Features

- **Programmable Audio Output for Interfacing with Common Audio DAC**
  - **PCM Format Compatible**
  - **I^2S Format Compatible**
- **8-bit MCU C51 Core-based (F_{MAX} = 20 MHz)**
- **2304 Bytes of Internal RAM**
- **64K Bytes of Code Memory**
  - **AT89C5132: Flash (100K Write/Erase Cycles)**
- **4K Bytes of Boot Flash Memory (AT89C5132)**
  - **ISP: Download from USB (standard) or UART (option)**
- **USB Rev 1.1 Device Controller**
  - **"Full Speed" Data Transmission**
- **Built-in PLL**
- **MultiMedia Card® Interface Compatibility**
- **Atmel DataFlash® SPI Interface Compatibility**
- **IDE/ATAPI Interface**
- **2 Channels 10-bit ADC, 8 kHz (8 True Bits)**
  - **Battery Voltage Monitoring**
  - **Voice Recording Controlled by Software**
- **Up to 44 Bits of General-purpose I/Os**
  - **4-bit Interrupt Keyboard Port for a 4 x n Matrix**
  - **SmartMedia® Software Interface**
- **Two Standard 16-bit Timers/Counters**
- **Hardware Watchdog Timer**
- **Standard Full Duplex UART with Baud Rate Generator**
- **Two Wire Master and Slave Modes Controller**
- **SPI Master and Slave Modes Controller**
- **Power Management**
  - **Power-on Reset**
  - **Software Programmable MCU Clock**
  - **Idle Mode, Power-down Mode**
- **Operating Conditions**
  - **3V, ±10%, 25 mA Typical Operating at 25°C**
  - **Temperature Range: -40°C to +85°C**
- **Packages**
  - **TQFP80, PLCC84 (Development Board Only)**
  - **Dice**

USB Microcontroller with 64K Bytes Flash Memory

AT89C5132

1. Description

The AT89C5132 is a mass storage device controlling data exchange between various Flash modules, HDD and CD-ROM.

The AT89C5132 includes 64K Bytes of Flash memory and allows In-System Programming through an embedded 4K Bytes of Boot Flash Memory.

The AT89C5132 include 2304 Bytes of RAM memory.

The AT89C5132 provides all the necessary features for man-machine interface including, timers, keyboard port, serial or parallel interface (USB, SPI, IDE), ADC input, I^2S output, and all external memory interface (NAND or NOR Flash, SmartMedia, MultiMedia, DataFlash cards).

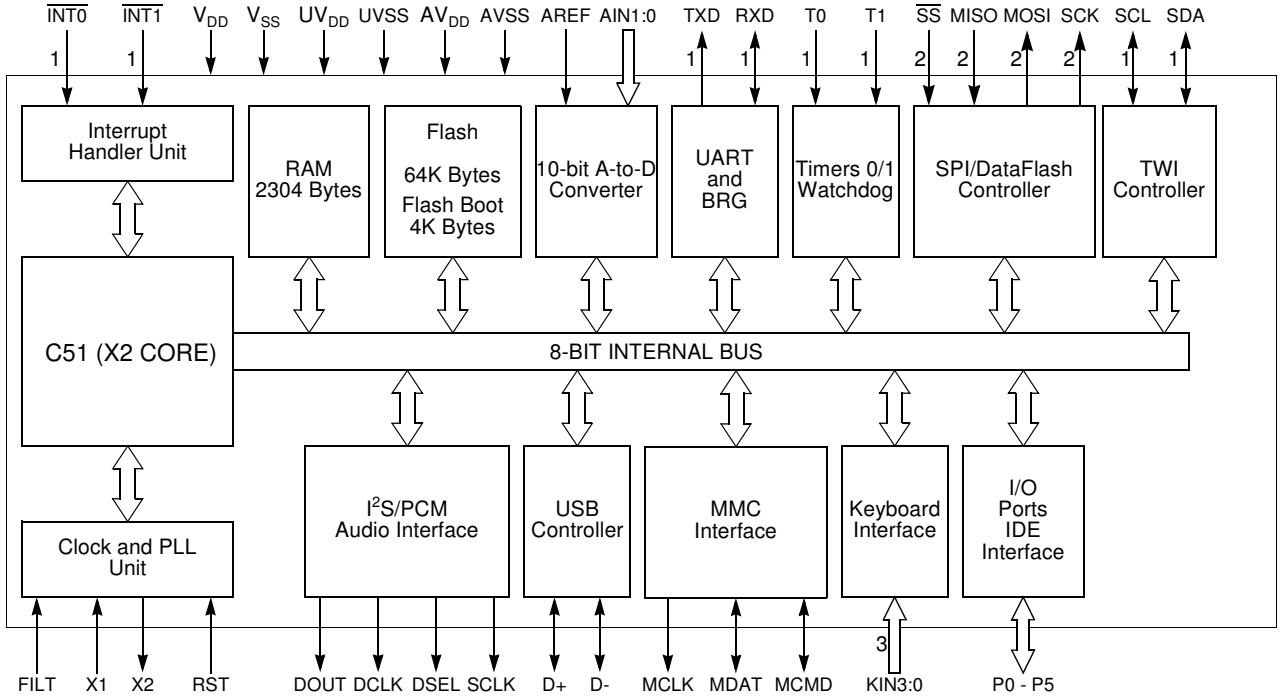
2. Typical Applications

- Flash Recorder/Writer
- PDA, Camera, Mobile Phone
- PC Add-on

4173E–USB–09/07

3. Block Diagram

Figure 3-1. AT89C5132 Block Diagram



Notes: 1. Alternate function of Port 3
2. Alternate function of Port 4
3. Alternate function of Port 1

