register.

Figure 72 gives the overflow period calculation formula.

Figure 71. Timer/Counter x (x= 0 or 1) in Mode 0

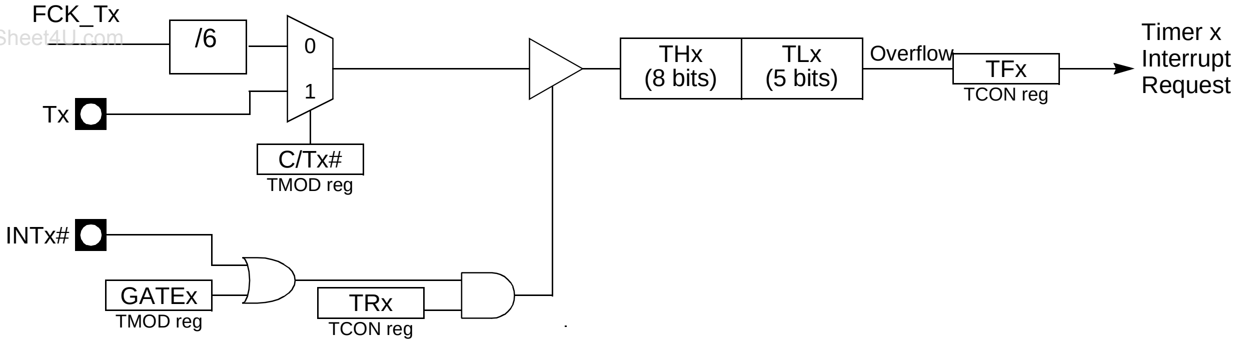


Figure 72. Mode 0 Overflow Period Formula

$$TFx_{PER} = \frac{6 \cdot (16384 - (THx, TLx))}{F_{CK\_Tx}}$$

Mode 1 (16-bit Timer)

Mode 1 configures Timer 0 as a 16-bit Timer with the TH0 and TL0 registers connected in a cascade (see Figure 73). The selected input increments the TL0 register.

Figure 74 gives the overflow period calculation formula when in timer mode.

Figure 73. Timer/Counter x (x = 0 or 1) in Mode 1

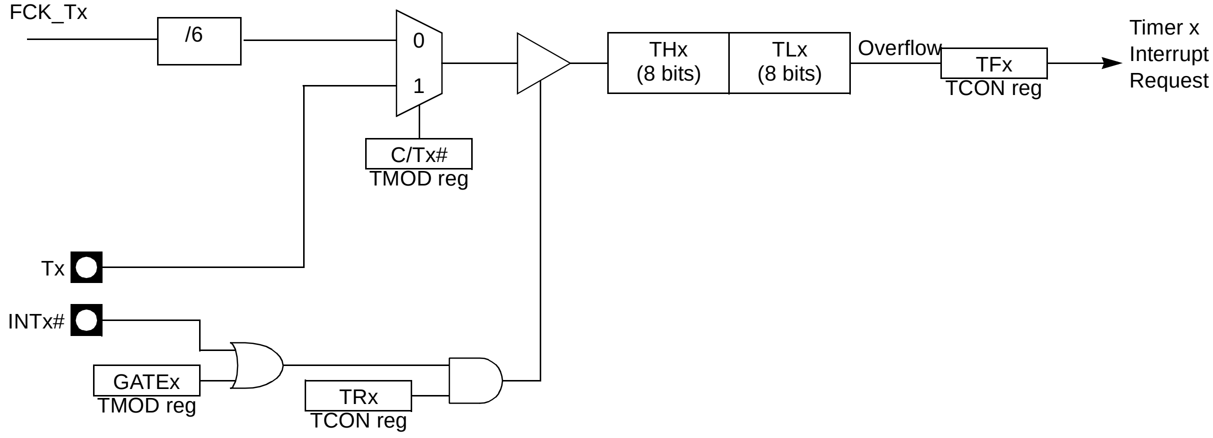


Figure 74. Mode 1 Overflow Period Formula

$$TFx_{PER} = \frac{6 \cdot (65536 - (THx, TLx))}{F_{CK\_Tx}}$$

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

Mode 2 configures Timer 0 as an 8-bit Timer (TL0 register) that automatically reloads from the TH0 register (see Figure 75). TL0 overflow sets the TF0 flag in the TCON register and reloads TL0 with the contents of TH0, which is preset by the software. When the interrupt request is serviced, the hardware clears TF0. The reload leaves TH0 unchanged. The next reload value may be changed at any time by writing it to the TH0 register.

Figure 76 gives the autoreload period calculation formula when in timer mode.

Figure 75. Timer/Counter x (x = 0 or 1) in Mode 2

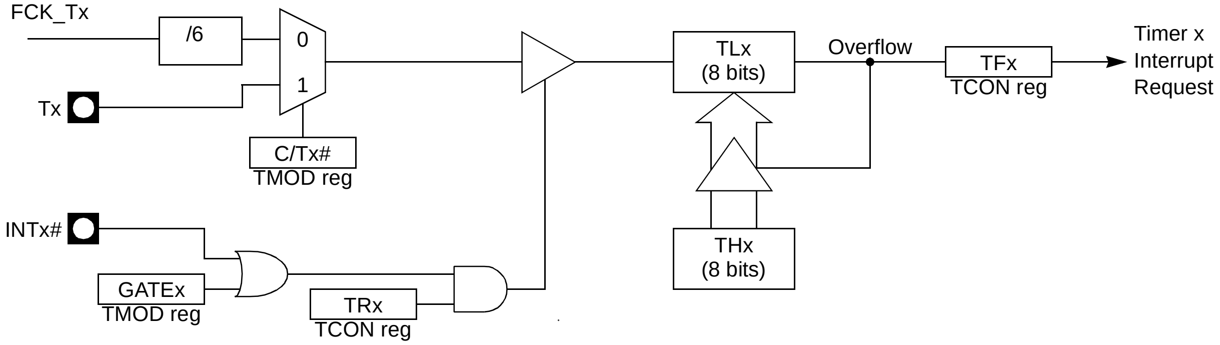


Figure 76. Mode 2 Autoreload Period Formula

$$TFx_{PER} = \frac{6 \cdot (256 - THx)}{F_{CK\_Tx}}$$

Mode 3 (Two 8-bit Timers)

Mode 3 configures Timer 0 so that registers TL0 and TH0 operate as 8-bit Timers (see Figure 77). 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 the TMOD register, and TR0 and TF0 in the TCON register in the normal manner. TH0 is locked into a Timer function (counting F_{UART}) 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 78 gives the autoreload period calculation formulas for both TF0 and TF1 flags.

Figure 77. Timer/Counter 0 in Mode 3: Two 8-bit Counters

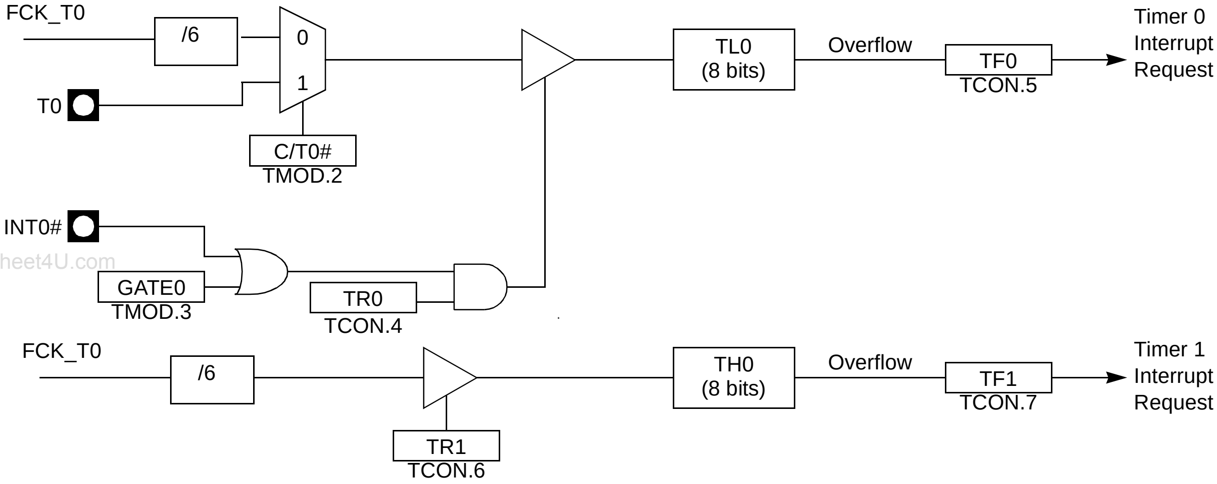


Figure 78. Mode 3 Overflow Period Formula

$$TF0_{PER} = \frac{6 \cdot (256 - TL0)}{F_{CK\_T0}} \qquad TF1_{PER} = \frac{6 \cdot (256 - TH0)}{F_{CK\_T0}}$$

Timer 1

Timer 1 is identical to Timer 0 except 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 an event Counter in three operating modes. Figure 71 through Figure 75 show the logical configuration for modes 0, 1, and 2. Mode 3 of Timer 1 is a hold-count mode.
- Timer 1 is controlled by the four high-order bits of the TMOD register (see Table 83 on page 115) and bits 2, 3, 6 and 7 of the TCON register (see Table 82 on page 114). The TMOD register selects the method of Timer gating (GATE1), Timer or Counter operation (C/T1#) and the operating mode (M11 and M01). The TCON register provides Timer 1 control functions: overflow flag (TF1), run control bit (TR1), interrupt flag (IE1) and the 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 and generates 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 the Timer/Counter before changing modes.

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 71). The upper 3 bits of TL1 register are ignored. Prescaler overflow increments the 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 73). The selected input increments the 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 the TH1 register on overflow (see Figure 75). TL1 overflow sets the TF1 flag in the TCON register and reloads TL1 with the contents of TH1, which is preset by the 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 the TR1 run control bit is not available i.e. when Timer 0 is in mode 3.

Registers

Timer/Counter Control Register

Table 82. TCON (S:88h)

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 the hardware when processor vectors interrupt routine. Set by the 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 the hardware when processor vectors interrupt routine. Set by the 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 the hardware when interrupt is processed if edge-triggered (see IT1). Set by the hardware when external interrupt is detected on the 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 the hardware when interrupt is processed if edge-triggered (see IT0). Set by the 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 83. Timer/Counter Mode Control Register - TMOD (S:89h)

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). Reloaded from TH1 at overflow. 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). Reloaded from TH0 at overflow. 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	

Reset Value = 0000 0000b

Table 84. Timer 0 High Byte Register - TH0 (S:8Ch)

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

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

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

7	6	5	4	3	2	1	0

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

Reset Value = 0000 0000b

Keyboard Interface

Only for AT8xC5122.

Introduction

The AT8xC5122/23 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 P5 and allow to exit from idle and power-down modes.

Description

The keyboard interfaces with the C51 core through 3 special function registers: KBLS, the Keyboard Level Selection register (Table 90 on page 120), KBE, The Keyboard interrupt Enable register (Table 89 on page 119), and KBF, the Keyboard Flag register (Table).

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 79). As detailed in Figure 80 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 allows keyboard arrangement from 1 by n to 8 by n matrix and allows usage of P5 inputs for other purpose.

The KBF.x flags are set by hardware when an active level is on input P5.x. They are automatically reset after any read access on KBF. If the content of KBF must be analyzed, the first read instruction must transfer KBF contend to another location. The KBF register cannot be written by software.

Figure 79. Keyboard Interface Block Diagram

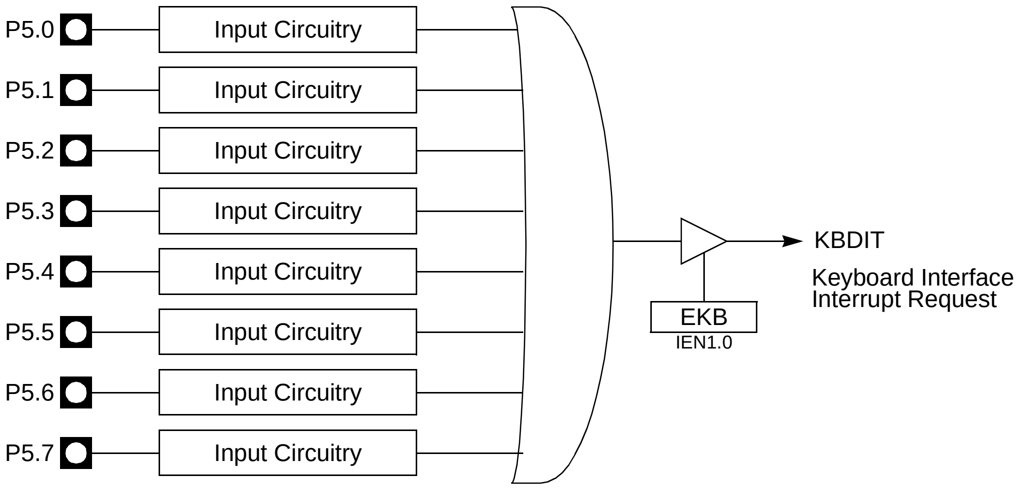
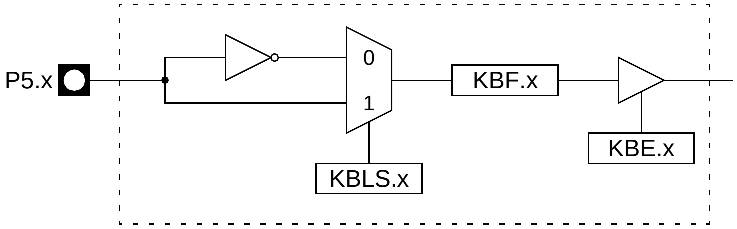


Figure 80. Keyboard Input Circuitry



Power Reduction Mode

P5 inputs allow exit from idle and power-down modes as detailed in Section "Power-Down Mode".

Registers

Table 88. Keyboard Flag Register - KBF (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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.
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. Cleared by hardware after the read of the KBF register.

