Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

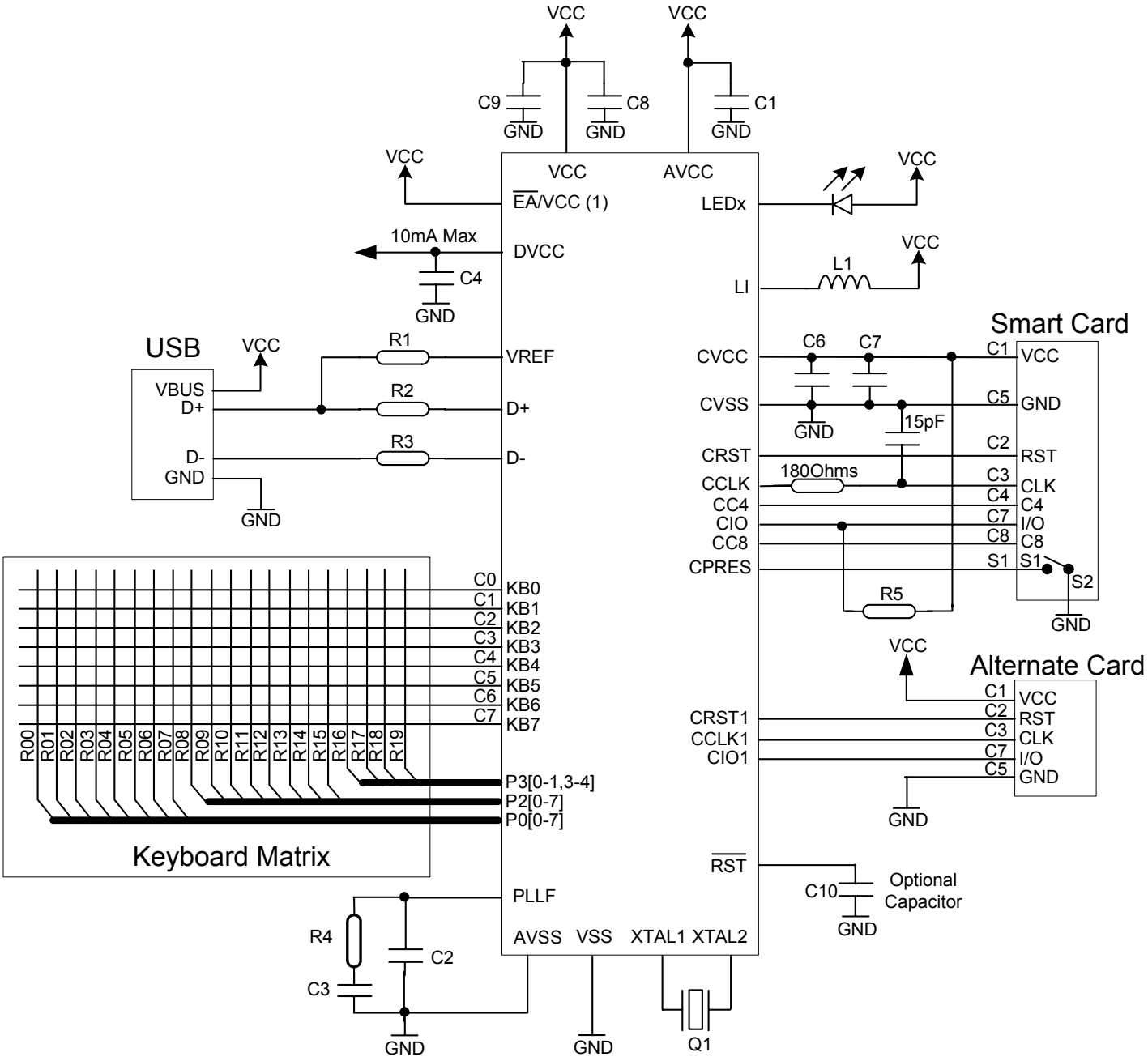
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	80C51
Core Size	8-Bit
Speed	48MHz
Connectivity	SmartCard, SPI, UART/USART, USB
Peripherals	LED, POR, WDT
Number of I/O	46
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFN
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c5122d-z1tum