4. Pin Description

Figure 4-1. AT89C5132 80-pin TQFP Package

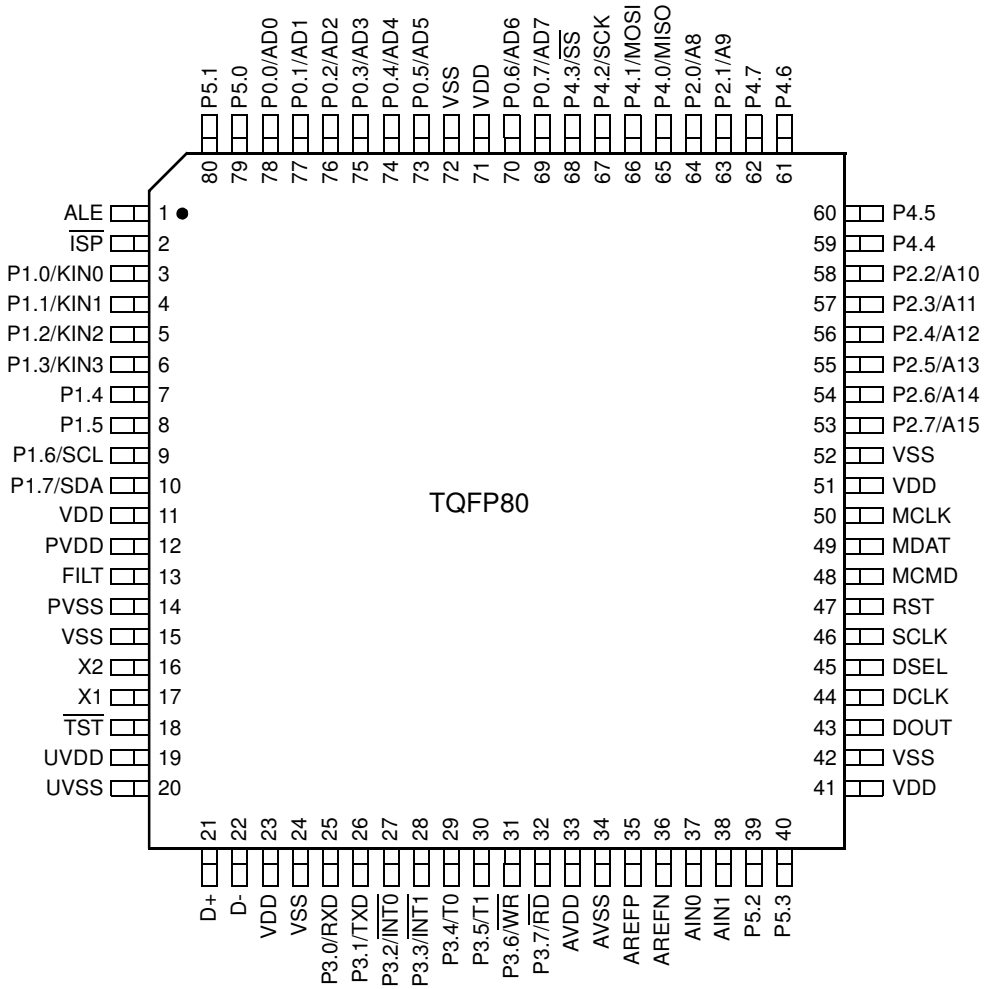
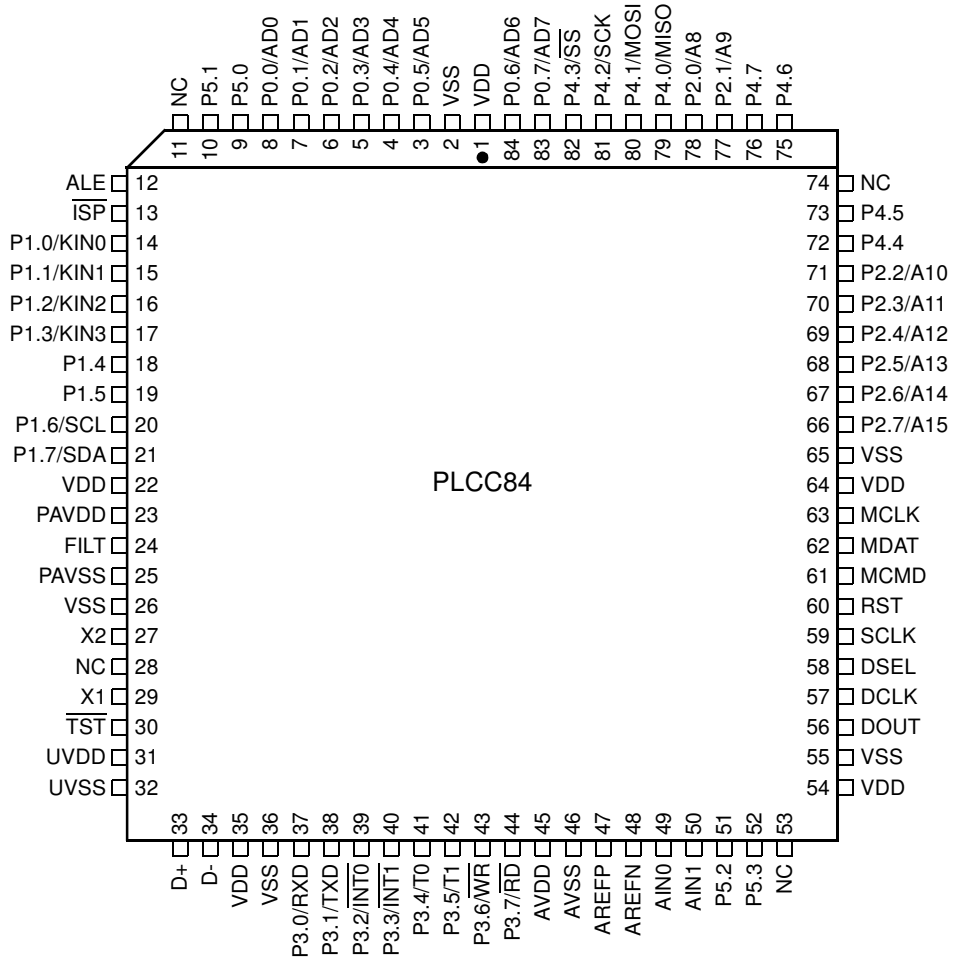


Figure 4-2. AT89C5132 84-pin PLCC [1]



Note: 1. For development board only.

4.1 Signals

All the AT89C5132 signals are detailed by functionality in Table 1 to Table 14.