Reset Value = 0000 0000b

Table 89. Keyboard Input Enable Register - KBE (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 90. Keyboard Level Selector Register - KBLS (9Ch)

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

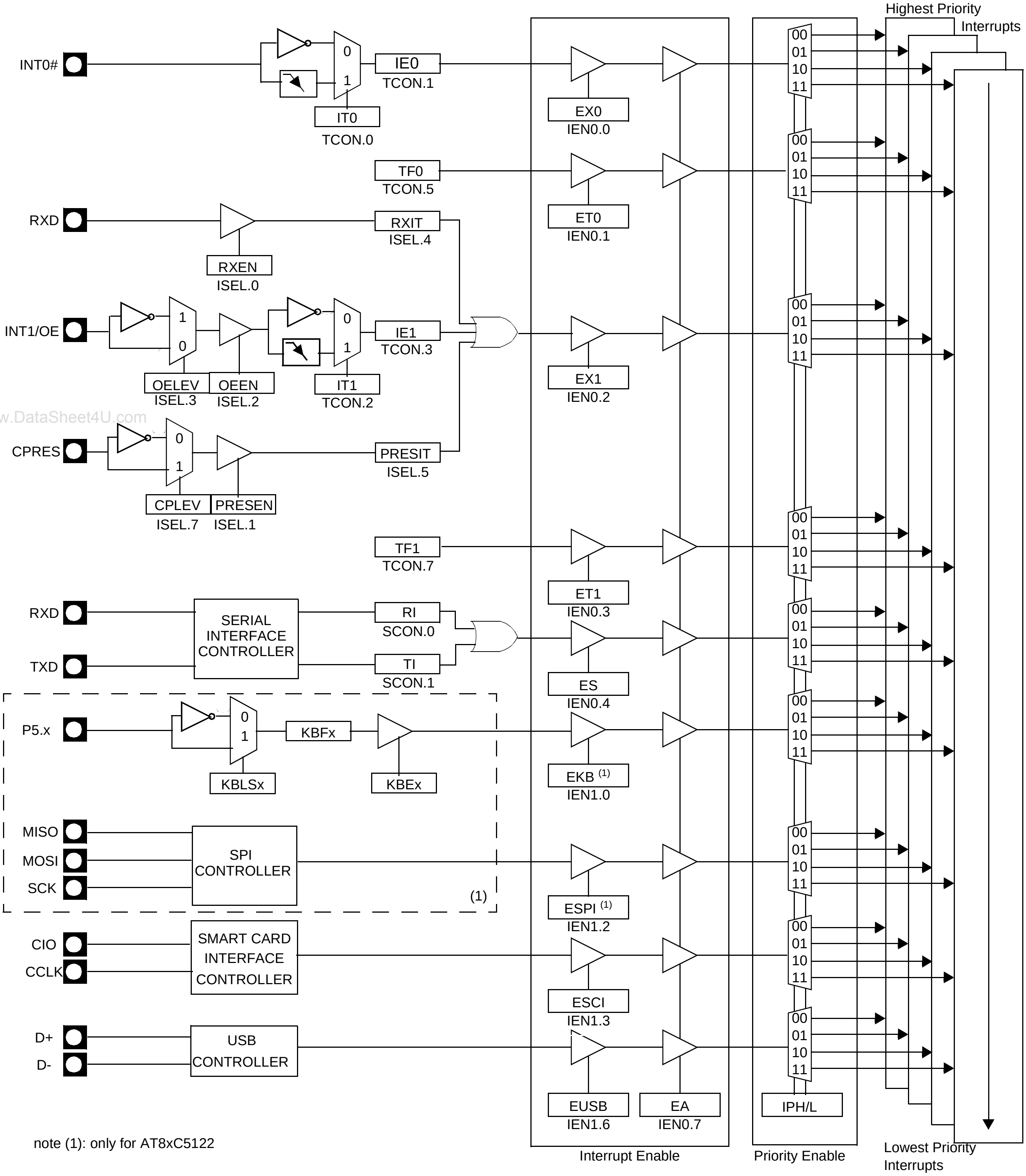
Bit Number	Bit Mnemonic	Description
7	KBLS7	**Keyboard line 7 Level Selection bit** Cleared to enable a low level detection on Port line 7. Set to enable a high level detection on Port line 7.
6	KBLS6	**Keyboard line 6 Level Selection bit** Cleared to enable a low level detection on Port line 6. Set 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

Interrupt System

The AT8xC5122/23 implements an interrupt controller with 15 inputs but only 9 are used: two external interrupts ($\overline{\text{INT0}}$ and $\overline{\text{INT1}}$), two timer interrupts (timers 0, 1), the serial port interrupt, SPI interrupt, Keyboard interrupt, USB interrupt and the SCIB global interrupt.

Figure 81. Interrupt Control System



Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable registers (Table 92 on page 125 and Table 93 on page 126). These registers also contain 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 Low registers (Table 95 on page 127 and Table 97 on page 129) and in the Interrupt Priority High register (Table 96 on page 128 and Table 99 on page 131) shows the bit values and priority levels associated with each combination.

$\overline{\text{INT1}}$ Interrupt Vector

The $\overline{\text{INT1}}$ interrupt is multiplexed with the following three inputs:

- INT1/$\overline{\text{OE}}$: Standard 8051 interrupt input
- RXD: Received data on UART
- CPRES: Insertion or remove of the main card

The setting configurations for each input is detailed below.

$\overline{\text{INT1}}$/OE Input

This interrupt input is active under the following conditions :

- It must be enabled by OEEN Bit (ISEL Register)
- It can be active on a level or falling edge following IT1 Bit (TCON Register) status
- If level triggering selection is set, the active level 0 or 1 can be selected with OELEV Bit (ISEL Register)

The Bit IE1 (TCON Register) is set by hardware when external interrupt detected. It is cleared when interrupt is processed.

RXD Input

A second vector interrupt input is the reception of a character. UART Rx input can generate an interrupt if enabled with Bit RXEN (ISEL.0). The global enable bits EX1 and EA must also be set.

Then, the Bit RXIT (ISEL Register) is set by hardware when a low level is detected on P3.0/RXD input.

CPRES Input

The third input is the detection of a level change on CPRES input (P1.2). This input can generate an interrupt if enabled with PRESEN (ISEL.1) , EX1 (IE0.2) and EA (IE0.7) Bits.

This detection is done according to the level selected with Bit CPLEV (ISEL.7).

Then the Bit PRESIT (ISEL.5) is set by hardware when the triggering conditions are met. This Bit must be cleared by software.

Registers

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 first. Thus within each priority level there is a second priority structure determined by the polling sequence.

Table 91. 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)

Table 92. Interrupt Enable Register 0 - IEN0 (A8h)

7	6	5	4	3	2	1	0
EA	-	-	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 - 5	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
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 93. Interrupt Enable Register 1 - IEN1 (B1h) for AT8xC5122

7	6	5	4	3	2	1	0
-	EUSB	-	-	ESCI	ESPI	-	EKB

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
6	EUSB	**USB Interrupt Enable bit** Cleared to disable USB interrupt . Set to enable USB interrupt.
5 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	ESCI	**SCI interrupt Enable bit** Cleared to disable SCIinterrupt . Set to enable SCI interrupt.
2	ESPI	**SPI interrupt Enable bit** Cleared to disable SPI interrupt . Set to enable SPI interrupt.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0	EKB	**Keyboard interrupt Enable bit** Cleared to disable keyboard interrupt . Set to enable keyboard interrupt.

Reset Value = X0XX 00X0b (Bit addressable)

Table 94. Interrupt Enable Register 1 - IEN1 (B1h) for AT8xC5123

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
6	EUSB	**USB Interrupt Enable bit** Cleared to disable USB interrupt . Set to enable USB interrupt.
5 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	ESCI	**SCI interrupt Enable bit** Cleared to disable SCIinterrupt . Set to enable SCI interrupt.
2		**Reserved** The value read from this bit is indeterminate. Do not change this bit.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0		**Reserved** The value read from this bit is indeterminate. Do not change this bit.

Reset Value = X0XX 0XXXb (Bit addressable)

Table 95. Interrupt Priority Low Register 0 - IPL0 (B8h)

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

Bit Number	Bit Mnemonic	Description
7 - 5	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
4	PSL	**Serial port 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 96. Interrupt Priority High Register 0 - IPH0 (B7h)

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

Bit Number	Bit Mnemonic	Description
7 - 5	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
4	PSH	**Serial port Priority High bit** <u>PSH</u> <u>PSL</u> <u>Priority Level</u> 0 0 Lowest 0 1 1 0 1 1 Highest
3	PT1H	**Timer 1 overflow interrupt Priority High bit** <u>PT1H</u> <u>PT1L</u> <u>Priority Level</u> 0 0 Lowest 0 1 1 0 1 1 Highest
2	PX1H	**External interrupt 1 Priority High bit** <u>PX1H</u> <u>PX1L</u> <u>Priority Level</u> 0 0 Lowest 0 1 1 0 1 1 Highest
1	PT0H	**Timer 0 overflow interrupt Priority High bit** <u>PT0H</u> <u>PT0L</u> <u>Priority Level</u> 0 0 Lowest 0 1 1 0 1 1 Highest
0	PX0H	External interrupt 0 Priority High bit <u>PX0H</u> <u>PX0L</u> <u>Priority Level</u> 0 0 Lowest 0 1 1 0 1 1 Highest

Reset Value = X000 0000b (Not bit addressable)

Table 97. Interrupt Priority Low Register 1 - IPL1 (B2h) for AT8xC5122

7	6	5	4	3	2	1	0
-	PUSBL	-	-	PSCIL	PSPIL	-	PKBDL

Bit Number	Bit Mnemonic	Description
7	-	Reserved The value read from this bit is indeterminate. Do not change this bit.
6	PUSBL	**USB Interrupt Priority bit** Refer to PUSBH for priority level.
5 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	PSCIL	**SCI Interrupt Priority bit** Refer to PSPIH for priority level.
2	PSPIL	**SPI Interrupt Priority bit** Refer to PSPIH for priority level.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0	PKBL	**Keyboard Interrupt Priority bit** Refer to PKBDH for priority level.

Reset Value = X00X 00X0b (Bit addressable)

Table 98. Interrupt Priority Low Register 1 - IPL1 (B2h) for AT8xC5123

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

Bit Number	Bit Mnemonic	Description
7	-	Reserved The value read from this bit is indeterminate. Do not change this bit.
6	PUSBL	**USB Interrupt Priority bit** Refer to PUSBH for priority level.
5 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	PSCIL	**SCI Interrupt Priority bit** Refer to PSPIH for priority level.
2		Reserved The value read from this bit is indeterminate. Do not change this bit.
1		**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0		Reserved The value read from this bit is indeterminate. Do not change this bit.

Reset Value = X0XX 0XXXb (Bit addressable)

Table 99. Interrupt Priority High Register 1 - IPH1 (B3h) for AT8xC5122

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
6	PUSBH	**USB Interrupt Priotity High bit** PUSBH PUSBL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
5-4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	PSCIH	**SCI Interrupt Priority High bit** PSCIH PSCIL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
2	PSPIH	**SPI Interrupt Priority High bit** PSPIH PSPIL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0	PKBH	**Keyboard Interrupt Priority High bit** PKBDH PKBDL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest

Reset Value = XXXX X000b (Not bit addressable)

Table 100. Interrupt Priority High Register 1 - IPH1 (B3h) for AT8xC5123

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
6	PUSBH	**USB Interrupt Priotity High bit** PUSBH PUSBL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
5-4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	PSCIH	**SCI Interrupt Priority High bit** PSCIH PSCIL Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
2		**Reserved** The value read from this bit is indeterminate. Do not change these bits.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0		**Reserved** The value read from this bit is indeterminate. Do not change these bits.

Reset Value = X0XX 0XXXb (Not bit addressable)

Table 101. Interrupt Enable Register - ISEL (S:A1h)

7	6	5	4	3	2	1	0
CPLEV	-	PRESIT	RXIT	OELEV	OEEN	PRESEN	RXEN

Bit Number	Bit Mnemonic	Description
7	CPLEV	**Card presence detection level** This bit indicates which CPRES level will bring about an interrupt Set this bit to indicate that Card Presence IT will appear if CPRES is at high level. Clear this bit to indicate that Card Presence IT will appear if CPRES is at low level.
6	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
5	PRESIT	**Card presence detection interrupt flag** Set by hardware Must be cleared by software
4	RXIT	**Received data interrupt flag** Set by hardware Must be cleared by software
3	OELEV	**OE/INT1 signal active level** Set this bit to indicate that high level is active. Clear this bit to indicate that low level is active.
2	OEEN	**OE/INT1 Interrupt Disable bit** Clear to disable INT1 interrupt Set to enable INT1 interrupt
1	PRESEN	**Card presence detection Interrupt Enable bit** Clear to disable the card presence detection interrupt coming from SCIB. Set to enable the card presence detection interrupt coming from SCIB.
0	RXEN	**Received data Interrupt Enable bit** Clear to disable the RxD interrupt. Set to enable the RxD interrupt (a minimal bit width of 100 µs is required to wake up from power-down) .

Reset Value = 0000 0000b

Interrupt Sources and Vector Addresses

Table 102. Interrupt Vectors

Interrupt Source	Polling Priority at Same Level	Vector Address
Reset	0 (Highest Priority)	C:0000h
INT0	1	C:0003h
Timer 0	2	C:000Bh
INT1	3	C:0013h
Timer 1	4	C:001Bh
UART	6	C:0023h
Reserved	7	C:002Bh
Reserved	5	C:0033h
Keyboard Controller [(1)]	8	C:003Bh
Reserved	9	C:0043h
SPI Controller [(1)]	10	C:004Bh
Smart Card Controller	11	C:0053h
Reserved	12	C:005Bh
Reserved	13	C:0063h
USB Controller	14	C:006Bh
Reserved	15 (Lowest Priority)	C:0073h

Note: 1. Only fot AT8xC5122

Reset and Power Monitor

Reset

From the system point of view, the reset controller must provide the following four functions:

- an active reset each time the reset pin is set to a low state.
- an active reset during power on sequence without the need of external components
- a device protection for preventing code execution if the power supply goes out of the functional range of the microcontroller’s core.
- a watchdog function

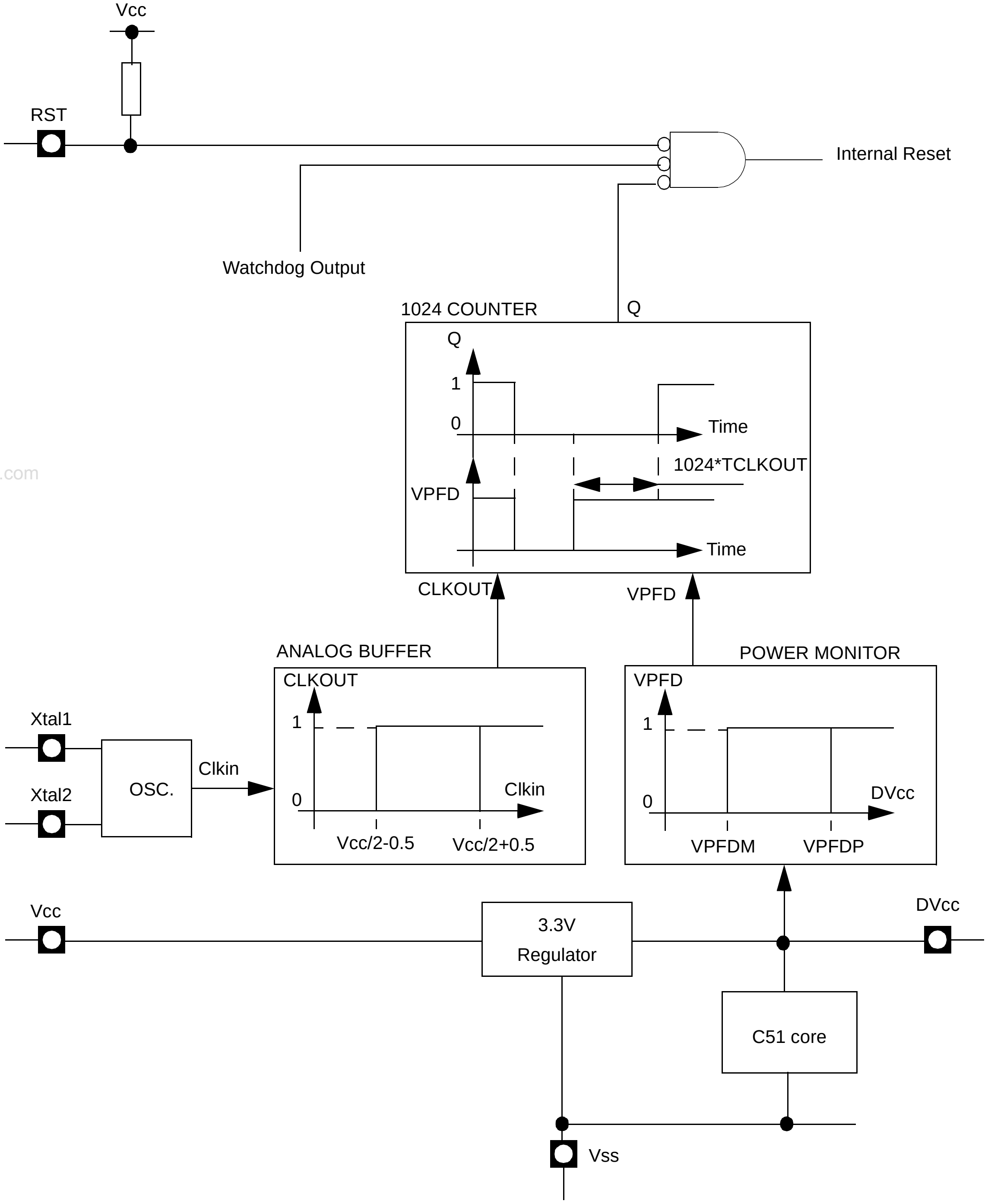
Therefore the RESET controller is fed by three sources:

- Signal coming from Reset pin

- Signal coming from Power Monitor circuit assuring the Power On Reset and the Power Fail detect functions

- Watchdog circuit

Figure 82. Reset Controller



Power Monitor

Overview

The Power Monitor function supervises the evolutions of the voltages feeding the microcontroller, and if needed, suspends its activity when the detected value is out of specification.

It guarantees to start up properly when AT8xC5122 is powered up and prevents code execution errors when the regulated power supply becomes lower than the functional threshold.