Pin Description

Table 2. Pin Description

Port	VQFP64	VQFP32	PLCC68	PLCC28	QFN64	QFN32	Internal Power Supply	ESD	I/O	Reset Level	Alt	Reset Config	Conf 1	Conf 2	Conf 3	Led
P0.0	30	-	41	-	30	-	VCC	2KV	I/O	Float	AD0	P0		KB_OUT	Push-pull	
P0.1	29	-	40	-	29	-	VCC	2KV	I/O	Float	AD1	P0		KB_OUT	Push-pull	
P0.2	28	-	39	-	28	-	VCC	2KV	I/O	Float	AD2	P0		KB_OUT	Push-pull	
P0.3	27	-	38	-	27	-	VCC	2KV	I/O	Float	AD3	P0		KB_OUT	Push-pull	
P0.4	25	-	36	-	25	-	VCC	2KV	I/O	Float	AD4	P0		KB_OUT	Push-pull	
P0.5	24	-	35	-	24	-	VCC	2KV	I/O	Float	AD5	P0		KB_OUT	Push-pull	
P0.6	23	-	34	-	23	-	VCC	2KV	I/O	Float	AD6	P0		KB_OUT	Push-pull	
P0.7	22	-	33	-	22	-	VCC	2KV	I/O	Float	AD7	P0		KB_OUT	Push-pull	
CIO	64	32	9	4	64	32	CVCC	6KV	I/O	0		Port51	CVCC inactive at reset. ESD tested with a 10µF on CVCC An external pull-up of 10K is recommended to support ICC's with too weak internal pull-ups.			
CC4	3	3	12	7	3	3	CVCC	6KV	I/O	0		Port51	CVCC inactive at reset ESD tested with a 10µF on CVCC			
P1.2	2	2	11	6	2	2	VCC	2KV	I/O	1	CPRES	Port51	Weak & medium pull-up can be disconnected			
CC4	9	5	18	9	9	5	CVCC	6KV	I/O	0		Port51	CVCC inactive at reset ESD tested with a 10µF on CVCC			
CCLK	12	6	21	10	12	6	CVCC	6KV	O	0		Push-pull	CVCC inactive at reset ESD tested with a 10µF on CVCC			
CRST	6	4	15	8	6	4	CVCC	6KV	O	0		Push-pull	CVCC inactive at reset ESD tested with a 10µF on CVCC			
P1.6	47	23	58	-	47	23	VCC	2KV	I/O	1	SS	Port51				
P1.7	62	31	7	-	62	31	VCC	2KV	I/O	1	CCLK1	Port51				
P2.0	58	-	3	-	58	-	VCC	2KV	I/O	1	A8	Port51	Push-pull	KB_OUT	Input WPU	
P2.1	57	-	2	-	57	-	VCC	2KV	I/O	1	A9	Port51	Push-pull	KB_OUT	Input WPU	
P2.2	56	-	1	-	56	-	VCC	2KV	I/O	1	A10	Port51	Push-pull	KB_OUT	Input WPU	
P2.3	52	-	65	-	52	-	VCC	2KV	I/O	1	A11	Port51	Push-pull	KB_OUT	Input WPU	
P2.4	51	-	64	-	51	-	VCC	2KV	I/O	1	A12	Port51	Push-pull	KB_OUT	Input WPU	
P2.5	50	-	63	-	50	-	VCC	2KV	I/O	1	A13	Port51	Push-pull	KB_OUT	Input WPU	



Notes :

1 - Pin configuration depends on product versions

Memory Organization

The AT83R5122, AT8xC5122/23 devices have separated address spaces for Program and Data Memory, as shown in Figure 12 on page 27, Figure 13 on page 29 and Figure 14 on page 30. The logical separation of Program and Data memory allows the Data Memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an-bit CPU. Nevertheless, 16-bit Data Memory addresses can also be generated through the DPTR register.

Program Memory Managament

Depending on the state of EA pin, the MCU fetches the code from internal or external program memory (ROMless mode)

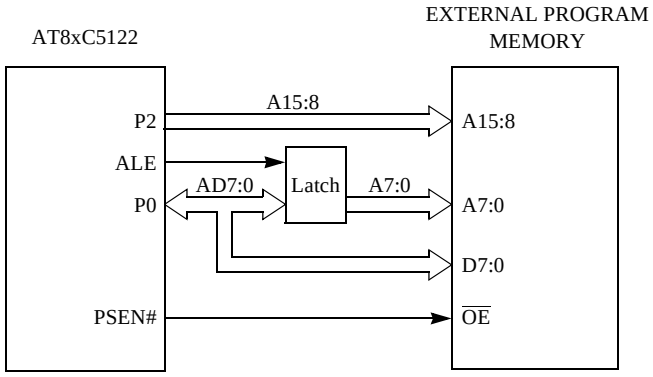
Warning : the EA pin can not be left floating, otherwise MCU may have an unpredictable behaviour.

If EA is strapped to VCC, the MCU fetches the code from the internal program memory. The way the MCU works in this mode depends on the device version. See next paragraphs for further details.