Table 1. Ports Signal Description

Signal Name	Type	Description	Alternate Function
P0.7:0	I/O	**Port 0** P0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high impedance inputs. To avoid any parasitic current consumption, floating P0 inputs must be polarized to V_{DD} or V_{SS}.	AD7:0
P1.7:0	I/O	**Port 1** P1 is an 8-bit bidirectional I/O port with internal pull-ups.	KIN3:0 SCL SDA

Signal Name	Type	Description	Alternate Function
P2.7:0	I/O	**Port 2** P2 is an 8-bit bidirectional I/O port with internal pull-ups.	A15:8
P3.7:0	I/O	**Port 3** P3 is an 8-bit bidirectional I/O port with internal pull-ups.	RXD TXD $\overline{INT0}$ $\overline{INT1}$ T0 T1 $\overline{WR}$ $\overline{RD}$
P4.7:0	I/O	**Port 4** P4 is an 8-bit bidirectional I/O port with internal pull-ups.	MISO MOSI SCK $\overline{SS}$
P5.3:0	I/O	**Port 5** P5 is a 4-bit bidirectional I/O port with internal pull-ups.	-

Table 2. Clock Signal Description

Signal Name	Type	Description	Alternate Function
X1	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. X1 is the clock source for internal timing.	-
X2	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 X2 unconnected.	-
FILT	I	**PLL Low Pass Filter input** FILT receives the RC network of the PLL low pass filter.	-

Table 3. Timer 0 and Timer 1 Signal Description

Signal Name	Type	Description	Alternate Function
$\overline{INT0}$	I	**Timer 0 Gate Input** $\overline{INT0}$ serves as external run control for timer 0, when selected by GATE0 bit in TCON register. **External Interrupt 0** $\overline{INT0}$ input sets IE0 in the TCON register. If bit IT0 in this register is set, bit IE0 is set by a falling edge on $\overline{INT0}$. If bit IT0 is cleared, bit IE0 is set by a low level on $\overline{INT0}$.	P3.2
$\overline{INT1}$	I	**Timer 1 Gate Input** $\overline{INT1}$ serves as external run control for timer 1, when selected by GATE1 bit in TCON register. **External Interrupt 1** $\overline{INT1}$ input sets IE1 in the TCON register. If bit IT1 in this register is set, bit IE1 is set by a falling edge on $\overline{INT1}$. If bit IT1 is cleared, bit IE1 is set by a low level on $\overline{INT1}$.	P3.3

Signal Name	Type	Description	Alternate Function
T0	I	**Timer 0 External Clock Input** When timer 0 operates as a counter, a falling edge on the T0 pin increments the count.	P3.4
T1	I	**Timer 1 External Clock Input** When timer 1 operates as a counter, a falling edge on the T1 pin increments the count.	P3.5

Table 4. Audio Interface Signal Description

Signal Name	Type	Description	Alternate Function
DCLK	O	**DAC Data Bit Clock**	-
DOUT	O	**DAC Audio Data**	-
DSEL	O	**DAC Channel Select Signal** DSEL is the sample rate clock output.	-
SCLK	O	**DAC System Clock** SCLK is the oversampling clock synchronized to the digital audio data (DOUT) and the channel selection signal (DSEL).	-

Table 5. USB Controller Signal Description

Signal Name	Type	Description	Alternate Function
D+	I/O	**USB Positive Data Upstream Port** This pin requires an external 1.5 KΩ pull-up to V_{DD} for full speed operation.	-
D-	I/O	**USB Negative Data Upstream Port**	-

Table 6. MutiMediaCard Interface Signal Description

Signal Name	Type	Description	Alternate Function
MCLK	O	**MMC Clock output** Data or command clock transfer.	-
MCMD	I/O	**MMC Command line** Bidirectional command channel used for card initialization and data transfer commands. To avoid any parasitic current consumption, unused MCMD input must be polarized to V_{DD} or V_{SS}.	-
MDAT	I/O	**MMC Data line** Bidirectional data channel. To avoid any parasitic current consumption, unused MDAT input must be polarized to V_{DD} or V_{SS}.	-

Table 7. UART Signal Description

Signal Name	Type	Description	Alternate Function
RXD	I/O	**Receive Serial Data** RXD sends and receives data in serial I/O mode 0 and receives data in serial I/O modes 1, 2 and 3.	P3.0
TXD	O	**Transmit Serial Data** TXD outputs the shift clock in serial I/O mode 0 and transmits data in serial I/O modes 1, 2 and 3.	P3.1

Table 8. SPI Controller Signal Description

Signal Name	Type	Description	Alternate Function
MISO	I/O	**SPI Master Input Slave Output Data Line** When in master mode, MISO receives data from the slave peripheral. When in slave mode, MISO outputs data to the master controller.	P4.0
MOSI	I/O	**SPI Master Output Slave Input Data Line** When in master mode, MOSI outputs data to the slave peripheral. When in slave mode, MOSI receives data from the master controller.	P4.1
SCK	I/O	**SPI Clock Line** When in master mode, SCK outputs clock to the slave peripheral. When in slave mode, SCK receives clock from the master controller.	P4.2
$\overline{SS}$	I	**SPI Slave Select Line** When in controlled slave mode, $\overline{SS}$ enables the slave mode.	P4.3

Table 9. TWI Controller Signal Description

Signal Name	Type	Description	Alternate Function
SCL	I/O	**TWI Serial Clock** When TWI controller is in master mode, SCL outputs the serial clock to the slave peripherals. When TWI controller is in slave mode, SCL receives clock from the master controller.	P1.6
SDA	I/O	**TWI Serial Data** SDA is the bidirectional Two Wire data line.	P1.7

Table 10. A/D Converter Signal Description

Signal Name	Type	Description	Alternate Function
AIN1:0	I	**A/D Converter Analog Inputs**	-
AREFP	I	**Analog Positive Voltage Reference Input**	-
AREFN	I	**Analog Negative Voltage Reference Input** This pin is internally connected to AVSS.	-

Table 11. Keypad Interface Signal Description

Signal Name	Type	Description	Alternate Function
KIN3:0	I	**Keypad Input Lines** Holding one of these pins high or low for 24 oscillator periods triggers a keypad interrupt.	P1.3:0

Table 12. External Access Signal Description

Signal Name	Type	Description	Alternate Function
A15:8	I/O	**Address Lines** Upper address lines for the external bus. Multiplexed higher address and data lines for the IDE interface.	P2.7:0
AD7:0	I/O	**Address/Data Lines** Multiplexed lower address and data lines for the external memory or the IDE interface.	P0.7:0
ALE	O	**Address Latch Enable Output** ALE signals the start of an external bus cycle and indicates that valid address information is available on lines A7:0. An external latch is used to demultiplex the address from address/data bus.	-
$\overline{ISP}$	I/O	**ISP Enable Input** This signal must be held to GND through a pull-down resistor at the falling reset to force execution of the internal bootloader.	-
$\overline{RD}$	O	**Read Signal** Read signal asserted during external data memory read operation.	P3.7
$\overline{WR}$	O	**Write Signal** Write signal asserted during external data memory write operation.	P3.6