This section describes the functions of the Power Monitor.

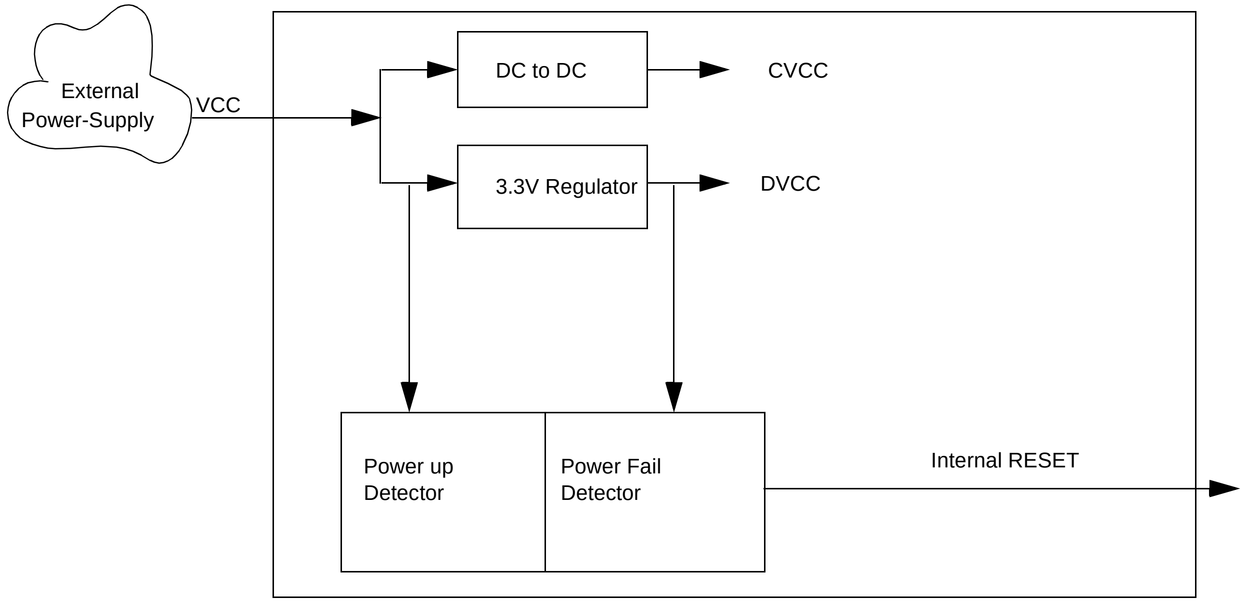
Description

In order to startup and to maintain properly the microcontroller operation, VCC has to be stabilized in the VCC operating range and the oscillator has to be stabilised with a nominal amplitude compatible with logic threshold.

This control is carried out during three phases which are the power-up, normal operation and stop. So it is in accordance with the following requirements:

- it guarantees an operationnal Reset when the microcontroller is powered
- and a protection if the power supply goes out from the functional range of the microcontroller.

Figure 83. Power Monitor Block Diagram



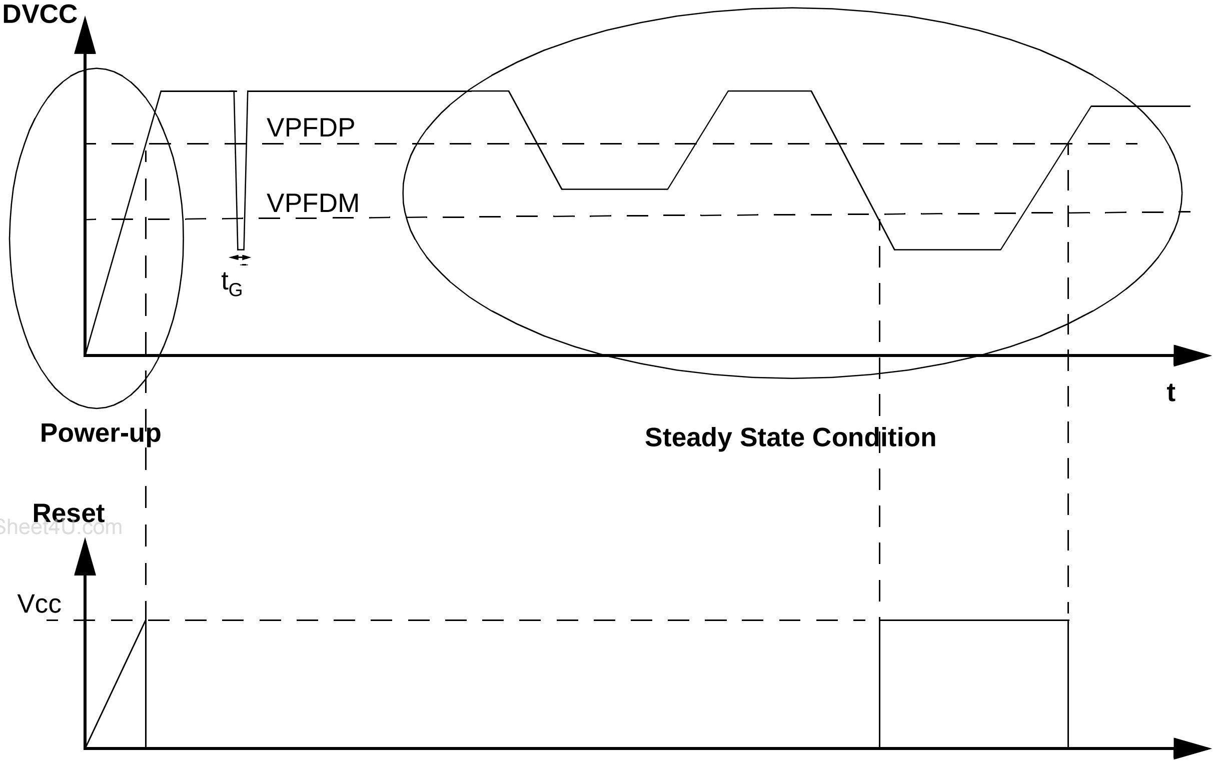
Power Monitor Diagram

The target of the Power monitor is to survey the power-supply in order to detect any voltage drops which are not in the target specification. This Power Monitor checks two kind of situations which occur:

- during the power-up condition, when VCC is reaching the product specification,
- during a steady-state condition, when VCC is stable but disturbed by any undesirable voltage drops.

Figure 84 shows some configurations which can be met by the Power monitor.

Figure 84. Power-up and Steady-state Conditions Monitored



Such device when it is integrated in a microcontroller, forces the CPU in reset mode when VCC reaches a voltage condition which is out of the specification.

The thresholds and their functions are:

- VPFDP: the output voltage of the regulator has reached a minimum functional value at the power-up. The circuit leaves the RESET mode.
- VPFDM: the output voltage of the regulator has reached a low threshold functional value for the microcontroller. An internal RESET is set.

A glitch filtering prevents the system to RESET when short duration glitches are carried on DVCC power-supply.

Watchdog Timer

AT8xC5122 contains a powerfull programmable hardware Watchdog Timer (WDT) that automatically resets the chip if its software fails to reset the WDT before the selected time interval has elapsed. It permits large Timeout ranking from 16 ms to 2s @Fosc = 12 MHz.

This WDT consist of a 14-bit counter plus a 7 - bit programmable counter, a Watchdog Timer reset register (WDTRST) and a Watchdog Timer programmation (WDTPRG) register. When exiting reset, the WDT is -by default- disable. To enable the WDT, the user has to write the sequence 1EH and E1H into WDRST register. When the Watchdog Timer is enabled, it will increment every machine cycle while the oscillator is running and there is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 96xT_{OSC}, where T_{OSC}=1/F_{OSC}. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

The WDT is controlled by two registers (WDTRST and WDTPRG).

Figure 85. Watchdog Timer

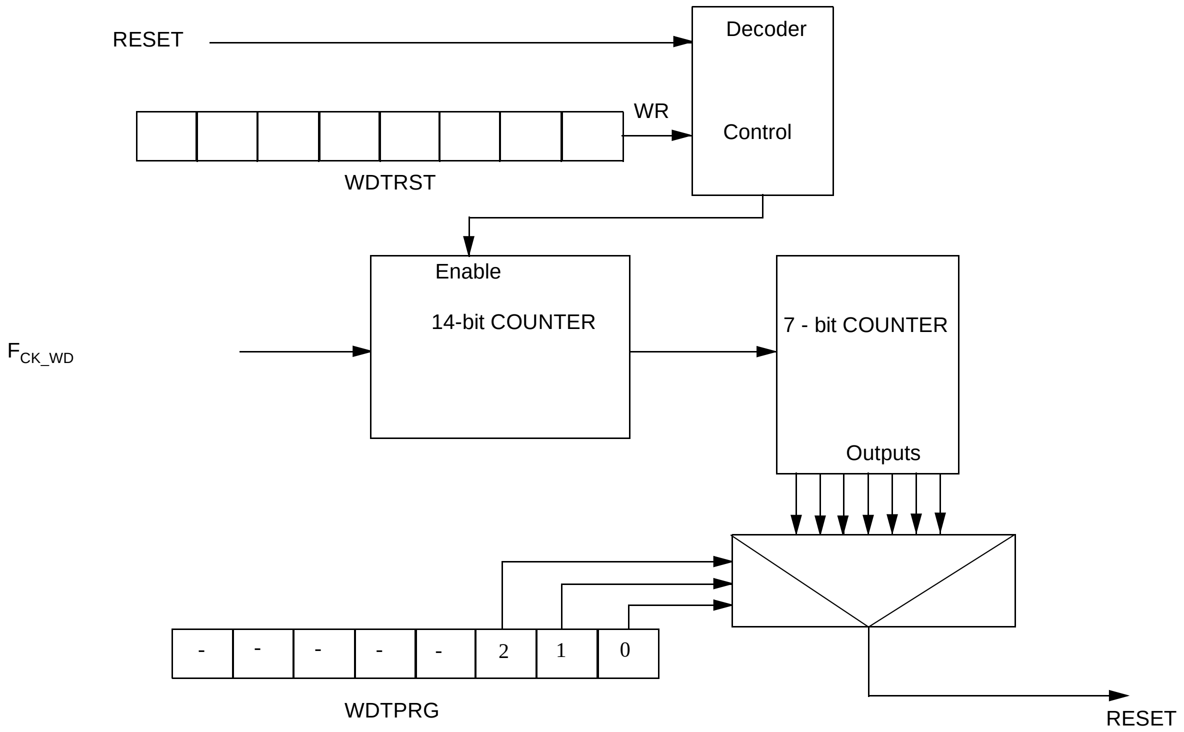


Table 103. Watchdog Timer Out Register - WDTPRG (0A7h)

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

Bit Number	Bit Mnemonic	Description
7 - 3	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
2	S2	WDT Time-out select bit 2
1	S1	WDT Time-out select bit 1
0	S0	WDT Time-out select bit 0

Reset Value = XXXX X000b

The three lower bits (S0, S1, S2) located into WDTPRG register enables to program the WDT duration.

Table 104. Machine Cycle Count

S2	S1	S0	Machine Cycle Count
0	0	0	2^{14} - 1
0	0	1	2^{15} - 1
0	1	0	2^{16} - 1
0	1	1	2^{17} - 1
1	0	0	2^{18} - 1
1	0	1	2^{19} - 1
1	1	0	2^{20} - 1
1	1	1	2^{21} - 1

To compute WD Timeout, the following formula is applied:

$$\text{Time Out} = 6 * (2^{14} * 2^{\text{Svalue}} - 1) / F_{CK\_WD}$$

Note: Svalue represents the decimal value of (S2 S1 S0) / CKRL represents the Prescaler

Table 105. Timeout value for Fosc = 12 MHz

S2	S1	S0	Timeout for F_{CK_WD}= 6 MHz
0	0	0	16.38 ms
0	0	1	32.77 ms
0	1	0	65.54 ms
0	1	1	131.07 ms
1	0	0	262.14 ms
1	0	1	524.29 ms
1	1	0	1.05 s
1	1	1	2.10 s

Table 106. Watchdog Timer Enable register (Write Only) - WDTRST (A6h)

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

Reset Value = XXXX XXXXb

The WDTRST register is used to reset/enable the WDT by writing 1EH then E1H in sequence.

Watchdog Timer During Power-down Mode 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 does whenever AT8xC5122 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 for the interrupt used to exit Power-down.

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

In the Idle mode, the oscillator continues to run. To prevent the WDT from resetting 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.

Power Management

Idle Mode

An instruction that sets PCON.0 indicates that it is the last instruction to be executed before going into the Idle mode. In the 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 occured during normal operation or during an 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 13, PCON register).

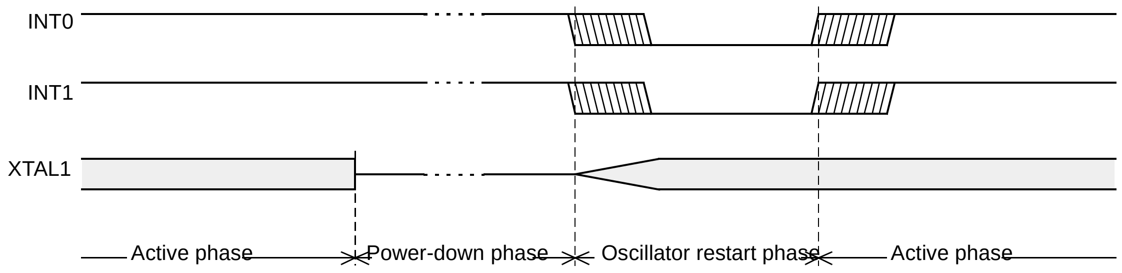
WARNING: To minimize power consumption, all peripherals and I/Os with static current consumption must be set in the proper state. I/Os programmed with low speed output configuration (KB_OUT) must be switch to push-pull or Standard C51 configuration before entering power-down. The CVCC generator must also be switch off.

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 $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, Keyboard, Card insertion/removal and USB 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 86. 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 is 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 be the one following the instruction that put AT8xC5122/23 into power-down mode.

Figure 86. Power-down Exit Waveform



Exit from power-down by reset redefines all the SFRs, exit from power-down by external interrupt does no affect the SFRs.

Exit from power-down by either reset or external interrupt does not affect the internal RAM content.

Note: If idle mode is activated with power-down mode (IDL and PD bits set), the exit sequence is unchanged, when execution is vectored to interrupt, PD and IDL bits are cleared and idle mode is not entered.

Table 107 shows the state of ports during idle and power-down modes.

Table 107. State of Ports

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3	PORTI2
Idle	Internal	1	1	Port Data[(1)]	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data	Port Data
Power-down	Internal	0	0	Port Dat*	Port Data	Port Data	Port Data	Port Data
Power-down	External	0	0	Floating	Port Data	Port Data	Port Data	Port Data

Note: 1. Port 0 can force a 0 level. A "one" will leave port floating.

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.

USB Interface

Suspend

The Suspend state can be detected by the USB controller if all the clocks are enabled and if the USB controller is enabled. The bit SPINT is set by hardware when an idle state is detected for more than 3 ms. This triggers a USB interrupt if enabled.

In order to reduce current consumption, the firmware can put the USB PAD in idle mode, stop the clocks and put the C51 in Idle or Power-down mode. The Resume detection is still active.
The USB PAD is put in idle mode when the firmware clear the SPINT bit. In order to avoid a new suspend detection 3ms later, the firmware has to disable the USB clock input using the SUSPCLK bit in the USBCON Register. The USB PAD automatically exits of idle mode when a wake-up event is detected.

The stop of the 48 MHz clock from the PLL should be done in the following order:

1. Disable of the 48 MHz clock input of the USB controller by setting to 1 the SUSPCLK bit in the USBCON register.
2. Disable the PLL by clearing the PLLEN bit in the PLLCON register.