If the EA is strapped to GND, the MCU fetches the code from external program memory. This mode is common for all device versions wich supports it. After reset, the CPU begins the execution from location 0000h. There can be up to 64 KBytes of program memory. In this mode, the internal program memories are disabled.

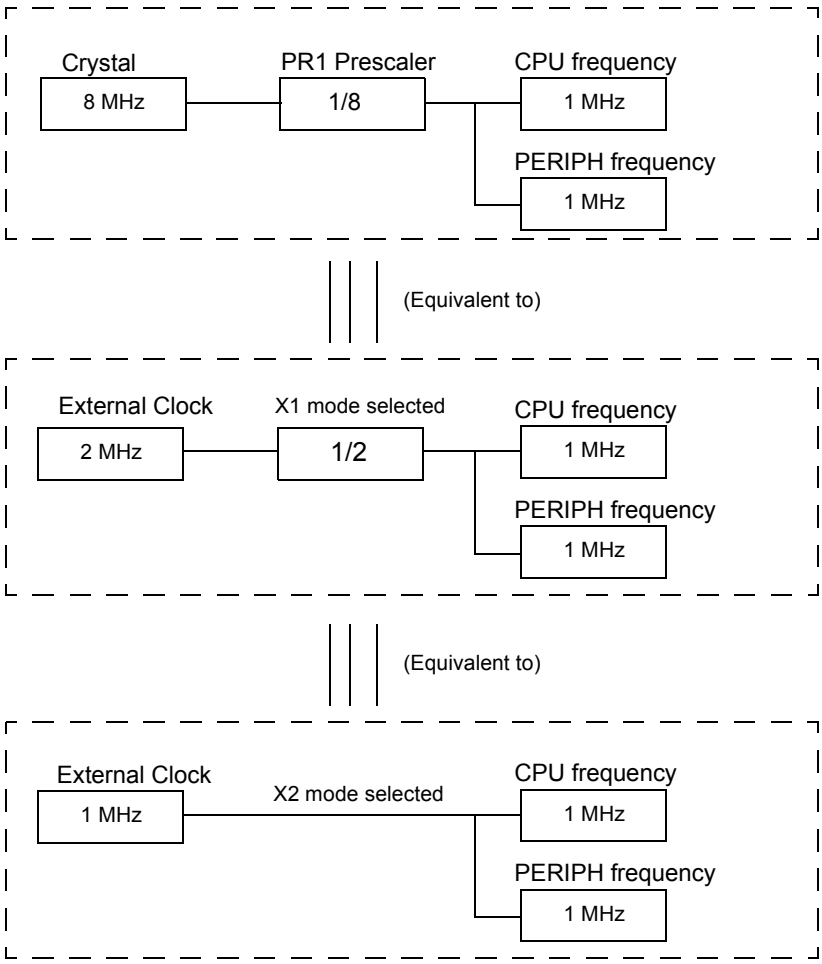
The hardware configuration for external program execution is shown in Figure 8.

Figure 8. Executing from External Program Memory



Note that the 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 serves as a multiplexed address/dat bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks the byte into an address latch. Meanwhile, Port 2 emits the high byte of the Program Counter (PCH). Then PSEN strobes the External Program Memory and the code byte is read into the MCU.

PSEN is not activated and Ports P0 and P2 are not affected during internal program fetches.