Table 13. System Signal Description

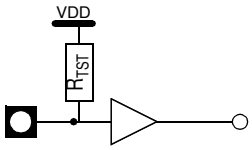
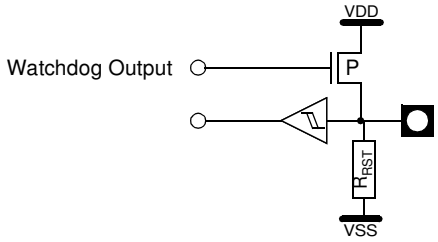
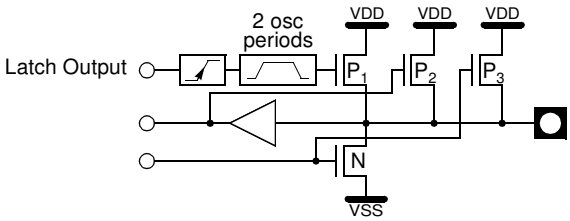
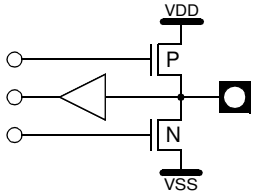
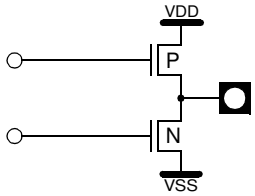
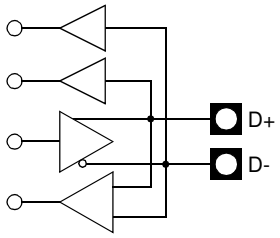
Signal Name	Type	Description	Alternate Function
RST	I	**Reset Input** Holding this pin high for 64 oscillator periods while the oscillator is running resets the device. The Port pins are driven to their reset conditions when a voltage lower than V_{IL} is applied, whether or not the oscillator is running. This pin has an internal pull-down resistor which allows the device to be reset by connecting a capacitor between this pin and V_{DD}. Asserting RST when the chip is in Idle mode or Power-Down mode returns the chip to normal operation.	-
$\overline{TST}$	I	**Test Input** Test mode entry signal. This pin must be set to V_{DD}.	-

Table 14. Power Signal Description

Signal Name	Type	Description	Alternate Function
VDD	PWR	**Digital Supply Voltage** Connect these pins to +3V supply voltage.	-
VSS	GND	**Circuit Ground** Connect these pins to ground.	-
AVDD	PWR	**Analog Supply Voltage** Connect this pin to +3V supply voltage.	-
AVSS	GND	**Analog Ground** Connect this pin to ground.	-
PVDD	PWR	**PLL Supply voltage** Connect this pin to +3V supply voltage.	-
PVSS	GND	**PLL Circuit Ground** Connect this pin to ground.	-
UVDD	PWR	**USB Supply Voltage** Connect this pin to +3V supply voltage.	-
UVSS	GND	**USB Ground** Connect this pin to ground.	-

4.2 Internal Pin Structure

Table 15. Detailed Internal Pin Structure

Circuit[1]	Type	Pins
	Input	$\overline{TST}$
	Input/Output	RST
	Input/Output	P1[2] P2[3] P3 P4 P53:0
	Input/Output	P0 MCMD MDAT $\overline{ISP}$ $\overline{PSEN}$
	Output	ALE SCLK DCLK DOUT DSEL MCLK
	Input/Output	D+ D-

Notes:
1. For information on resistors value, input/output levels, and drive capability, refer to the Section “DC Characteristics”, page 183.
2. When the Two Wire controller is enabled, P_1, P_2, and P_3 transistors are disabled allowing pseudo open-drain structure.
3. In Port 2, P_1 transistor is continuously driven when outputting a high level bit address (A15:8).

5. Address Spaces

The AT8xC5132 derivatives implement four different address spaces:

- Program/Code Memory
- Boot Memory
- Data Memory
- Special Function Registers (SFRs)

5.0.1 Code Memory

The AT89C5132 implements 64K Bytes of on-chip program/code memory in Flash technology.

The Flash memory increases ROM functionality by enabling in-circuit electrical erasure and programming. Thanks to the internal charge pump, the high voltage needed for programming or erasing Flash cells is generated on-chip using the standard V_{DD} voltage. Thus, the AT89C5132 can be programmed using only one voltage and allows in application software programming commonly known as IAP. Hardware programming mode is also available using specific programming tools.

5.0.2 Boot Memory

The AT89C5132 implements 4K Bytes of on-chip boot memory provided in Flash technology. This boot memory is delivered programmed with a standard bootloader software allowing in system programming commonly known as ISP. It also contains some Application Programming Interfaces routines commonly known as API allowing user to develop his own bootloader.

5.0.3 Data Memory

The AT89C5132 derivatives implement 2304 bytes of on-chip data RAM. This memory is divided in two separate areas:

- 256 bytes of on-chip RAM memory (standard C51 memory).
- 2048 bytes of on-chip expanded RAM memory (ERAM accessible via MOVX instructions).

6. Clock Controller

The AT89C5132 clock controller is based on an on-chip oscillator feeding an on-chip Phase Lock Loop (PLL). All internal clocks to the peripherals and CPU core are generated by this controller.

6.1 Oscillator

The AT89C5132 X1 and X2 pins are the input and the output of a single-stage on-chip inverter (see Figure 6-1) that can be configured with off-chip components such as a Pierce oscillator (see Figure 6-2). Value of capacitors and crystal characteristics are detailed in the Section “DC Characteristics”.

The oscillator outputs three different clocks: a clock for the PLL, a clock for the CPU core, and a clock for the peripherals as shown in Figure 6-1. These clocks are either enabled or disabled, depending on the power reduction mode as detailed in the section“Power Management” on page 44. The peripheral clock is used to generate the Timer 0, Timer 1, MMC, ADC, SPI, and Port sampling clocks.

Figure 6-1. Oscillator Block Diagram and Symbol

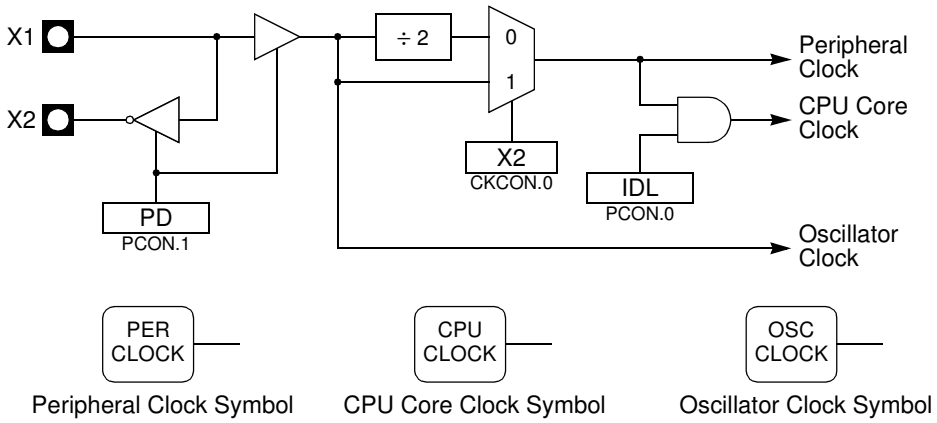
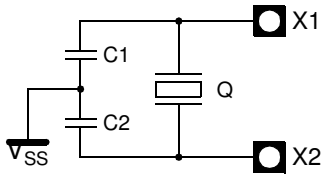