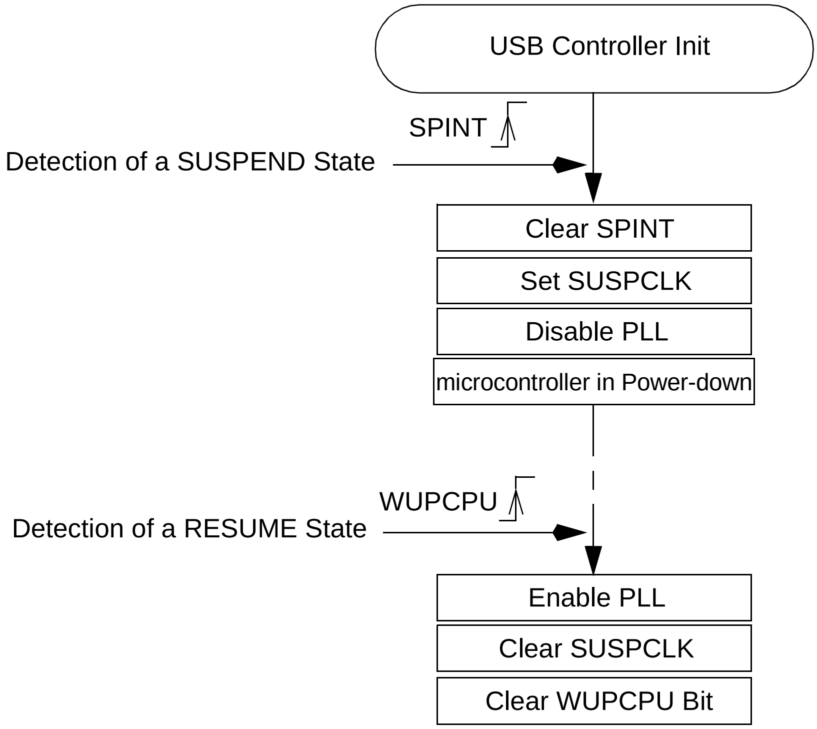
Resume

When the USB controller is in Suspend state, the Resume detection is active even if all the clocks are disabled and if the C51 is in Idle or Power-down mode. The WUPCPU bit is set by hardware when a non-idle state occurs on the USB bus. This triggers an interrupt if enabled. This interrupt wakes up the CPU from its Idle or Power-down state and the interrupt function is then executed. The firmware will first enable the 48 MHz generation and then reset to 0 the SUSPCLK bit in the USBCON register if needed.

The firmware has to clear the SPINT bit in the USBINT register before any other USB operation in order to wake up the USB controller from its Suspend mode.

The USB controller is then re-activated.

Figure 87. Example of a Suspend/Resume Management



Smart Card Interface

Entering in Power-down Mode

In order to reduce the power consumption, a power-down or idle mode can be invoked by software (see Table 13, PCON register). Before activating these modes the application will need to:

Power-off the Smart Card Interface by applying the following sequence:

- Set CRST pin at low level by clearing the bit CARDRST in SCCON register.
- Set CCLK pin at low level by clearing the bit CLK then the CARDCLK in SCCON register.
- Set CIO pin at low level by clearing the bit UART in SCICR register then the bit CARDIO in SCCON register.
- Power the Smart Interface off by clearing the CARDVCC bit in SCCON register. This instruction enables to switch DC/DC converter off.

CPRES input:

- Set the bit PRSEN in ISEL register
- Set the bit EX1 in IE0 register
- Set the bit EA in the IE0 register
- Invert the bit CPLEV in ISEL register (INT1 interrupt vector)
- Clear the bit PRESIT in the ISEL register

Exiting from Power-down Mode

The microcontroller will exit from Power-down or Idle modes upon a reset or INT1 interrupt which is a multiplexing of the interruptions generated by the CPRES pin (Card detection), RxD flag (UART reception) and INT1 pin.

Keyboard Interface

Only for AT8xC5122.

Entering in Power-down Mode

In order to reduce the power consumption, the microcontroller can be set in power-down or idle mode by software (see Table 13, PCON register). Before activating these modes the application will need to configure the keyboard interface as follows:

- Set all keyboard's ouputs pins KB Rx at low level by writing a 0 on the ports. This operation has a double effect:
  - any key that is pressed generates an interrupt capable of waking-up the microcontroller,
  - Set all bits KBE.x in KBE registers to enable interrupts.

Exiting from Power-down Mode

The microcontroller will exit from Power-down Mode upon a reset or any interrupt generated by a key press. Note that 1024 clocks are necessary to exit from power-down mode when a keyboard interrupt occurs. This means that there will be a delay between the time at which the key is pressed and the time at which the application is able to identify the key.

Registers

Table 108. Auxiliary Register - AUXR (8Eh)

7	6	5	4	3	2	1	0
DPU	-	-	-		XRS0	EXTRAM	AO

Bit Number	Bit Mnemonic	Description
7	DPU	**Disable weak Pull-up** Reset weak pull-up is enable Set weak pull-up is disable
6-3	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
2	XRS0	**XRAM Size** 0 256 bytes (default) 1 512 bytes
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 instructione is used.

Reset Value = 0XXX X000b

Table 109. Power Control Register - PCON (S: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 for UART** Set to select double baud rate in mode 1,2 or 3
6	SMOD0	**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 change 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

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.

Data Memory Management

Expanded RAM (XRAM)

The AT8xC5122/23 and AT8xC5123 provides additional Bytes of random access memory (RAM) space for increased data parameter handling and high level language usage.

AT8xC5122/23 and AT8xC5123 devices have expanded RAM in external data space; maximum size and location are described in Table 110.

Table 110. Description of Expanded RAM

		Address	
	XRAM size	Start	End
AT8xC5122 and AT8xC5123	512	00h	1FFh

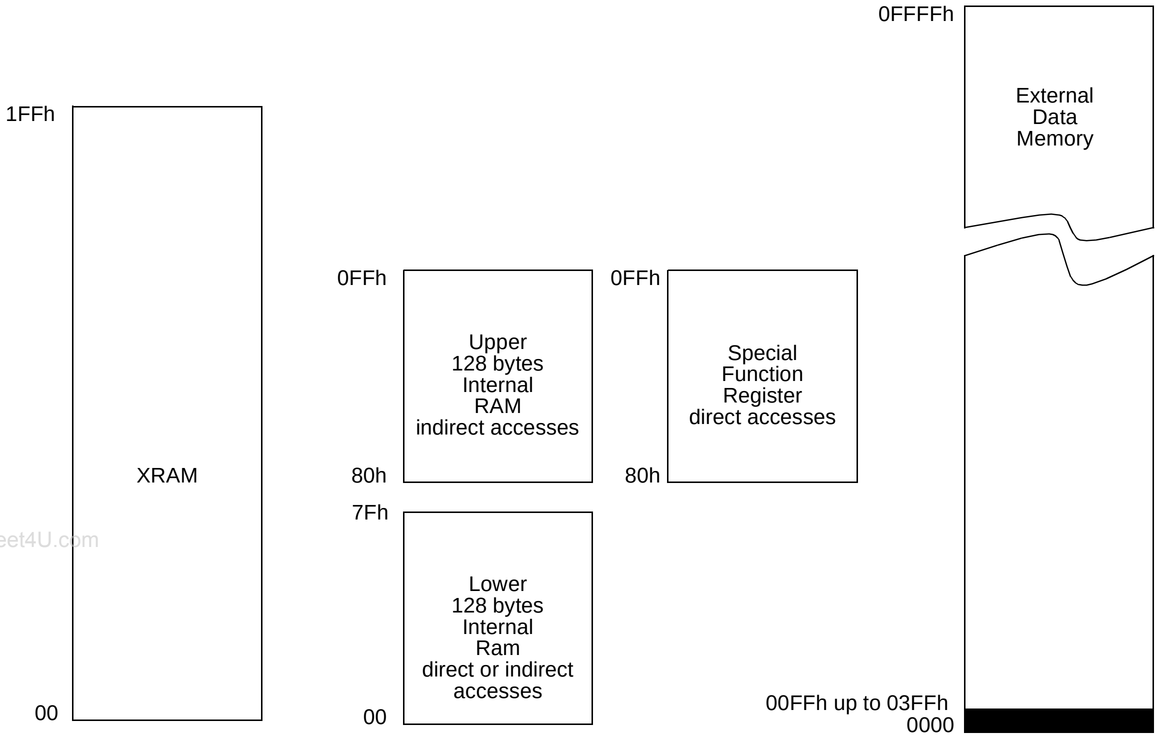
The AT8xC5122/23 and AT8xC5123 have internal data memory that is mapped into four separate segments.

The four segments are:

- The Lower 128 bytes of RAM (addresses 00h to 7Fh) are directly and indirectly addressable.
- The Upper 128 bytes of RAM (addresses 80h to FFh) are indirectly addressable only.
- The Special Function Registers, SFRs, (addresses 80h to FFh) are directly addressable only.
- The expanded RAM bytes are indirectly accessed by MOVX instructions, and with the EXTRAM bit cleared in the AUXR register (see Table 108 on page 146)

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 88. Internal and External Data Memory Address



When an instruction accesses an internal location above address 7Fh, the CPU knows whether the access is in 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 bit XRS0 is used to hide a part of the available XRAM as explained in Table 108 on page 146. 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. **A**n 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 Port 0 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. Port 2 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 not be located in the XRAM.

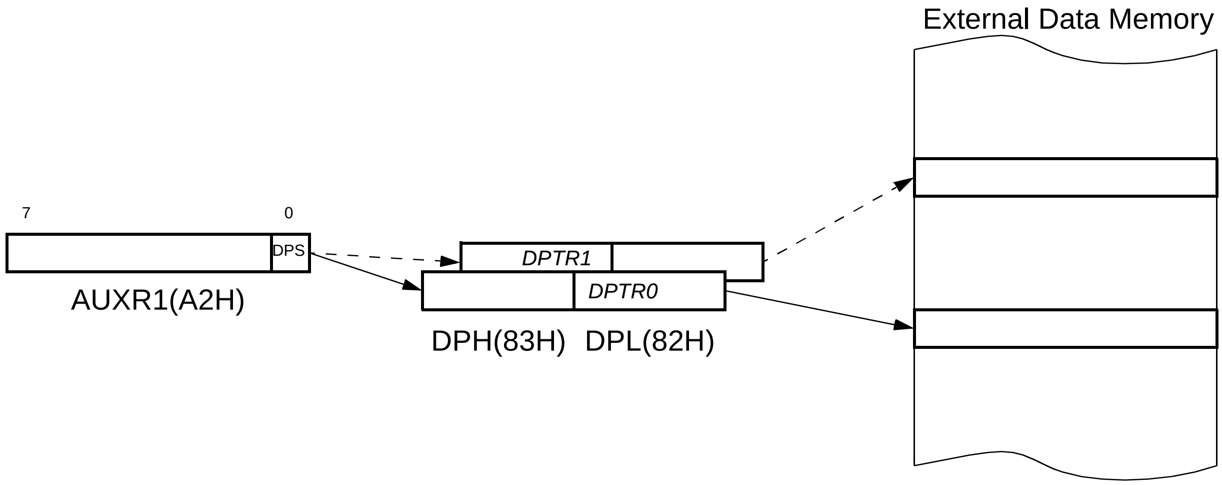
The M0 bit allows to stretch the XRAM timings; if M0 is set, the read and write pulses are extended from 6 to 30 clock periods. This is useful to access external slow peripherals.

Dual Data Pointer Register (DDPTR)

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 112) that allow the program code to switch between them (Figure 89).

Figure 89. Use of Dual Pointer



a. 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 QU 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. For example, 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.

Registers

See Table 108 on page 146 for the definition of AUXR register.

Table 111. Auxiliary Register 1 AUXR1- (0A2h) for AT8xC5122

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

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
5	ENBOOT	**Enable Boot ROM (ROM/CRAM version only)** Set this bit to map the Boot ROM from 8000h to FFFFh. If the PC increments beyond 7FFFh address, the code is fetch from internal ROM Clear this bit to disable Boot ROM. If the PC increments beyond 7FFFh address, the code is fetch from external code memory (C51 standard roll over function) This bit is forced to 1 at reset
4	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
3	GF3	This bit is a general-purpose user flag.
2	0	Always cleared.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0	DPS	**Data Pointer Selection** Cleared to select DPTR0. Set to select DPTR1.

Reset Value = XX1X XX0X0b (Not bit addressable)

Table 112. Auxiliary Register 1 AUXR1- (0A2h) for AT8xC5123

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

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
5		**Reserved** The value read from this bit is indeterminate. Do not change these bits.
4	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
3	GF3	This bit is a general-purpose user flag.
2	0	Always cleared.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0	DPS	**Data Pointer Selection** Cleared to select DPTR0. Set to select DPTR1.

Reset Value = XXXX XX0X0b (Not bit addressable)

Program Memory Management

The AT8xC5122/23 will be available in two configurations:

- CRAM / ROM
- ROM
- Both configurations provide:
  - 32K Bytes of Internal ROM (only 30K Bytes for AT8xC5123)
  - 256 bytes of RAM
  - 512 bytes of internal XRAM

The CRAM/ROM configuration contain an ISP software (Bootloader) and a ROM monitor (Emulator) mainly dedicated for pre-production, code development and debug, or specific applications while the ROM configuration contains the final customer application.

ROM Configuration Register

The ROM Configuration Register is masked and is not accessible by the MCU. It contains fuse bits which enable the product configuration.

Table 113. ROM Configuration Register for AT8xC5122

7	6	5	4	3	2	1	0
	BLJRB				LB2	LB1	LB0

<table>
<tr><th>Bit Number</th><th>Bit Mnemonic</th><th colspan="5">Description</th></tr>
<tr><td>7</td><td>-</td><td colspan="5"><b>Reserved</b></td></tr>
<tr><td>6</td><td>BLJRB</td><td colspan="5"><b>Bootloader Jump Rom Bit</b>
Set to configure User Code in ROM with a reset@0000h
Clear to configure in CRAM/ROM (Bootloader mode with a reset@F800h)</td></tr>
<tr><td>5 - 3</td><td>-</td><td colspan="5"><b>Reserved</b></td></tr>
<tr><td rowspan="4">2 - 0</td><td rowspan="4">LB2:0</td><td colspan="5"><b>Program Lock Bits</b>
The program lock bits protects the on-chip program against software piracy.
The AT8xC5122 products are delivered with the lowest protection level as detailed in the following table:</td></tr>
<tr><th>LB2</th><th>LB1</th><th>LB0</th><th>Security Level</th><th>Protection Description</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>No program lock features enabled.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2</td><td>Read test mode function is disabled.
High pin count package
MOVC instruction executed from external program memory are disabled from fetching code bytes from internal memory.
$\overline{EA}$ is sampled and latched on reset.
External execution is enabled.
Checksum control is enabled.
Low pin count package
Checksum control is enabled.</td></tr>
</table>

Table 114. ROM Configuration Register for AT8xC5123

7	6	5	4	3	2	1	0
-	-	-	-	-	LB2	LB1	LB0

Bit Number	Bit Mnemonic	Description
7 - 3	-	**Reserved**

<table>
<tr><th>Bit
Number</th><th>Bit
Mnemonic</th><th colspan="5">Description</th></tr>
<tr><td rowspan="4">2 - 0</td><td rowspan="4">LB2:0</td><td colspan="5">**Program Lock Bits**
The program lock bits protects the on-chip program against software piracy.
The AT8xC5122 products are delivered with the lowest protection level as detailed in the following table:</td></tr>
<tr><th>LB2</th><th>LB1</th><th>LB0</th><th>Security
Level</th><th>Protection Description</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>No program lock features enabled.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2</td><td>Read test mode function is disabled.
High pin count package
MOVC instruction executed from external program memory are disabled from fetching code bytes from internal memory.
$\overline{EA}$ is sampled and latched on reset.
External execution is enabled.
Checksum control is enabled.
Low pin count package
Checksum control is enabled.</td></tr>
</table>

CRAM/ROM Configuration

The split of internal memory spaces depends on the product and is detailed below.

The instruction codes are fetched from external or internal program memory depending on the logic state of microcontroller's EA pin. Only valid for high pin count packages (VQFP64).

Case EA = 0

After the reset, the program counter is initialised to 0000h and the code instructions are fetched from external program memory.

Case EA = 1

The ROM contains the bootloader code. After reset sequence the program counter is initialized to F800h and the bootloader is run. The bootloader downloads the application into the CRAM. The sources to download the application from are in priority:

- an external 32K EEPROM attached to a synchronous serial interface
- a host attached to a USB interface
- a host attached to a RS232C interface

Note: Since the ROM is mapped in the upper 32K of program memory, the bootloader is not able to provide direct interrup services.