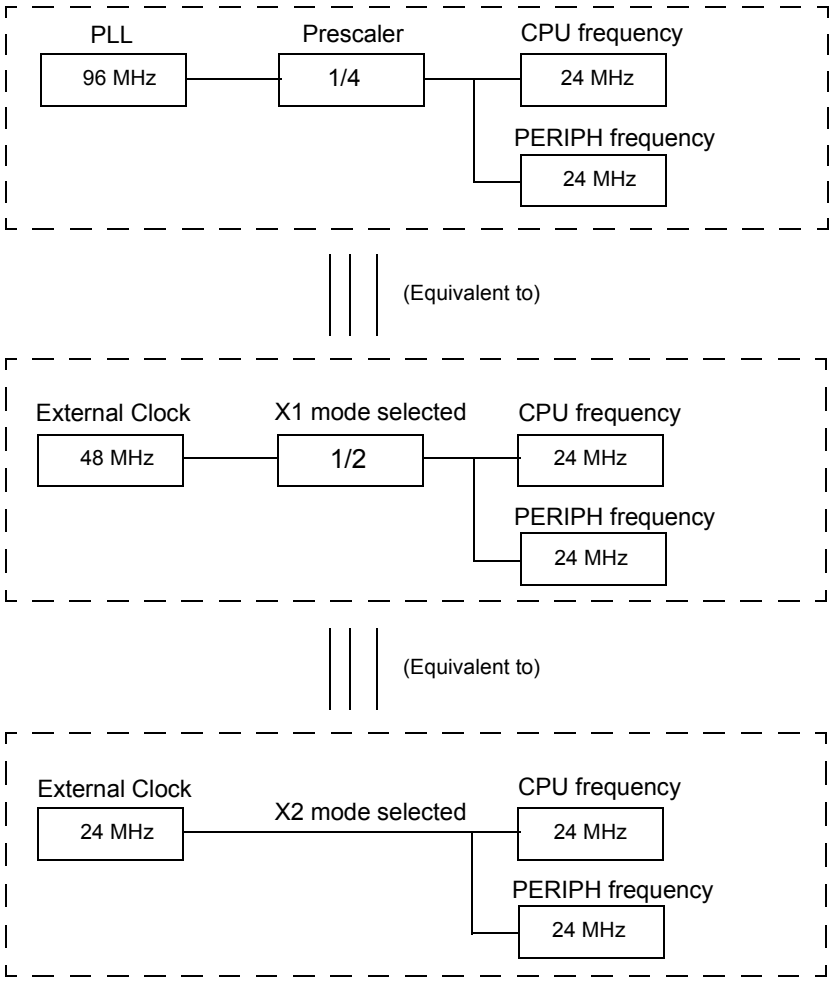


Use of PLL Clock

When the CPU clock is fed by the PLL, the X2 mode is forbidden. The bit X2 must always remain cleared in CKCON0 register. As the PR1 prescaler is always different from 1/2, the usual X1 mode can not be defined. So it is necessary to define an equivalent X1 or X2 mode from equivalent clock circuits, as in previous section.

Example: PR1=1/4, PLL feeds the CPU. The CPU works in this case at 24 MHz. This frequency could also be obtained by an equivalent clock circuit where the on-chip oscillator would run at 48 MHz in X1 mode or at 24 Mhz in X2 mode. So we can say that in this configuration, the CPU works at 48 MHz / X1 or 24 MHz / X2 (See figures below).

As the X2 bit is cleared in CKCON0 register, we have always $F_{CK\_IDLE} = F_{CK\_PERIPH}$.



SCIB Clock

The Smart Card Interface Block (SCIB) uses two clocks :

- 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 or XTAL1 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 frequency must be lower than CPU clock frequency.

During SCIB 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 41 on page 72 and Table 42 on page 72.

Two conditions must be met for a correct use of the SCIB:

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

Table 27. 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 1/2 prescaler. Set to select the 1/2 output for this peripheral.

Reset Value = XXXX XXX0b

Table 28. 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 29. 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

Registers

Table 36. 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 100K 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 37. Port Mode Register 0 - PMOD0 (91h) for AT83C5123

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 these bits are indeterminate. Do not set these bit.
3	CPRESRES	**Card Presence Pull-up resistor** Cleared to connect the internal 100K pull-up Set to disconnect the internal pull-up
2-0	-	**Reserved** The value read from these bits are indeterminate. Do not set these bit.

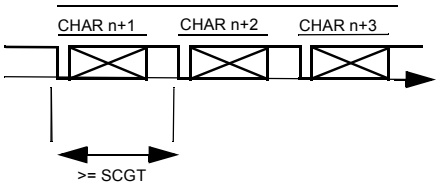
Reset Value = 00xx 0xxxb

The Guard Time counter is an 9 bit counter It is initialized at 001h at the start of a transmission by the Terminal. It then increments itself at each ETU until it reach the 9 bit value loaded into the SCGT1[0] concatenated with SCGT0[7:0]. At this time a new Terminal transmission is enabled and the Guard Time Counter stop incrementing. As soon as a new transmission start, the Guard Time Counter is re-initialized at 1 decimal value.

It should be noted that the value of the Guard Time Counter cannot be red. Reading SCGT1,0 only gives the minimum time between 2 characters that the Guard Time Counter will allow.

Care must be taken with the Guard Time Counter which counts the duration between the leading edges of 2 consecutive characters. This correspond to the character duration (10 ETU) plus the Guard Time as defined by the ISO and EMV recommendations. To program Guard Time = 2 : 2 stop bits between 2 characters which is equivalent to the minimum delay of 12 ETUs between the leading edges of 2 consecutive characters, SCGT1[0],SCGT0[7:0] should be loaded with the value 12 decimal. See Figure 30

Figure 30. Guard Time.