Figure 6-2. Crystal Connection



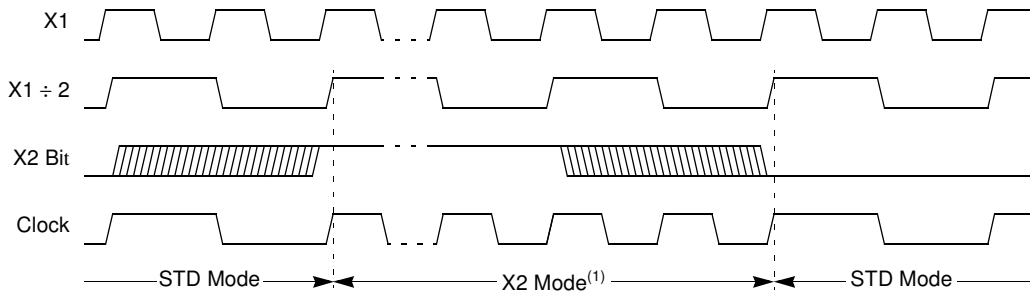
6.2 X2 Feature

Unlike standard C51 products that require 12 oscillator clock periods per machine cycle, the AT89C5132 needs only 6 oscillator clock periods per machine cycle. This feature called the “X2 feature” can be enabled using the X2 bit(1) in CKCON (see Table 1) and allows the AT89C5132 to operate in 6 or 12 oscillator clock periods per machine cycle. As shown in Figure 6-1, both CPU and peripheral clocks are affected by this feature. Figure 6-3 shows the X2 mode switching waveforms. After reset, the standard mode is activated. In standard mode, the CPU and periph-

eral clock frequency is the oscillator frequency divided by 2 while in X2 mode, it is the oscillator frequency.

Note: 1. The X2 bit reset value depends on the X2B bit in the Hardware Security Byte (see Table 12 on page 24). Using the AT89C5132 (Flash Version) the system can boot either in standard or X2 mode depending on the X2B value. Using AT83C51SND1C (ROM Version) the system always boots in standard mode. X2B bit can be changed to X2 mode later by software.

Figure 6-3. Mode Switching Waveforms



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

6.3 PLL

6.3.1 PLL Description

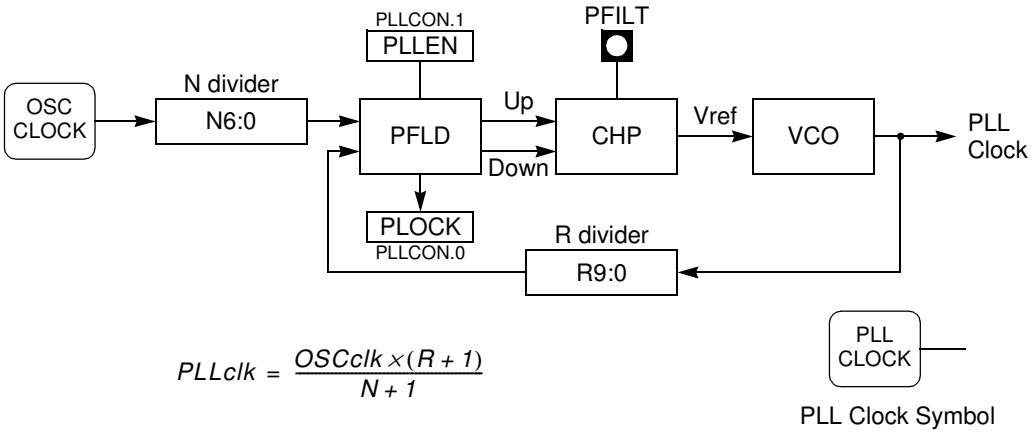
The AT89C5132 PLL is used to generate internal high frequency clock (the PLL Clock) synchronized with an external low-frequency (the Oscillator Clock). The PLL clock provides the audio interface, and the USB interface clocks. Figure 6-4 shows the internal structure of the PLL.

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 (see Table 3) 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 PFILT pin (see Figure 6-5). Value of the filter components are detailed in the Section “DC Characteristics”.

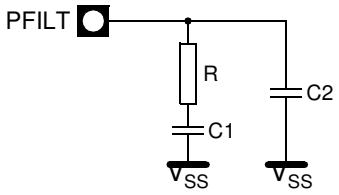
The VCO block is the Voltage Controlled Oscillator controlled by the voltage V_{ref} produced by the charge pump. It generates a square wave signal: the PLL clock.

Figure 6-4. PLL Block Diagram and Symbol



$$PLLclk = \frac{OSCclk \times (R + 1)}{N + 1}$$

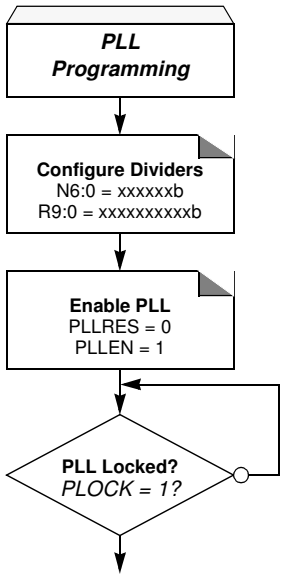
Figure 6-5. PLL Filter Connection



6.3.2 PLL Programming

The PLL is programmed using the flow shown in Figure 6-6. As soon as clock generation is enabled, the user must wait until the lock indicator is set to ensure the clock output is stable. The PLL clock frequency will depend on the audio interface clock frequencies.