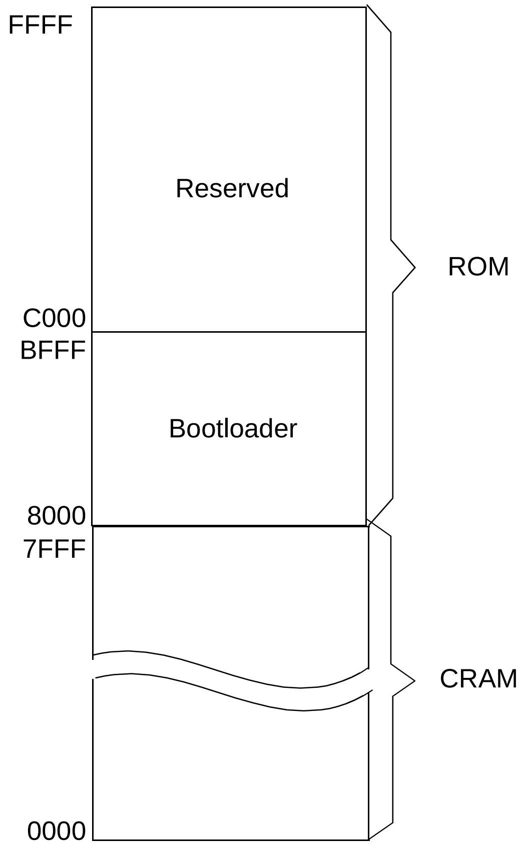
Rollover Function

Once the application is running in the internal lower 32K memory space, it can roll over in the external upper 32K memory space. A bit called ENBOOT and contained in AUXR1 register enables to select this function. ENBOOT bit is set to 1 by the reset function.

Mapping

The program memory space is described in Figure 90.

Figure 90. ROM contents (CRAM / ROM Configuration)



ROM Configuration

The product provides the following program and data memory spaces.

The instruction codes are fetched from external or internal program memory depending on the logic state of microcontroller's EA pin. Only valid for high pin count packages (VQFP64).

Case EA = 0

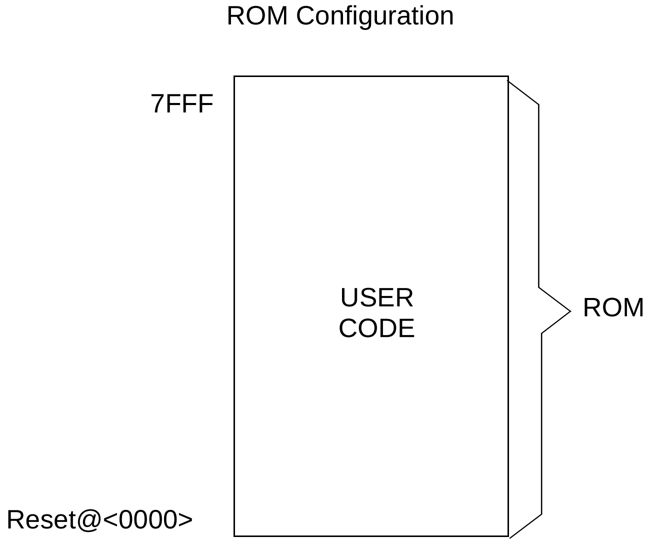
After the reset the program counter is initialised to 0000h and the code instructions are fetched from external program memory.

Case EA = 1

After a reset the program counter is initialized to 0000h and the customer application contained in internal ROM is executed from address 0000h to 7FFFh.

The program memory space is described in figure below:

Figure 91. ROM Contents (ROM Configuration)



Memory Mapping

In the products ROM versions, the following internal spaces are defined:

- RAM
- XRAM
- CRAM: 32K Bytes Program RAM Memory
- ROM

The specific accesses from/to these memories are:

- XRAM: if the bit RPS in RCON (described below) is reset, MOVX instructions address the XRAM space.
- CRAM: if the bit RPS in RCON is set, MOVX instructions address the CRAM space.

Table 115. RAM Configuration Register - RCON (D1h)

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

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	RPS	**CRAM Space Map Bit** Set to map the CRAM space during MOVX instructions Clear to map the Data space during MOVX. This bit has priority over the EXTRAM bit.
2-0	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.

Reset Value = XXXX 0XXXb

CRAM Version without Customer Mask

Two memory blocks are implemented:

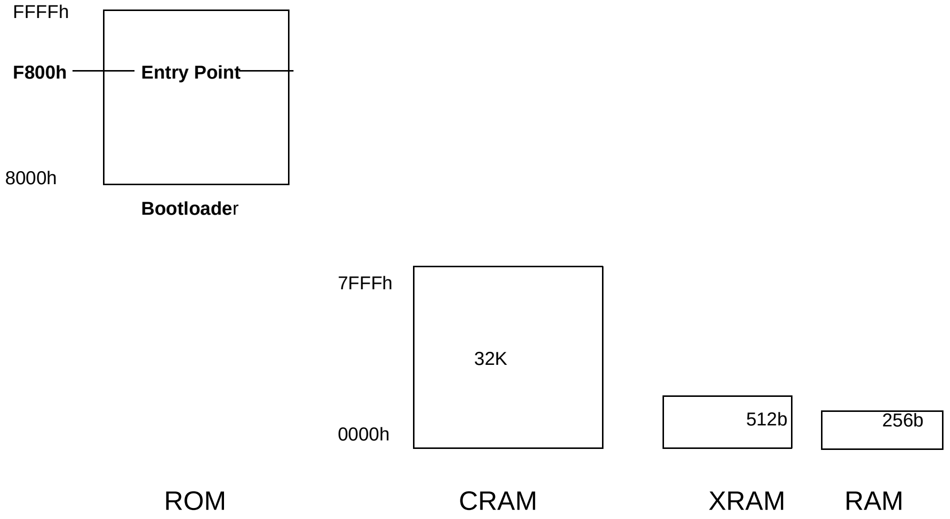
- The ROM memory contains the Bootloader program.
- The CRAM is the Application program memory.

After a Reset, the program is downloaded, as described above, from:

- either an external EEPROM, or
- from an host conected on RS232 serial link.

into the program CRAM memory of 32K bytes. Then the Program Counter is set at address 0000h of the CRAM space and the program is executed.

Figure 92. CRAM+ROM Mapping



CRAM Version with Customer Mask and ROM Version

In this version, the customer program is masked in 32K bytes ROM.

Two memory areas are implemented:

- The customer program is masked in ROM during the final production phase. The ROM Size will be determinated at mask generation process depending of the program size.
- In the CRAM+ROM product, the CRAM is not used.

The Bootloader Jump ROM Bit (BLJRB) is set to enable the user ROM program which is executed after reset.

Electrical Characteristics

Absolute Maximum Ratings

Ambiant Temperature Under Bias ....-25°C to 85°C

Storage Temperature .... -65°C to + 150°C

Voltage on V_{CC} to V_{SS} ....-0.5 V to + 6.0V

Voltage on Any Pin to V_{SS}........................-0.5 V to V_{CC} + 0.5 V

Power Dissipation TBD W

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 value is based on the maximum allowable die temperature and the thermal resistance of the package.

DC Parameters

T_A = -40 to +85°C; V_{SS} = 0 V, F= 0 to 16MHz

X2 Core

V_{CC} = 3.6V to 5.5V on -M Version

Table 116. Core DC Parameters (XTAL, $\overline{RST}$, P0, P2, P3, P4, P5, ALE, $\overline{PSEN}$, $\overline{EA}$)

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 XTAL1, $\overline{RST}$	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage, XTAL1, $\overline{RST}$	0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage: P0, ALE, PSEN			0.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage: P0, ALE, PSEN	0.9 V_{CC}			V	I_{OH} = -40 µA
V_{OL1}	Output Low Voltage: P2, P3, P4, P5, P1.2, P1.6, P1.7,			0.45	V	I_{OL} = 0.8 mA
V_{OH1}	Output High Voltage: P2, P3, P4, P5, P1.2, P1.6, P1.7	0.9 V_{CC}			V	I_{OH} = -10 µA
I_{IL}	Logical 0 Input Current ports 2 to 5 and P1.2, P1.6, P1.7, if Weak pull-up enabled			-50	µA	Vin = 0.45 V
I_{LI}	Input Leakage Current			±10	µA	0.45 V < V_{IN} < V_{CC}
I_{TL}	Logical 1 to O transistion Current, Port 51 configuration			–650	µA	V_{IN} = 2 V
R_{MEDIUM}	Medium Pullup Resistor		10		kΩ	
R_{WEAK}	Weak Pullup Resistor		100		kΩ	
CIO	Capacitance of I/O Buffer			10	pF	Fc = 1 MHz T_A = 25°C
CIO	Capacitance of I/O Buffer			10	pF	Fc = 1 MHz T_A = 25°C
DI_{CC}	Digital Supply Output Current	10			mA	C_L = 100 nF F= 16 MHz X1
DV_{CC}	Digital Supply Voltage	3	3.3	3.6	V	C_L = 100 nF

Table 116. Core DC Parameters (XTAL, $\overline{\text{RST}}$, P0, P2, P3, P4, P5, ALE, $\overline{\text{PSEN}}$, $\overline{\text{EA}}$) (Continued)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{PFDP}	Power fail high level threshold		2.8		V	
V_{PFDM}	Power fail low level threshold		2.6		V	
$t_{rise,}$ t_{fall}	V_{DD} rise and fall time	1µs		600	second	
R_{RST}	Internal reset pull-up resistor		15		kΩ	
C	External reset capacitor	150 nF	1 µF			
Trst	Reset duration	10	70		ms	Trst=0.7*R*C

Operating I_{CC} Test Condition

Figure 93. I_{CC} Test Condition, Idle Mode

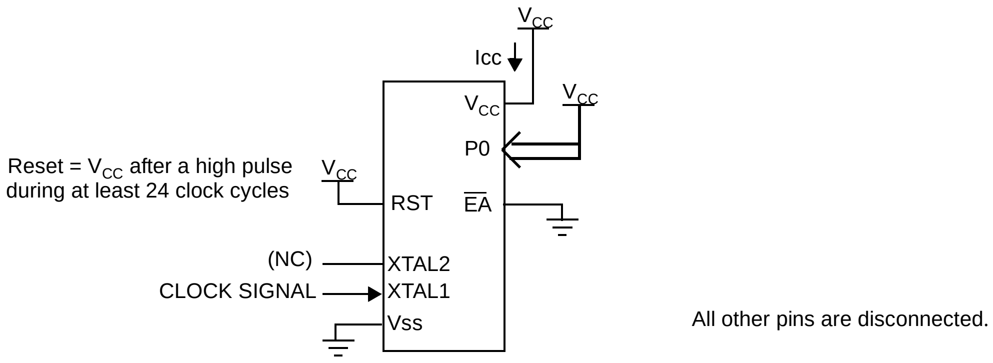

VCC
Icc
VCC
VCC
P0
Reset = VCC after a high pulse
during at least 24 clock cycles
VCC
RST
EA
(NC)
XTAL2
CLOCK SIGNAL
XTAL1
Vss
All other pins are disconnected.

Figure 94. I_{CC} Test Condition, Power-down Mode

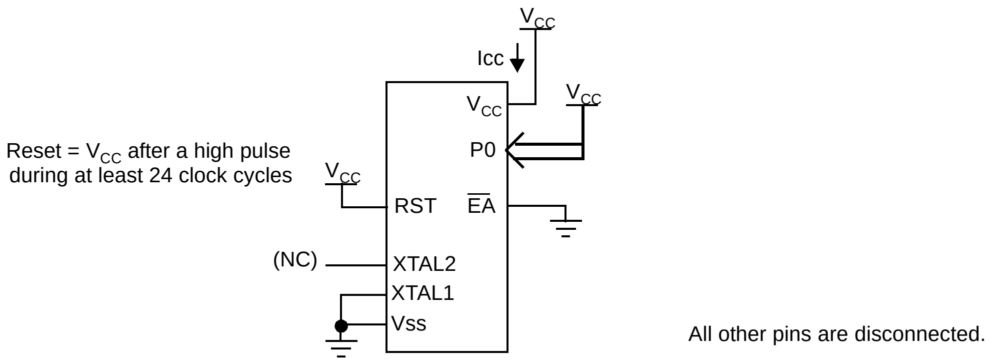
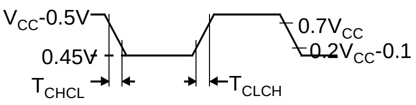

VCC
Icc
VCC
VCC
P0
Reset = VCC after a high pulse
during at least 24 clock cycles
VCC
RST
EA
(NC)
XTAL2
XTAL1
Vss
All other pins are disconnected.

Figure 95. Clock Signal Waveform for I_{CC} Tests in Idle Mode



$T_{CLCH} = T_{CHCL} = 5$ ns.

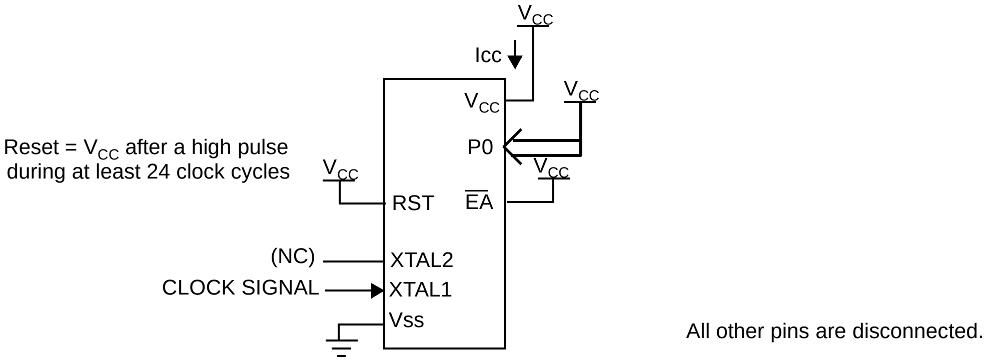

V_{CC}
Icc
V_{CC}
V_{CC}
P0
Reset = V_{CC} after a high pulse during at least 24 clock cycles
V_{CC}
V_{CC}
RST
$\overline{EA}$
(NC)
XTAL2
CLOCK SIGNAL
XTAL1
Vss
All other pins are disconnected.