Block Guard Time Counter

The Block Guard Time counter provides a way to program a minimum time between the leading edge of the start bit of a character received from the ICC and the leading edge of the start bit of a character sent by the terminal. ISO IEC 7816-3 and EMV recommend a fixed Block Guard Time of 22 ETUs. The AT8xC5122/23 offer the possibility to extend this delay up to 512 ETUs.

The Block Guard Time is a 9 bit counter. When the Block Guard Time mode is enabled (BGTEN=1 in SCSR register) The Block Guard Time counter is initialized at 000h at the start of each character transmissions from the ICC. It then increments at each ETU until it reach the 9 bit value loaded into shadow SCGT1,0 registers, or until it is re-initialized by the start of an new transmission from the ICC. If the Block Guard Time counter reaches the 9 bit value loaded into shadow SCGT1,0 registers, a transmission by the TERMINAL is enabled, and the Block Guard Time counter stop incrementing. The Block Guard Time counter is re-initialized at the start of each TERMINAL transmission.

The SCGT1 SCGT0 shadow registers are loaded with the content of GT[8-0] contained in the registers SCGT1[0),SCGT0(7:0] with the rising edge of the bit BGTEN in the SCSR register. See Figure 32.

Table 48. Smart Card UART Interrupt Enabling 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	**UART Transmit Buffer Empty Interrupt Enabled** 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 Overflow Interrupt Enabled** Clear this bit to disable the Card Current Overflow interrupt. Set this bit to enable the Card Current Overflow interrupt.
4	EVCARDER	**Card Voltage Error Interrupt Enabled** Clear this bit to disable the Card Voltage Error interrupt. Set this bit to enable the Card Voltage Error interrupt.
3	ESCWTI	**WaitingTime Counter Timeout Interrupt Enabled** Clear this bit to disable the Waiting Time Counter timeout interrupt. Set this bit to enable the Waiting Time Counter timeout interrupt.
2	ESCTI	**UART Transmitted Character Interrupt Enabled** Clear this bit to disable the Smart Card UART Transmitted Character interrupt. Set this bit to enable the Smart Card UART Transmitted Character interrupt.
1	ESCRI	**UART Received Character Interrupt Enabled** Clear this bit to disable the Smart Card UART Received Character interrupt. Set this bit to enable the Smart Card UART Received Character interrupt.
0	ESCPI	**Character Reception Parity Error Interrupt Enabled** Clear this bit to disable the Smart Card Character Reception Parity Error interrupt. Set this bit to enable the Smart Card Character Reception Parity Error interrupt.

Reset Value = 0X00 0000b

Table 49. 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 character received from the ICC and the first character sent by the Terminal. 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 transmission (1 original + 4 repetitions) before parity error indication (conform to EMV) Set this bit to select 4 times transmission (1 original + 3 repetitions) 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 PR3 clock for CCLK1 (P1.7) pin Clear to select P1.7 port bit
0	SCRS	**Smart Card Register Selection** The SCRS bit selects which set of the SCIB registers is accessed.

Reset Value = X000 1000b

Table 50. Smart Card Transmit / Receive Buffer - SCIBUF (S:AA)

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

Bit Number	Bit Mnemonic	Description
-	-	**Smart Card Transmit / Receive Buffer** - A new byte can be written in the buffer to be transmitted on the I/O pin when SCTBE bit is set. The bits are sorted and copied on the I/O pin versus the active convention. - A new byte received from I/O pin is ready to be read when SCRI bit is set. The bits are sorted versus the active convention.

Reset Value = 0000 0000b

Table 53. 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. This is equal to ISO IEC Guard Time +10 (see Guard Time Counter description. According to ISO IEC 7816,the time between 2 consecutive leading edge start bits can be set between 11 and 266 (11 to 254+12 ETUs).

Reset Value = 0000 1100b

Table 54. 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 53).

Reset Value = XXXX XXX0b

Table 55. Smart Card Character/Block Waiting 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]	**Waiting Time Byte3** Used together with WT[23:0] in registers SCWT2,SCWT1, SCWT0 (see Table 56).

Reset Value = 0000 0000b

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 different depending upon the required Card Vcc. One procedure apply for Card Vcc =< 3 volts and one procedure for Card Vcc = 5 volts.