Figure 6-6. PLL Programming Flow



6.4 Registers

Table 1. CKCON Register

CKCON (S:8Fh) – Clock Control Register

7	6	5	4	3	2	1	0
TWIX2	WDX2	-	SIX2	-	T1X2	T0X2	X2

Bit Number	Bit Mnemonic	Description
7	TWIX2	**Two-Wire Clock Control Bit** Set to select the oscillator clock divided by 2 as TWI clock input (X2 independent). Clear to select the peripheral clock as TWI clock input (X2 dependent).
6	WDX2	**Watchdog Clock Control Bit** Set to select the oscillator clock divided by 2 as watchdog clock input (X2 independent). Clear to select the peripheral clock as watchdog clock input (X2 dependent).
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	SIX2	**Enhanced UART Clock (Mode 0 and 2) Control Bit** Set to select the oscillator clock divided by 2 as UART clock input (X2 independent). Clear to select the peripheral clock as UART clock input (X2 dependent)..
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	T1X2	**Timer 1 Clock Control Bit** Set to select the oscillator clock divided by two as Timer 1 clock input (X2 independent). Clear to select the peripheral clock as Timer 1 clock input (X2 dependent).
1	T0X2	**Timer 0 Clock Control Bit** Set to select the oscillator clock divided by two as timer 0 clock input (X2 independent). Clear to select the peripheral clock as timer 0 clock input (X2 dependent).
0	X2	**System Clock Control Bit** Clear to select 12 clock periods per machine cycle (STD mode, $F_{CPU} = F_{PER} = F_{OSC}/2$). Set to select 6 clock periods per machine cycle (X2 mode, $F_{CPU} = F_{PER} = F_{OSC}$).

Reset Value = 0000 000Xb

Table 2. PLLNDIV Register

PLLNDIV (S:EEh) – PLL N Divider Register

7	6	5	4	3	2	1	0
-	N6	N5	N4	N3	N2	N1	N0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
6-0	N6:0	**PLL N Divider** 7-bit N divider.

Reset Value = 0000 0000b

Table 3. PLLCON Register

PLLCON (S:E9h) – PLL Control Register

7	6	5	4	3	2	1	0
R1	**R0**	**-**	**-**	**PLLRES**	**-**	**PLLEN**	**PLOCK**

Bit Number	Bit Mnemonic	Description
7 - 6	R1:0	**PLL Least Significant Bits R Divider** 2 LSB of the 10-bit R divider.
5 - 4	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
3	PLLRES	**PLL Reset Bit** Set this bit to reset the PLL. Clear this bit to free the PLL and allow enabling.
2	-	**Reserved** The values read from this bit is always 0. Do not set this 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 1000b

Table 4. PLLRDIV Register

PLLRDIV (S:EFh) – PLL R Divider Register

7	6	5	4	3	2	1	0
R9	**R8**	**R7**	**R6**	**R5**	**R4**	**R3**	**R2**

Bit Number	Bit Mnemonic	Description
7 - 0	R9:2	**PLL Most Significant Bits R Divider** 8 MSB of the 10-bit R divider.

Reset Value = 0000 0000b

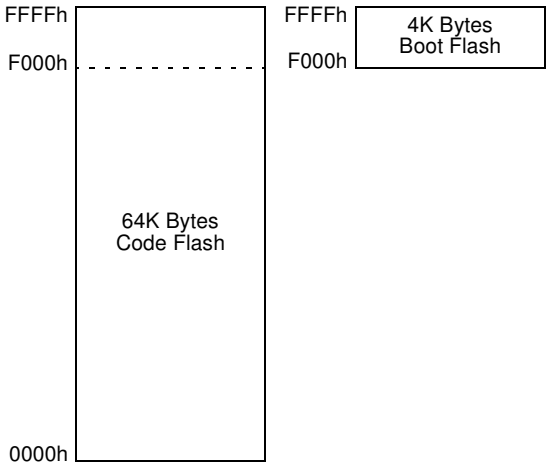
7. Program/Code Memory

The AT89C5132 implements 64K Bytes of on-chip program/code memory. Figure 7-1 shows the split of internal and external program/code memory spaces depending on the product.

The Flash memory increases EPROM and ROM functionality by in-circuit electrical erasure and programming. The high voltage needed for programming or erasing Flash cells is generated on-chip using the standard V_{DD} voltage, made possible by the internal charge pump. Thus, the AT89C5132 can be programmed using only one voltage and allows in application software programming. Hardware programming mode is also available using common programming tools. See the application note 'Programming T89C51x and AT89C51x with Device Programmers'.

The AT89C5132 implements an additional 4K Bytes of on-chip boot Flash memory provided in Flash memory. This boot memory is delivered programmed with a standard bootloader software allowing In-System Programming (ISP). It also contains some Application Programming Interfaces (API), allowing In Application Programming (IAP) by using user's own bootloader.

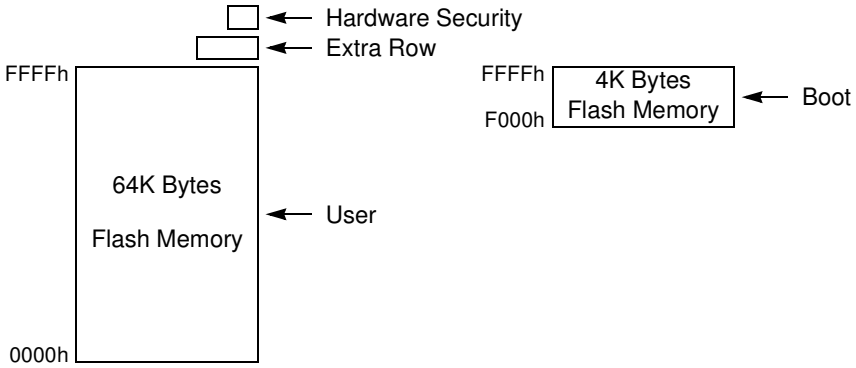
Figure 7-1. Program/Code Memory Organization



7.1 Flash Memory Architecture

As shown in Figure 7-2 the AT89C5132 Flash memory is composed of four spaces detailed in the following paragraphs.

Figure 7-2. AT89C5132 Memory Architecture



7.1.1 User Space

This space is composed of a 64K Bytes Flash memory organized in 512 pages of 128 Bytes. It contains the user's application code. This space can be read or written by both software and hardware modes.

7.1.2 Boot Space

This space is composed of a 4K Bytes Flash memory. It contains the bootloader for In-System Programming and the routines for In-System Application Programming.

This space can only be read or written by hardware mode using a parallel programming tool.

7.1.3 Hardware Security Space

This space is composed of one byte: the Hardware Security Byte (HSB see Table 7) divided in two separate nibbles see Table 7. The MSN contains the X2 mode configuration bit and the Boot Loader Jump Bit as detailed in section "Boot Memory Execution" and can be written by software while the LSN contains the lock system level to protect the memory content against piracy as detailed in section "Hardware Security System" and can only be written by hardware.