Figure 96. I_{CC} Test Condition, Active Mode

LED's

Table 117. LED Outputs DC Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
I_{OL}	Output Low Current, P3.6 and P3.7 LED modes	1 2 5	2 4 10	4 8 20	mA mA mA	2 mA configuration 4 mA configuration 10 mA configuration

Note: 1. (T_A = -20°C to +50°C, V_{CC} - V_{OL} = 2 V ± 20%)

Smart Card Interface

Table 118. Smart Card 5V Interface DC Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
CI_{CC}	Card Supply Current	60		121 105 102	mA	VCC = 5.5V VCC = 4V VCC = 2.85V
CV_{CC}	Card Supply Voltage	4.6		5.4	V	CIcc = 60 mA
	Ripple on Vcard			200	mV	0 < CIcc < 60 mA
CV_{CC}	Spikes on Vcard	4.6		5.4	V	Maxi. charge 20 nA Max. duration 400 ns Max. variation CIcc 100 mA
T_{VHLI}	Vcard to 0			750	µs	CIcc = 0 Vcard = 5V to 0.4V

Table 119. Smart Card 3V Interface DC Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
CI_{CC}	Card Supply Current	60		110 89 110	mA	VCC = 5.5V VCC = 4V VCC = 2.85V
CV_{CC}	Card Supply Voltage	2.76		3.24	V	CIcc = 60 mA
	Ripple on Vcard			200	mV	0 < CIcc < 60 mA
CV_{CC}	Spikes on Vcard	2.76		3.24	V	Maxi. charge 10nA.s Max. duration 400 ns Max. variation CIcc 50mA

Table 119. Smart Card 3V Interface DC Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
T_{VHLI}	CVcc to 0			750	µs	Icard=0 Vcard = 5V to 0.4V

Table 120. Smart Card 1.8V Interface DC parameters

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
CI_{CC}	Card Supply Current	20		109 100 82	mA	VCC = 5.5V VCC = 4V VCC = 2.85V
CV_{CC}	Card Supply Voltage	1.68		1.92	V	CIcc = 20 mA
CV_{CC}	Spikes on Vcard	1.68		1.92	V	
T_{VHLI}	CVcc to 0			750	µs	CIcc = 0 CV_{CC} = 5V to 0.4V

Notes: 1. Test conditions, Capacitor 10 µF, Inductance 10 µH.
2. Ceramic X7R, SMD type capacitor with minimum ESR or 250 mΩ is mandatory

Table 121. Smart Card Clock DC parameters (Port P1.4)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{OL}	Output Low Voltage	0[1] 0[1]		$0.2xV_{CC}$ 0.4	V	I_{OL} = 20 µA (1.8V, 3V) I_{OL}= 50 µA (5V)
I_{OL}	Output Low Current			15	mA	
V_{OH}	Output High Voltage	0.7 CV_{CC} 0.7 CV_{CC} 0.7 CV_{CC} CV_{CC} - 0.5		CV_{CC} CV_{CC} CV_{CC} CV_{CC}	V V V V	I_{OH} = 20 µA (1.8V) I_{OH} = 20 µA (3V) I_{OH} = 20 µA (5V) I_{OH} = 50 µA (5V)
I_{OH}	Output High Current			15	mA	
t_R t_F	Rise and Fall delays			16 22.5 50	ns	C_{IN}=30pF (5V) C_{IN}=30pF (3V) C_{IN}=30pF (1.8V)
	Voltage Stability	-0.25 CV_{CC}-0.5		0.4 CV_{CC} CV_{CC} + 0.25	V	Low level High level
	Frequency variation			1%		
	Cycle ratio	45%		55%		

Note: 1. The voltage on CLK should remain between -0.3V and V_{CC}+0.3V during dynamic operation

Table 122. Smart Card I/O DC Parameters (P1.0)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	0[1] 0[1]		0.5 0.15 CV_{CC}	V	I_{IL}= 500 µA I_{IL} = 20 µA
I_{IL}	Input Low Current			500	µA	

Table 122. Smart Card I/O DC Parameters (P1.0) (Continued)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}	Input High Voltage	0.7 CV_{CC}		CV_{CC}	V	I_{IH} = -20 µA
I_{IH}	Input High Current			-20 / +20	µA	
V_{OL}	Output Low Voltage	0(1)		0.4 0.4 0.3	V	I_{OL} = 1mA (5V) I_{OL} = 1mA (3V) I_{OL} = 1mA (1.8V)
I_{OL}	Output Low Current			15	mA	
V_{OH}	Output High Voltage	0.8 CV_{CC} 0.7 CV_{CC}		CV_{CC} (1)	V	I_{OH} = 20 µA (5V) I_{OH} = 20 µA (3V, 1.8V)
I_{OH}	Output High Current			15	mA	
	Voltage Stability	-0.25 0.8 CV_{CC}		0.4 CV_{CC} + 0.25	V	Low level High level
t_R t_F	Rise and Fall delays			0.8	µs	C_{IN}=30pF

Note: 1. The voltage on RST should remain between -0.3V and V_{CC}+0.3V during dynamic operation.

Table 123. Smart Card RST, CC4, CC8, DC Parameters (Port P1.5, P1.3, P1.1)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{OL}	Output Low Voltage	0(1) 0(1)		0.12 x V_{CC} 0.4	V	I_{OL} = 20 µA I_{OL}= 50 µA
I_{OL}	Output Low Current			15	mA	
V_{OH}	Output High Voltage	CV_{CC} - 0.5 0.8 x V_{CC}		CV_{CC} CV_{CC} (1)	V	I_{OH} = 50 µA I_{OH} = 20 µA
I_{OH}	Output High Current			15	mA	
t_R t_F	Rise and Fall delays			0.8	µs	C_{IN}=30 pF
	Voltage Stability	-0.25 CV_{CC}-0.5		0.4 x CV_{CC} CV_{CC} + 0.25		Low level High level

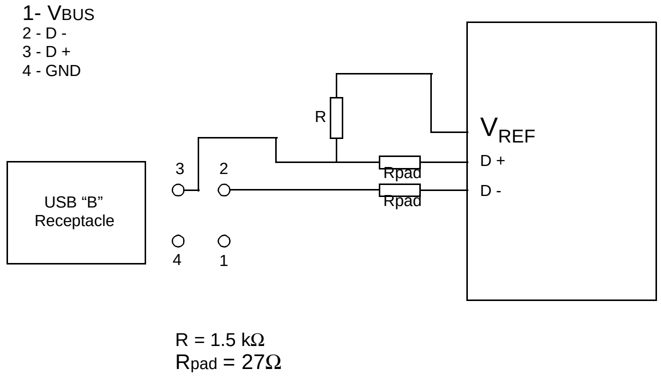
Note: 1. The voltage on RST should remain between -0.3V and V_{CC}+0.3V during dynamic operation.

Table 124. Card Presence DC Parameters (P1.2)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
I_{OL1}	CPRES weak pull-up output current	3	10	25	µA	P1.2=1, short to VSS Pull-up enabled

USB Interface

Figure 97. USB Interface



Symbol	Parameter	Min	Typ[5]	Max	Unit
V_{REF}	USB Reference Voltage	3.0		3.6	V
V_{IH}	Input High Voltage for D+ and D- (driven)	2.0			V
V_{IHZ}	Input High Voltage for D+ and D- (floating)	2.7		3.6	V
V_{IL}	Input Low Voltage for D+ and D-			0.8	V
V_{OH}	Output High Voltage for D+ and D-	2.8		3.6	V
V_{OL}	Output Low Voltage for D+ and D-	0.0		0.3	V

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 $\overline{\text{PSEN}}$ Low.

T_A = -40°C to +85°C; V_{SS} = 0V; V_{CC} = 5V ±10%; F = 0 to 40 MHz.

(Load Capacitance for port 0, ALE and PSEN = 60 pF; Load Capacitance for all other outputs = 60 pF.)

Table 125, Table 128 and Table 131 give the description of each AC symbols.

Table 126, Table 130 and Table 132 give for each range the AC parameter.

Table 127, Table 130 and Table 133 give the frequency derating formula of the AC parameter for each speed range description. To calculate each AC symbols. take the x value and use this value in the formula.

Example: T_{LLIV} and 20 MHz, Standard clock.
x = 30 ns
T = 50 ns
T_{CCIV} = 4T - x = 170 ns

External Program Memory Characteristics

Table 125. 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

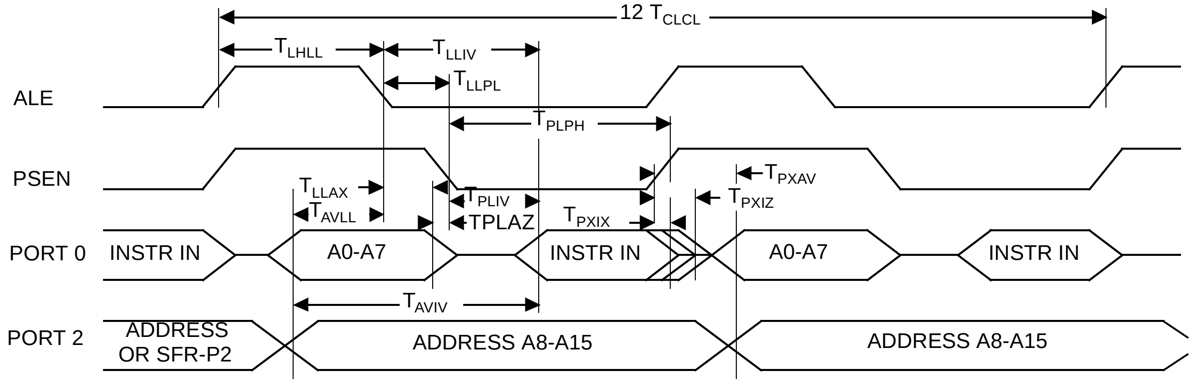
Table 126. AC Parameters for a Fix Clock (F = 40 MHz)

Symbol	Min	Max	Units
T	TBD		ns
T_{LHLL}	TBD		ns
T_{AVLL}	TBD		ns
T_{LLAX}	TBD		ns
T_{LLIV}		TBD	ns
T_{LLPL}	TBD		ns
T_{PLPH}	TBD		ns
T_{PLIV}		TBD	ns
T_{PXIX}	TBD		ns
T_{PXIZ}		TBD	ns
T_{AVIV}		TBD	ns
T_{PLAZ}		TBD	ns

Table 127. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X parameter	Units
T_{LHLL}	Min	2 T - x	T - x	10	ns
T_{AVLL}	Min	T - x	0.5 T - x	15	ns
T_{LLAX}	Min	T - x	0.5 T - x	15	ns
T_{LLIV}	Max	4 T - x	2 T - x	30	ns
T_{LLPL}	Min	T - x	0.5 T - x	10	ns
T_{PLPH}	Min	3 T - x	1.5 T - x	20	ns
T_{PLIV}	Max	3 T - x	1.5 T - x	40	ns
T_{PXIX}	Min	x	x	0	ns
T_{PXIZ}	Max	T - x	0.5 T - x	7	ns
T_{AVIV}	Max	5 T - x	2.5 T - x	40	ns
T_{PLAZ}	Max	x	x	10	ns

External Program Memory Read Cycle



External Data Memory Characteristics

Table 128. 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

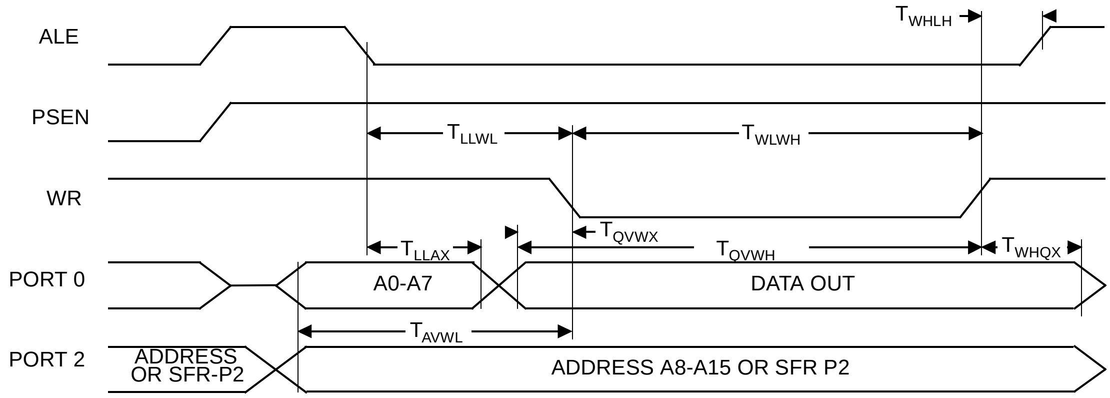
Table 129. AC Parameters for a Variable Clock (F = 40 MHz)

Symbol	Min	Max	Units
T_{RLRH}	TBD		ns
T_{WLWH}	TBD		ns
T_{RLDV}		TBD	ns
T_{RHDX}	TBD		ns
T_{RHDZ}		TBD	ns
T_{LLDV}		TBD	ns
T_{AVDV}		TBD	ns
T_{LLWL}	TBD	TBD	ns
T_{AVWL}	TBD		ns
T_{QVWX}	TBD		ns
T_{QVWH}	TBD		ns
T_{WHQX}	TBD		ns
T_{RLAZ}		TBD	ns
T_{WHLH}	TBD	TBD	ns

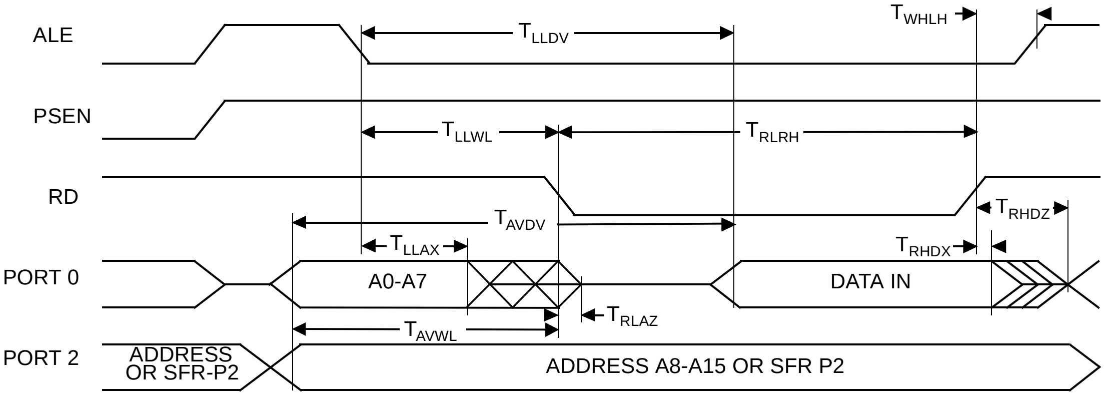
Table 130. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X parameter	Units
T_{RLRH}	Min	6 T - x	3 T - x	20	ns
T_{WLWH}	Min	6 T - x	3 T - x	20	ns
T_{RLDV}	Max	5 T - x	2.5 T - x	25	ns
T_{RHDX}	Min	x	x	0	ns
T_{RHDZ}	Max	2 T - x	T - x	20	ns
T_{LLDV}	Max	8 T - x	4T -x	40	ns
T_{AVDV}	Max	9 T - x	4.5 T - x	60	ns
T_{LLWL}	Min	3 T - x	1.5 T - x	25	ns
T_{LLWL}	Max	3 T + x	1.5 T + x	25	ns
T_{AVWL}	Min	4 T - x	2 T - x	25	ns
T_{QVWX}	Min	T - x	0.5 T - x	15	ns
T_{QVWH}	Min	7 T - x	3.5 T - x	25	ns
T_{WHQX}	Min	T - x	0.5 T - x	10	ns
T_{RLAZ}	Max	x	x	0	ns
T_{WHLH}	Min	T - x	0.5 T - x	15	ns
T_{WHLH}	Max	T + x	0.5 T + x	15	ns

External Data Memory Write Cycle



External Data Memory Read Cycle



Serial Port Timing - Shift Register Mode

Table 131. Symbol Description (F = 40 MHz)

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

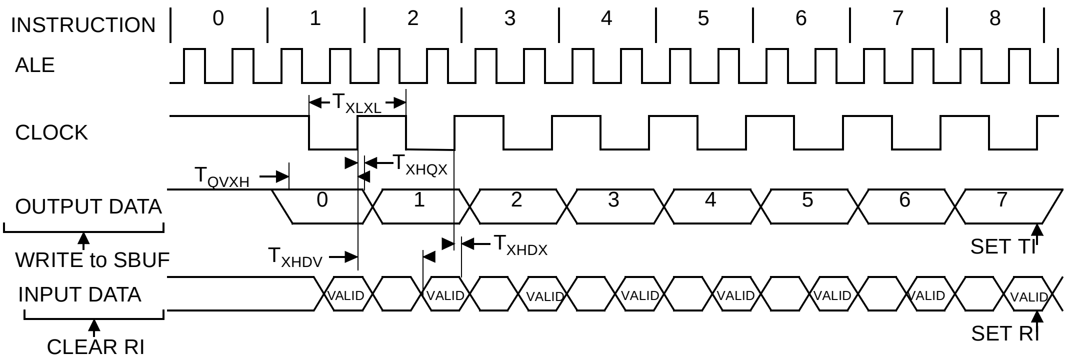
Table 132. AC Parameters for a Fix Clock (F = 40 MHz)

Symbol	Min	Max	Units
T_{XLXL}	TBD		ns
T_{QVHX}	TBD		ns
T_{XHQX}	TBD		ns
T_{XHDX}	TBD		ns
T_{XHDV}		TBD	ns

Table 133. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X parameter for -M range	Units
T_{XLXL}	Min	12 T	6 T		ns
T_{QVHX}	Min	10 T - x	5 T - x	50	ns
T_{XHQX}	Min	2 T - x	T - x	20	ns
T_{XHDX}	Min	x	x	0	ns
T_{XHDV}	Max	10 T - x	5 T- x	133	ns

Shift Register Timing Waveform

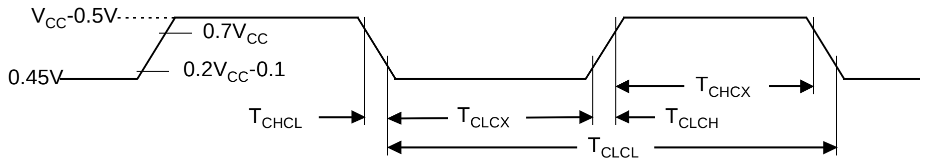


External Clock Drive Characteristics (XTAL1)

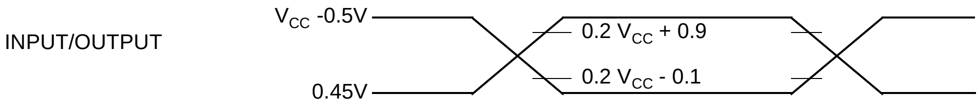
Table 134. AC Parameters

Symbol	Parameter	Min	Max	Units
T_{CLCL}	Oscillator Period	25		ns
T_{CHCX}	High Time	5		ns
T_{CLCX}	Low Time	5		ns
T_{CLCH}	Rise Time		5	ns
T_{CHCL}	Fall Time		5	ns
T_{CHCX}/T_{CLCX}	Cyclic ratio in X2 mode	40	60	%

External Clock Drive Waveforms

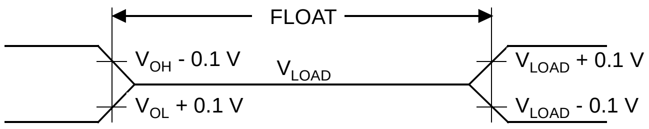


AC Testing Input/Output Waveforms



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”.