The initialization procedure involves :

- Select the CVCC level by means of bits VCARD[1:0] in SCICR 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)

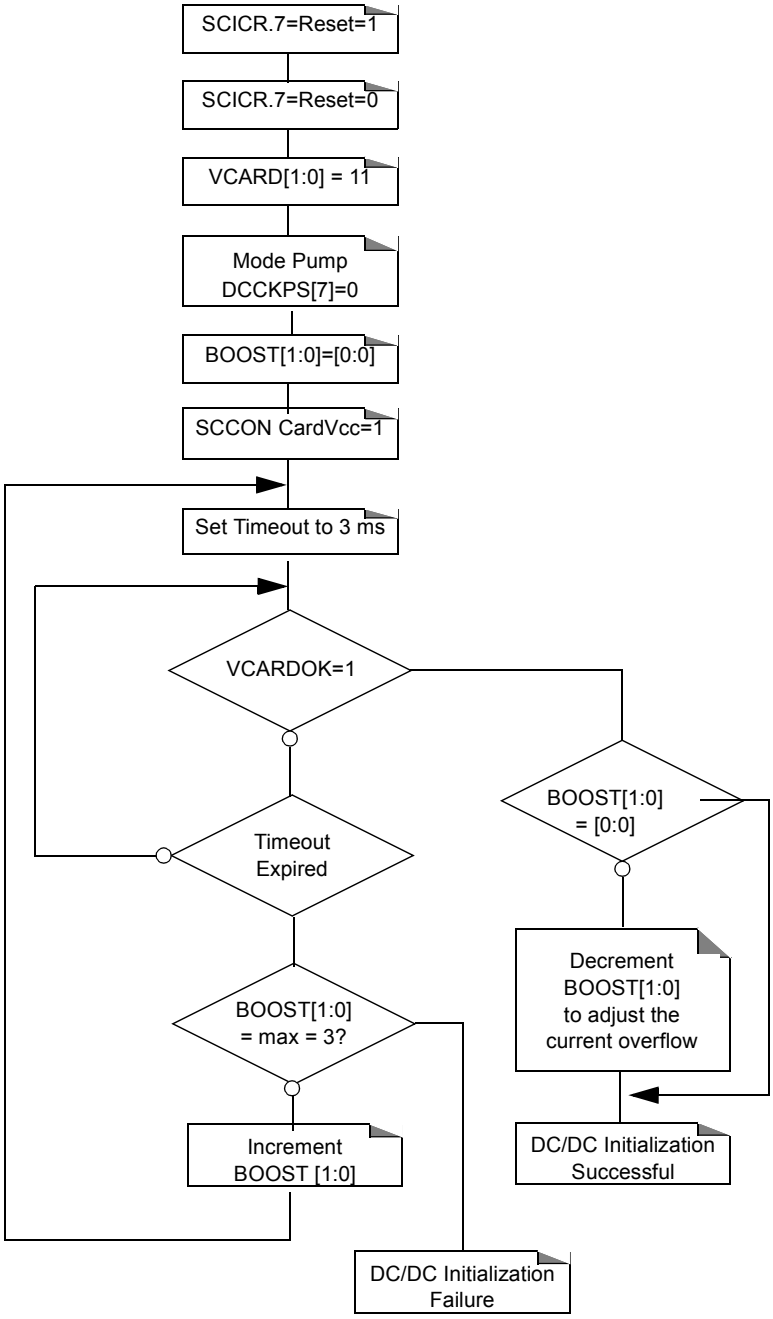
Procedure for CVcc =< 3 volts

The DC/DC regulation mode must be selected for Card Vcc = 1.8 volts and Card Vcc = 3 volts (MODE = 1 in DCCKPS register) The detailed procedures is described in flow chart of Figure 45. for Card Vcc = 1.8 volts and in the flow chart of Figure 46. for Card Vcc = 3 volts

Procedure for CVcc = 5volts

The DC/DC pump mode must be selected (MODE = 0 in DCCKPS register). The detailed procedure is described in flow chart of Figure 47.

Figure 47. Card Vcc = 5V Initialization Procedure



While VCC remains higher than 4.0V 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 VCC > 4.0V and 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 is

Table 66. 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	EEORINT	**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 65 on page 116). Clear this bit to disable Suspend Interrupts.

Reset Value = 0001 0000b

Table 67. 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

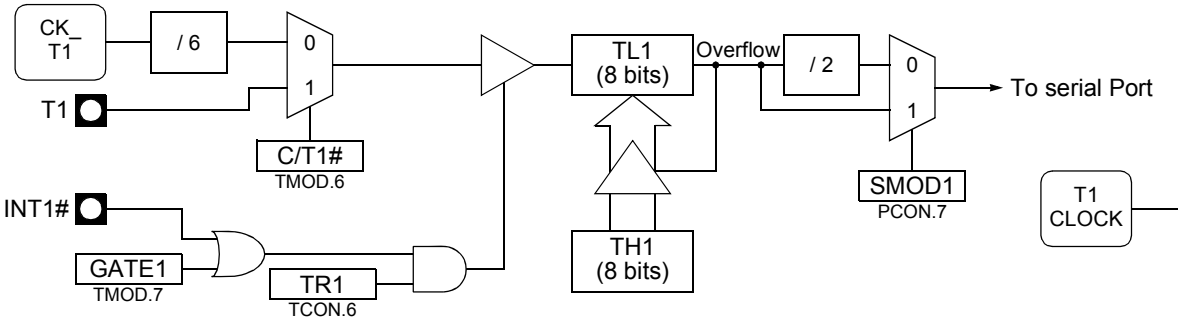
Table 70. USB Endpoint Status and Control Register X - UEPSTAX (S:CEh) X=EPNUM set in UEPNUM Register)