7.1.4 Extra Row Space

This space is composed of two Bytes:

- The Software Boot Vector (SBV see Table 8).
  This byte is used by the software bootloader to build the boot address.
- The Software Security Byte (SSB see Figure).
  This byte is used to lock the execution of some bootloader commands.

7.2 Hardware Security System

The AT89C5132 implements three lock Bits LB2:0 in the LSN of HSB (see Table 7) providing three levels of security for user's program as described in Table 7 while the AT83C51SND1C is always set in read disabled mode.

- Level 0 is the level of an erased part and does not enable any security feature.
- Level 1 locks the hardware programming of both user and boot memories.
- Level 2 locks hardware verifying of both user and boot memories.
- Level 3 locks the external execution.

Table 5. Lock Bit Features[1]

Level	LB2[2]	LB1	LB0	Internal Execution	External Execution	Hardware Verifying	Hardware Programming	Software Programming
0	U	U	U	Enable	Enable	Enable	Enable	Enable
1	U	U	P	Enable	Enable	Enable	Disable	Enable
2	U	P	X	Enable	Enable	Disable	Disable	Enable
3[3]	P	X	X	Enable	Disable	Disable	Disable	Enable

Notes: 1. U means unprogrammed, P means programmed and X means don't care (programmed or unprogrammed).
2. LB2 is not implemented in the AT89C5132 products.
3. AT89C5132 products are delivered with third level programmed to ensure that the code programmed by software using ISP or user's bootloader is secured from any hardware piracy.

7.3 Boot Memory Execution

As internal C51 code space is limited to 64K Bytes, some mechanisms are implemented to allow boot memory to be mapped in the code space for execution at addresses from F000h to FFFFh. The boot memory is enabled by setting the ENBOOT bit in AUXR1 (see Table 6). The three ways to set this bit are detailed in the following sections.

7.3.1 Software Boot Mapping

The software way to set ENBOOT consists in writing to AUXR1 from the user's software. This enables bootloader or API routines execution.

7.3.2 Hardware Condition Boot Mapping

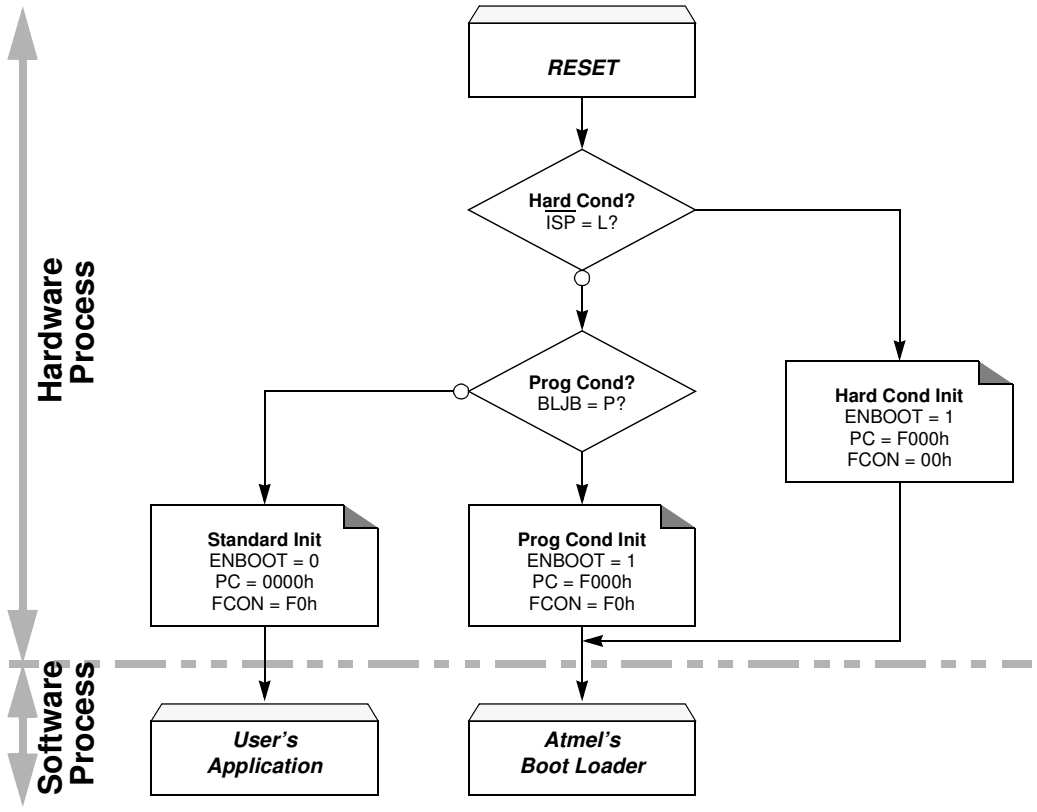
The hardware condition is based on the $\overline{ISP}$ pin. When driving this pin to low level, the chip reset sets ENBOOT and forces the reset vector to F000h instead of 0000h in order to execute the bootloader software.

As shown in Figure 7-3, the hardware condition always allows in-system recovery when user's memory has been corrupted.

7.3.3 Programmed Condition Boot Mapping

The programmed condition is based on the Bootloader Jump Bit (BLJB) in HSB. As shown in Figure 7-3, when this bit is programmed (by hardware or software programming mode), the chip resets ENBOOT and forces the reset vector to F000h instead of 0000h, in order to execute the bootloader software.

Figure 7-3. Hardware Boot Process Algorithm



The software process (bootloader) is detailed in the AT89C5132 Bootloader datasheet.

7.3.4 Preventing Flash Corruption

See “Reset Recommendation to Prevent Flash Corruption” on page 45.

7.4 Registers

Table 6. AUXR1 Register

AUXR1 (S:A2h) – Auxiliary Register 1

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

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The values read from these Bits are indeterminate. Do not set these Bits.
5	ENBOOT	**Enable Boot Flash** Set this bit to map the boot Flash in the code space between at addresses F000h to FFFFh. Clear this bit to disable boot Flash.
4	-	**Reserved** The values read from this bit is indeterminate. Do not set this bit.
3	GF3	**General Flag** This bit is a general-purpose user flag.
2	0	**Always Zero** This bit is stuck to logic 0 to allow INC AUXR1 instruction without affecting GF3 flag.
1	-	**Reserved for Data Pointer Extension.**
0	DPS	**Data Pointer Select Bit** Set to select second data pointer: DPTR1. Clear to select first data pointer: DPTR0.