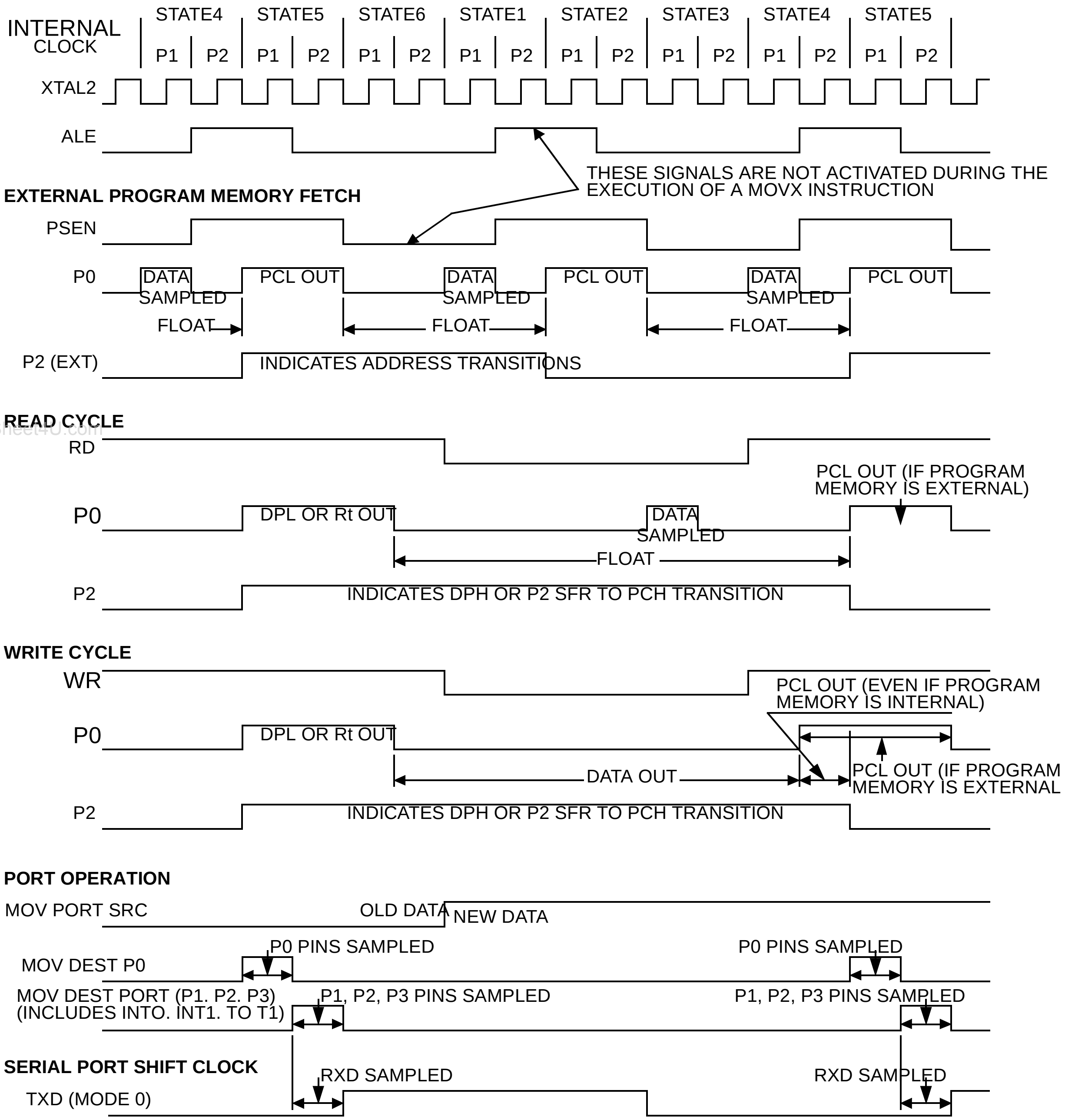
Float Waveforms



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.


INTERNAL CLOCK
STATE4
STATE5
STATE6
STATE1
STATE2
STATE3
STATE4
STATE5
P1
P2
XTAL2
ALE
THESE SIGNALS ARE NOT ACTIVATED DURING THE EXECUTION OF A MOVX INSTRUCTION
EXTERNAL PROGRAM MEMORY FETCH
PSEN
P0
DATA SAMPLED
PCL OUT
FLOAT
P2 (EXT)
INDICATES ADDRESS TRANSITIONS
READ CYCLE
RD
P0
DPL OR Rt OUT
DATA SAMPLED
FLOAT
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
P2
INDICATES DPH OR P2 SFR TO PCH TRANSITION
WRITE CYCLE
WR
PCL OUT (EVEN IF PROGRAM MEMORY IS INTERNAL)
P0
DPL OR Rt OUT
DATA OUT
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL
P2
INDICATES DPH OR P2 SFR TO PCH TRANSITION
PORT OPERATION
MOV PORT SRC
OLD DATA
NEW DATA
MOV DEST P0
P0 PINS SAMPLED
P0 PINS SAMPLED
MOV DEST PORT (P1. P2. P3)
(INCLUDES INTO. INT1. TO T1)
P1, P2, P3 PINS SAMPLED
P1, P2, P3 PINS SAMPLED
SERIAL PORT SHIFT CLOCK
TXD (MODE 0)
RXD SAMPLED
RXD SAMPLED

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) $\overline{RD}$ and $\overline{WR}$ propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

USB Interface

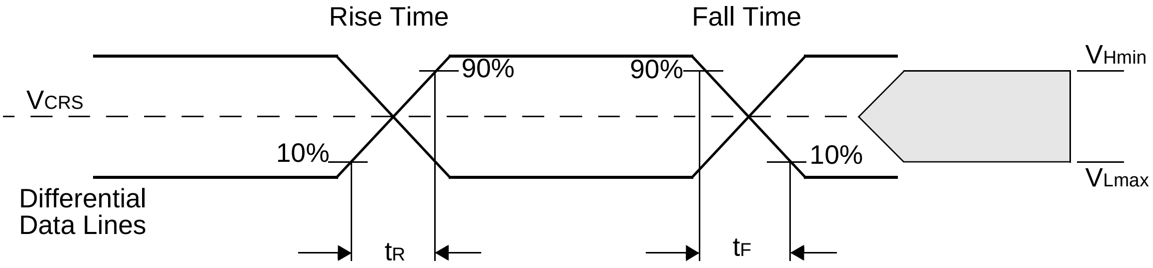
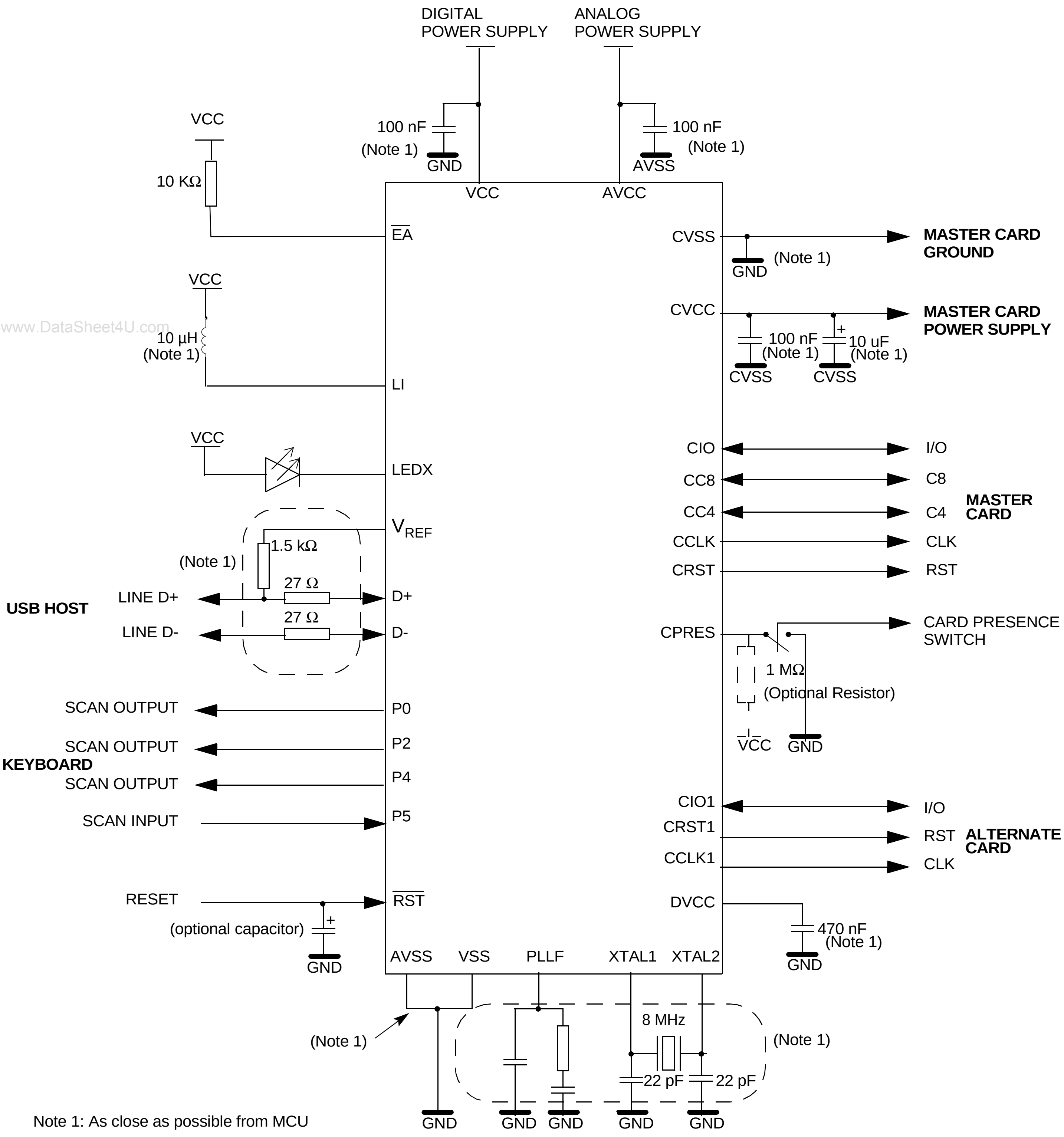


Table 135. USB AC Parameters

Symbol	Parameter	Min	Typ[5]	Max	Unit
t_R	Rise Time	4		20	ns
t_F	Fall Time	4		20	ns
t_{FDRATE}	Full-speed Data Rate	11.9700		12.0300	Mb/s
V_{CRS}	Crossover Voltage	1.3		2.0	V
t_{DJ1}	Source Jitter Total to next transaction	-3.5		3.5	ns
t_{DJ2}	Source Jitter Total for paired transactions	-4		4	ns
t_{JR1}	Receiver Jitter to next transaction	-18.5		18.5	ns
t_{JR2}	Receiver Jitter for paired transactions	-9		9	ns