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 118). 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 116) 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 65 on page 116).
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 116) 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 65 on page 116). 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 116) 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 65). This bit should be cleared by the device firmware before setting TXRDY.

Reset Value = 0000 0000b

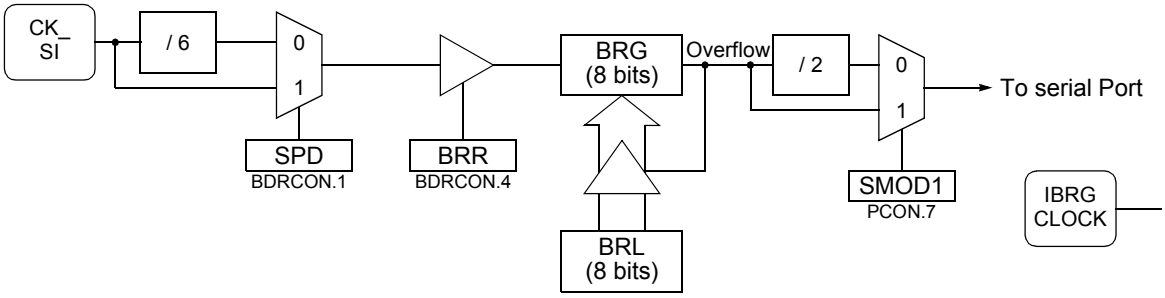
Figure 67. Timer 1 Baud Rate Generator Block Diagram



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 68 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 Table 82 on page 134). The Internal Baud Rate Generator is enabled by setting BRR bit in BDRCON register. SMOD1 bit in PCON register allows doubling of the generated baud rate.

Figure 68. 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 69 shows the serial port block diagram in Mode 0.

Baud Rate Selection (Mode 0)

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

As shown in Figure 72, the selection is done using M0SRC bit in BDRCON register.

Figure 73 gives the baud rate calculation formulas for each baud rate source.

Figure 72. Baud Rate Source Selection (Mode 0)

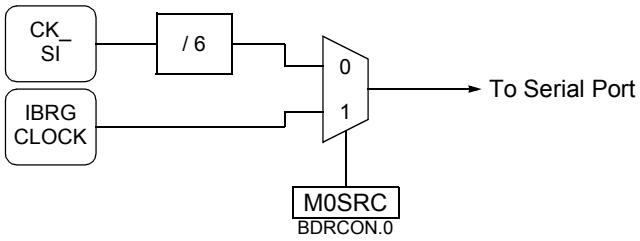


Figure 73. Baud Rate Formulas (Mode 0)

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

a. Fixed Formula

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

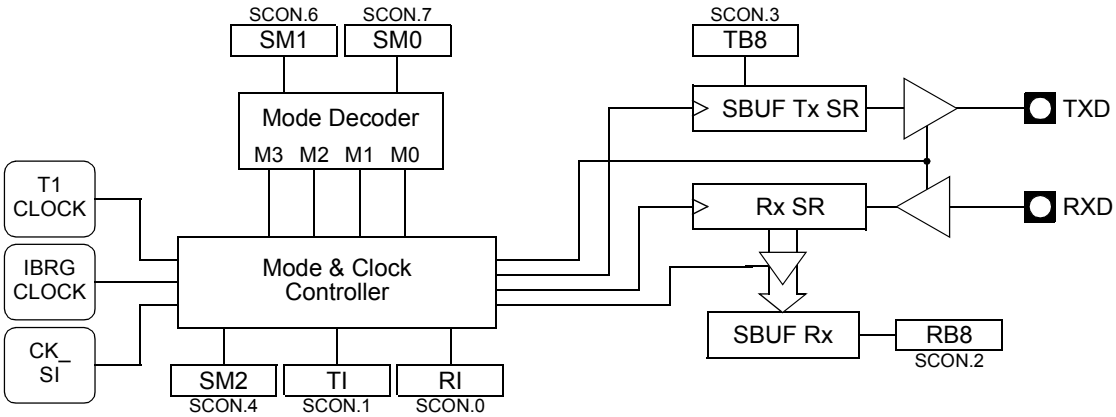
$$BRL = 256 - \frac{2^{SMOD1} \cdot F_{CK\_SI}}{6^{(1-SPD)} \cdot 32 \cdot Baud\_Rate}$$

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 74 shows the Serial Port block diagram in such asynchronous modes.