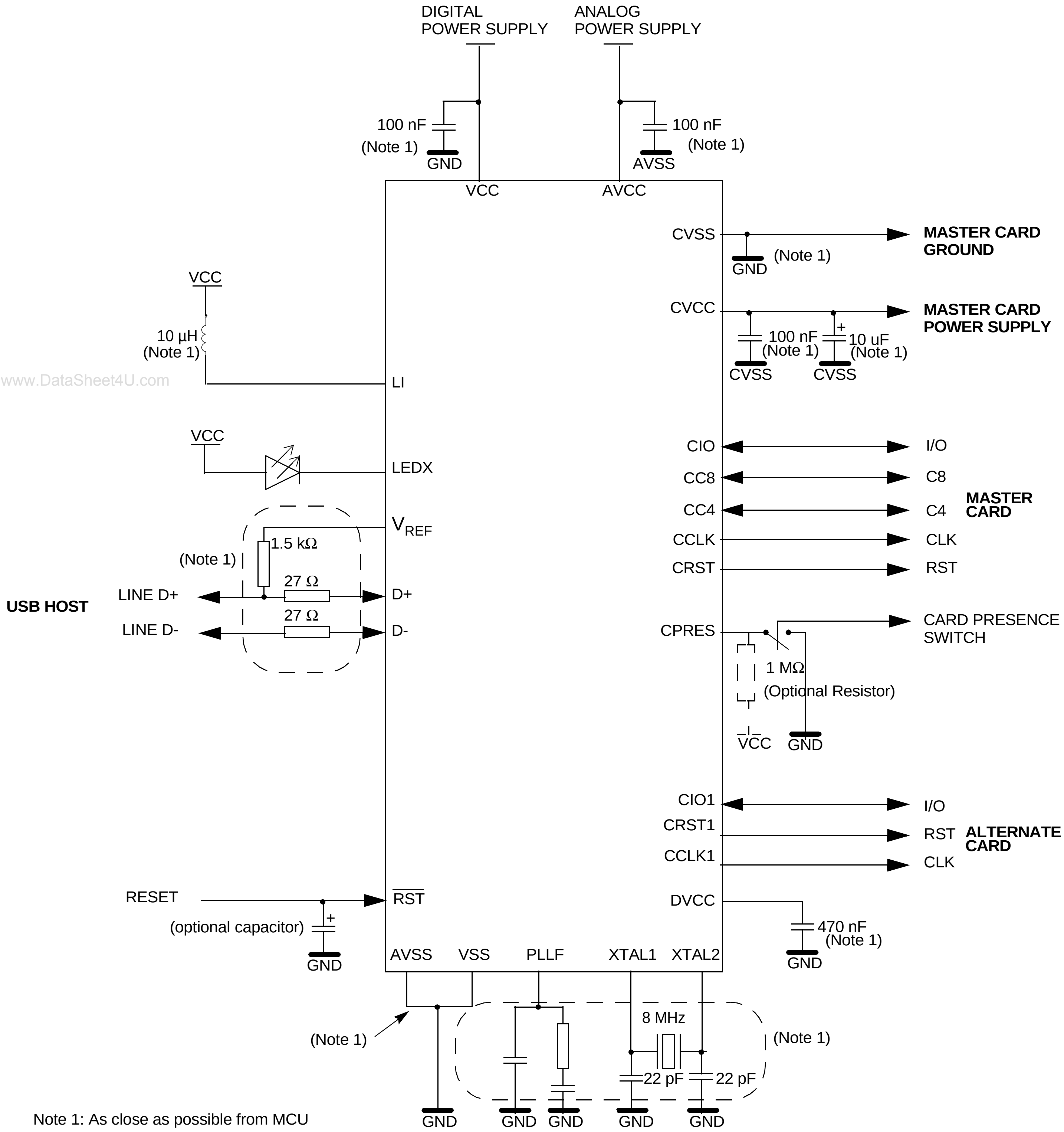
Typical Application

Figure 98. Typical Smart Card Reader and Keyboard Application Schematic



Note 1: As close as possible from MCU

Figure 99. Typical Smart Card Reader Application Schematic



Note 1: As close as possible from MCU

Ordering Information

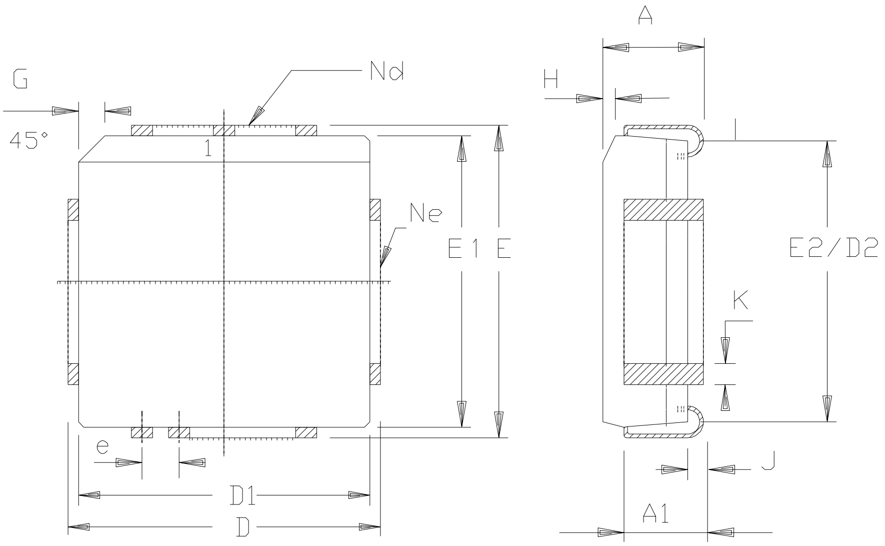
Part Number	Memory Size (bytes)	Supply Voltage (V)	Temperature Range	Max Frequency (MHz)	Package	Packing
AT83C5122xxx-RDTIM	32K ROM	3.6 - 5.5	Industrial	32	VQFP64	Tray
AT83C5122xxx-RDRIM	32K ROM	3.6 - 5.5	Industrial	32	VQFP64	Tape & Reel
AT83C5122xxx-SISIM	32K ROM	3.6 - 5.5	Industrial	32	PLCC28	Stick
AT83C5122xxx-SIRIM	32K ROM	3.6 - 5.5	Industrial	32	PLCC28	Tape & Reel
AT85EC5122-RDVIM	32K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	VQFP64	Tray & Dry pack
AT85EC5122-RDFIM	32K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	VQFP64	Tray & Reel & Dry pack
AT85EC5122-SIUIM(1)	32K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	PLCC28	Stick & Dry pack
AT85EC5122-SIXIM(1)	32K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	PLCC28	Tray & Reel & Dry pack
AT85C5122xxx-RDTIM	RAM	3.6 - 5.5	Industrial	32	VQFP64	Tray
AT85C5122xxx-RDRIM	RAM	3.6 - 5.5	Industrial	32	VQFP64	Tape & Reel
AT85C5122xxx-SISIM	RAM	3.6 - 5.5	Industrial	32	PLCC28	Stick
AT85C5122xxx-SIRIM	RAM	3.6 - 5.5	Industrial	32	PLCC28	Tape & Reel
AT89C5122-RDTIM(1)	32K Flash RAM	3.6 - 5.5	Industrial	32	VQFP64	Tray
AT89C5122-RDRIM(1)	32K Flash RAM	3.6 - 5.5	Industrial	32	VQFP64	Tape & Reel
AT89C5122-SISIM(1)	32K Flash RAM	3.6 - 5.5	Industrial	32	PLCC28	Stick
AT89C5122-SIRIM(1)	32K Flash RAM	3.6 - 5.5	Industrial	32	PLCC28	Tape & Reel
AT83C5123xxx-RATIM	30K ROM	3.6 - 5.5	Industrial	32	LQFP32	Tray
AT83C5123xxx-RARIM	30K ROM	3.6 - 5.5	Industrial	32	LQFP32	Tape & Reel
AT83C5123xxx-SISIM	30K ROM	3.6 - 5.5	Industrial	32	PLCC28	Stick
AT83C5123xxx-SIRIM	30K ROM	3.6 - 5.5	Industrial	32	PLCC28	Tape & Reel
AT83EC5123xxx-RAVIM	30K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	LQFP32	Tray & Dry pack
AT83EC5123xxx-RAFIM	30K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	LQFP32	Tray & Reel & Dry pack
AT83EC5123xxx-SIUIM(1)	30K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	PLCC28	Stick & Dry pack

Part Number	Memory Size (bytes)	Supply Voltage (V)	Temperature Range	Max Frequency (MHz)	Package	Packing
AT83EC5123xxx-SIXIM[1]	30K ROM + 512 Bytes EEPROM	3.6 - 5.5	Industrial	32	PLCC28	Tray & Reel & Dry pack

Note: 1. Check avaibility with sales office

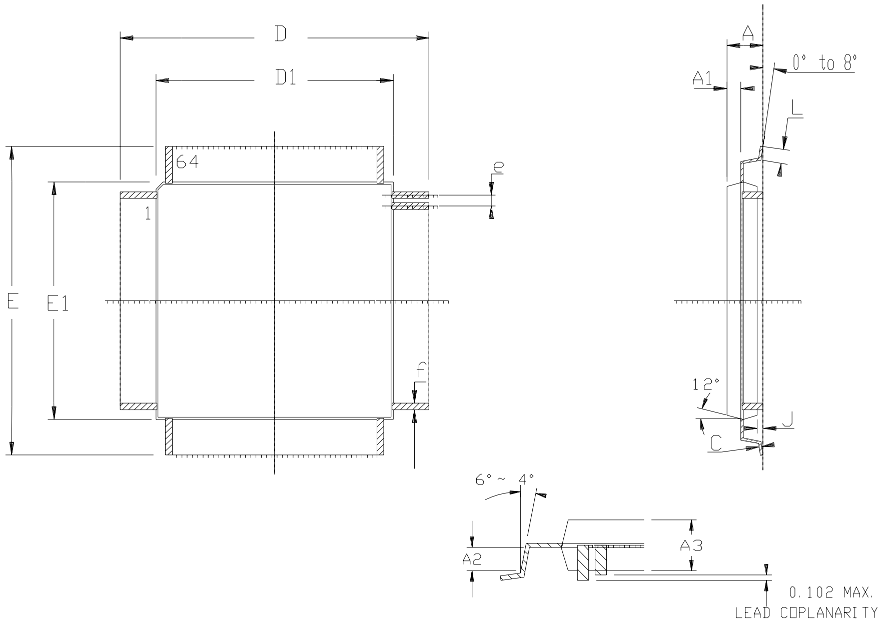
Packaging Information

PLCC28



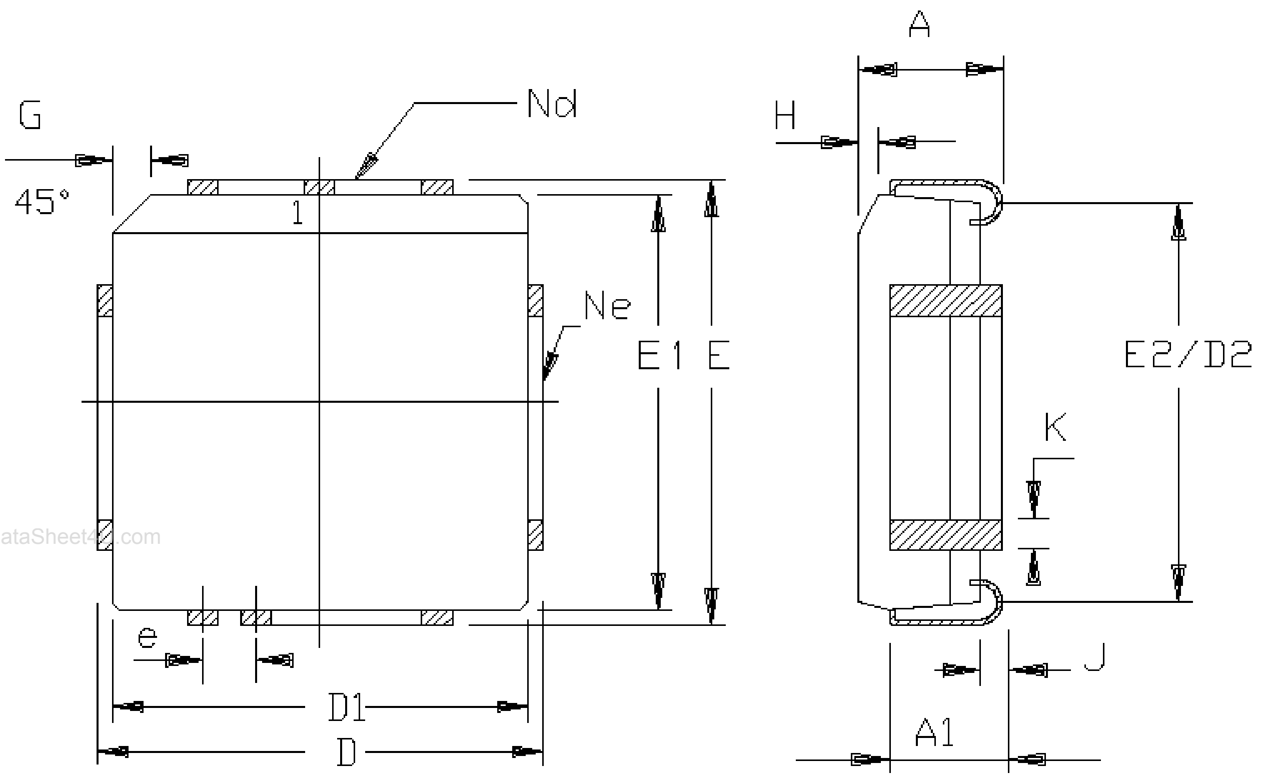
	MM		INCH	
A	4.20	4.57	.165	.180
A1	2.29	3.04	.090	.120
D	12.32	12.57	.485	.495
D1	11.43	11.58	.450	.456
D2	9.91	10.92	.390	.430
E	12.32	12.57	.485	.495
E1	11.43	11.58	.450	.456
E2	9.91	10.92	.390	.430
e	1.27	BSC	.050	BSC
G	1.07	1.22	.042	.048
H	1.07	1.42	.042	.056
J	0.51	-	.020	-
K	0.33	0.53	.013	.021
Nd	7		7	
Ne	7		7	
PKG STD	00			

VQFP64



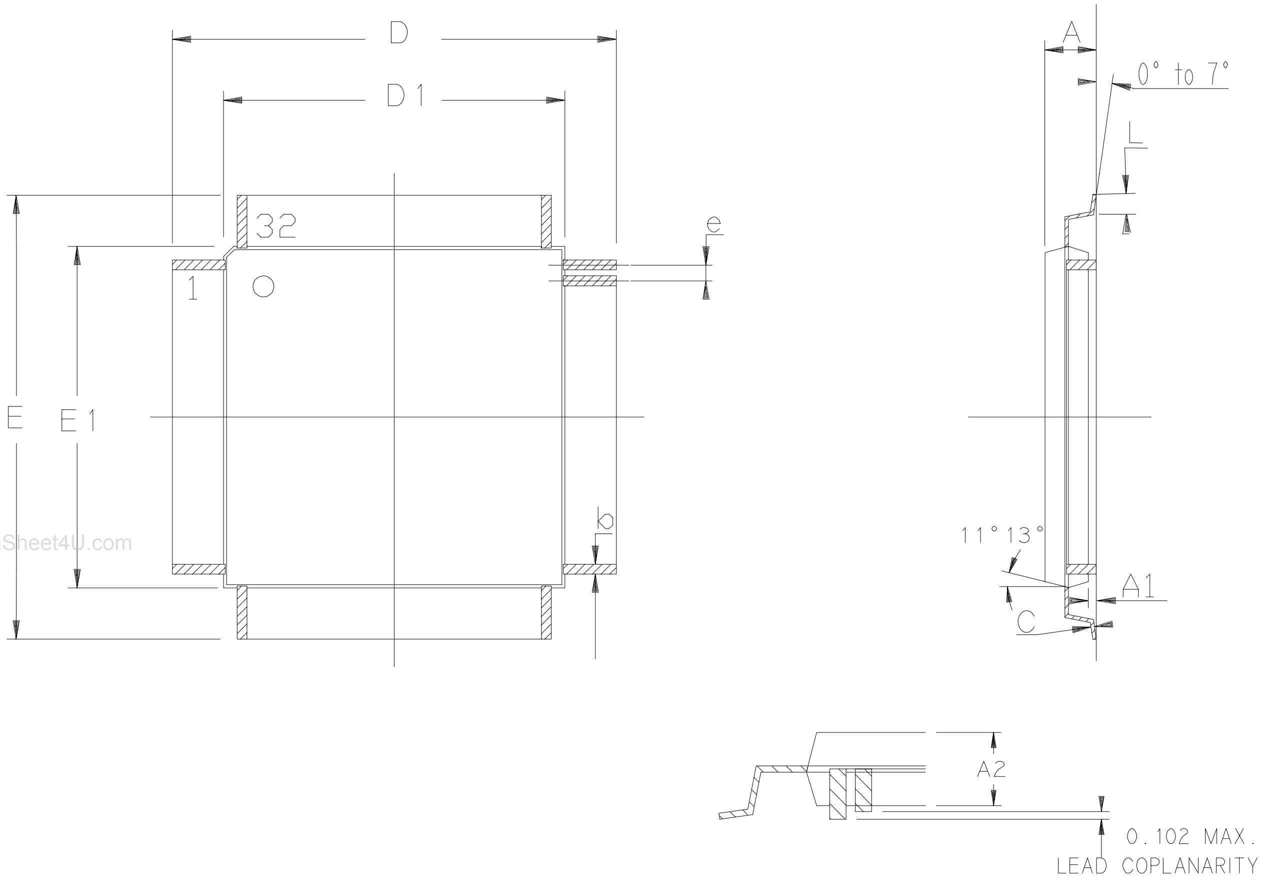
	MM		INCH	
	Min	Max	Min	Max
A	-	1.60	-	.063
A1	0.64 REF		.025 REF	
A2	0.64 REF		.025 REF	
A3	1.35	1.45	.053	.057
D	11.75	12.25	.463	.483
D1	9.90	10.10	.390	.398
E	11.75	12.25	.463	.483
E1	9.90	10.10	.390	.398
J	0.05	-	.002	-
L	0.45	0.75	.018	.030
e	0.50 BSC		.0197 BSC	
f	0.25 BSC		.010 BSC	

PLCC68



	MM		INCH	
A	4.20	5.08	.165	.200
A1	2.29	3.30	.090	.130
D	25.02	25.27	.985	.995
D1	24.13	24.33	.950	.958
D2	22.61	23.62	.890	.930
E	25.02	25.27	.985	.995
E1	24.13	24.33	.950	.958
E2	22.61	23.62	.890	.930
e	1.27	BSC	.050	BSC
G	1.07	1.22	.042	.048
H	1.07	1.42	.042	.056
J	0.51	-	.020	-
K	0.33	0.53	.013	.021
Nd	17		17	
Ne	17		17	
PKG STD	00			

LQFP32



	MM		INCH	
	Min	Max	Min	Max
A	-	1.60	-	.063
A1	0.05	0.15	.002	.006
A2	1.35	1.45	.053	.057
C	0.09	0.20	.004	.008
D	9.00 BSC		.354 BSC	
D1	7.00 BSC		.276 BSC	
E	9.00 BSC		.354 BSC	
E1	7.00 BSC		.276 BSC	
L	0.45	0.75	.018	.030
e	0.80 BSC		.0315 BSC	
b	0.30	0.45	.012	.018

Datasheet Change Log

Changes from 4202A to 4122B

1. Product AT8xEC5122 added.
2. Products AT83C5123 and AT83EC5123 added.

Table of Contents

Atmel Corporation

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

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

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

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

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

e-mail

literature@atmel.com

Web Site

http://www.atmel.com

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

♻ Printed on recycled paper.