Figure 74. 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 75) 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 85. Data Transmission Format (CPHA = 0)

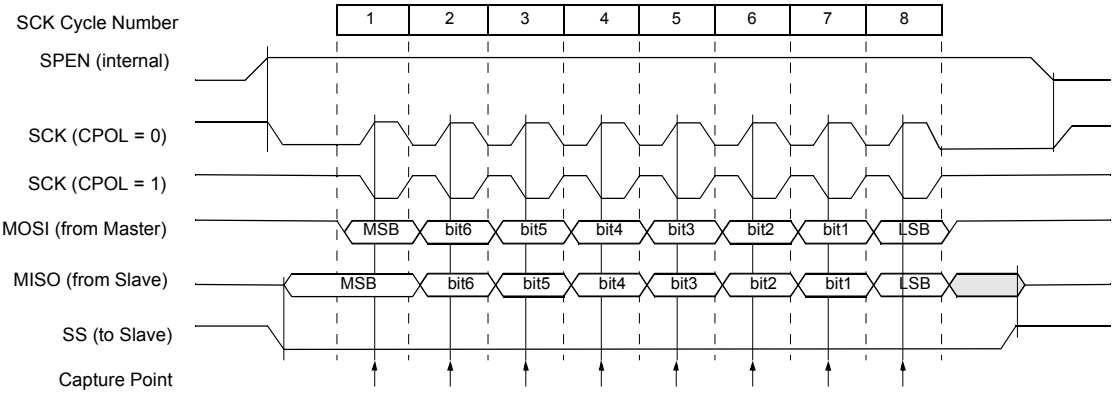
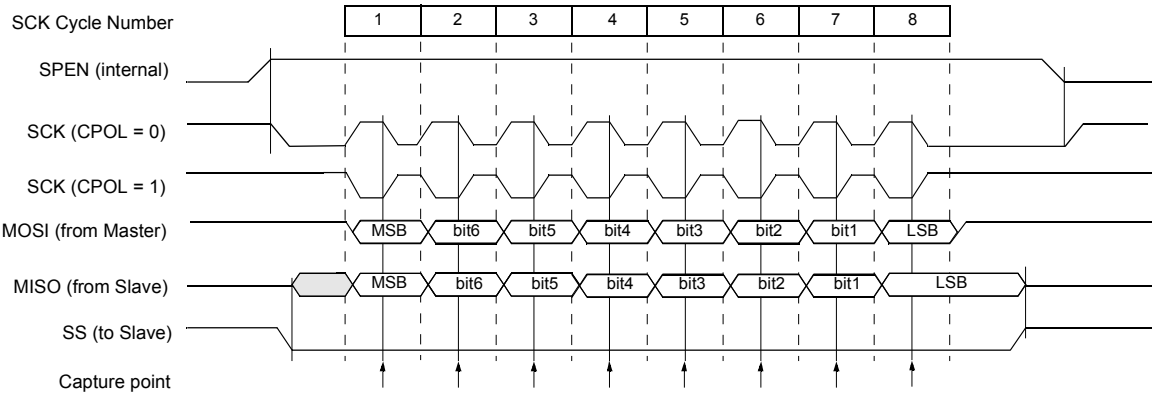


Figure 86. Data Transmission Format (CPHA = 1)



As shown in Figure 85, 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 87).

Figure 87. CPHA/$\overline{SS}$ Timing

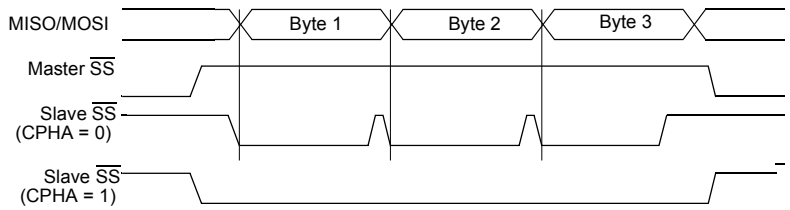


Figure 86 shows an SPI transmission in which CPHA is “1”. In this case, the Master begins driving its MOSI pin on the first SCK edge. Therefore, the Slave uses the first SCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmis-

sions (Figure 87). 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 84. 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.

Serial Peripheral DATa Register (SPDAT)

The Serial Peripheral Data Register (Table 87) 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 87. 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