Reset Value = XXXX 00X0b

7.5 Hardware Bytes

Table 7. HSB Byte – Hardware Security Byte

7	6	5	4	3	2	1	0
X2B	BLJB	-	-	-	LB2	LB1	LB0

Bit Number	Bit Mnemonic	Description
7	X2B[1]	**X2 Bit** Program this bit to start in X2 mode. Unprogram (erase) this bit to start in standard mode.
6	BLJB[2]	**Boot Loader Jump Bit** Program this bit to execute the boot loader at address F000h on next reset. Unprogram (erase) this bit to execute user’s application at address 0000h on next reset.
5 - 4	-	**Reserved** The value read from these bits is always unprogrammed. Do not program these bits.
3	-	**Reserved** The value read from this bit is always unprogrammed. Do not program this bit.

Bit Number	Bit Mnemonic	Description
2 - 0	LB2:0	**Hardware Lock Bits** Refer to for bits description.

Reset Value = XXUU UXXX, UUUU UUUU after an hardware full chip erase.

Note: 1. X2B initializes the X2 bit in CKCON during the reset phase.
2. In order to ensure boot loader activation at first power-up, AT89C5132 products are delivered with BLJB programmed.
3. Bits 0 to 3 (LSN) can only be programmed by hardware mode.

Table 8. SBV Byte – Software Boot Vector

7	6	5	4	3	2	1	0
ADD15	ADD14	ADD13	ADD12	ADD11	ADD10	ADD9	ADD8

Bit Number	Bit Mnemonic	Description
7 - 0	ADD15:8	**MSB of the user's bootloader 16-bit address location** Refer to the bootloader datasheet for usage information (bootloader dependent).

Reset Value = XXXX XXXX, UUUU UUUU after an hardware full chip erase.

Table 9. SSB Byte – Software Security Byte

7	6	5	4	3	2	1	0
SSB7	SSB6	SSB5	SSB4	SSB3	SSB2	SSB1	SSB0

Bit Number	Bit Mnemonic	Description
7 - 0	SSB7:0	**Software Security Byte Data** Refer to the bootloader datasheet for usage information (bootloader dependent).

Reset Value = XXXX XXXX, UUUU UUUU after an hardware full chip erase.

8. Data Memory

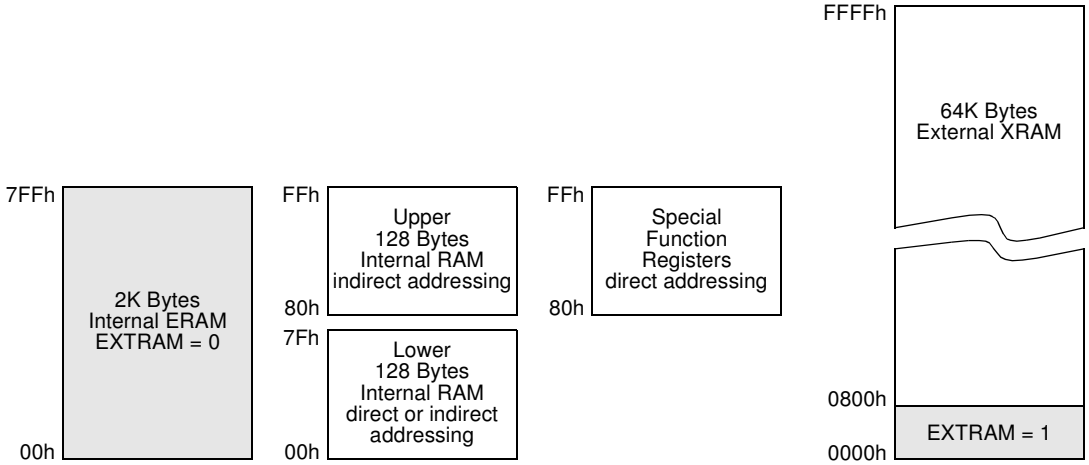
The AT89C5132 provides data memory access in two different spaces:

1. The internal space mapped in three separate segments:
   - The lower 128 Bytes RAM segment
   - The upper 128 Bytes RAM segment
   - The expanded 2048 Bytes RAM segment
2. The external space.

A fourth internal segment is available but dedicated to Special Function Registers, SFRs, (addresses 80h to FFh) accessible by direct addressing mode. For information on this segment, refer to the section “Special Function Registers”, page 29.

Figure 8-1 shows the internal and external data memory spaces organization.

Figure 8-1. Internal and External Data Memory Organization



8.1 Internal Space

8.1.1 Lower 128 Bytes RAM

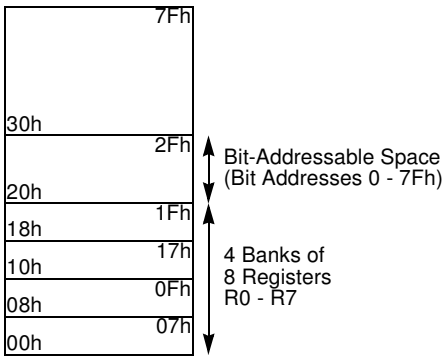
The lower 128 Bytes of RAM (see Figure 8-2) are accessible from address 00h to 7Fh using direct or indirect addressing modes. The lowest 32 Bytes are grouped into 4 banks of 8 registers (R0 to R7). Two Bits RS0 and RS1 in PSW register (see Table 13) select which bank is in use according to Table 10. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing, and can be used for context switching in interrupt service routines.

Table 10. Register Bank Selection

RS1	RS0	Description
0	0	Register bank 0 from 00h to 07h
0	1	Register bank 1 from 08h to 0Fh
1	0	Register bank 2 from 10h to 17h
1	1	Register bank 3 from 18h to 1Fh

The next 16 Bytes above the register banks form a block of bit-addressable memory space. The C51 instruction set includes a wide selection of single-bit instructions, and the 128 Bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00h to 7Fh.

Figure 8-2. Lower 128 Bytes Internal RAM Organization



8.1.2 Upper 128 Bytes RAM

The upper 128 Bytes of RAM are accessible from address 80h to FFh using only indirect addressing mode.

8.1.3 Expanded RAM

The on-chip 2K Bytes of expanded RAM (ERAM) are accessible from address 0000h to 07FFh using indirect addressing mode through MOVX instructions. In this address range, EXTRAM bit in AUXR register (see Table 14) is used to select the ERAM (default) or the XRAM. As shown in Figure 8-1 when EXTRAM = 0, the ERAM is selected and when EXTRAM = 1, the XRAM is selected, See “External Space” on page 23.

The ERAM memory can be resized using XRS1:0 Bits in AUXR register to dynamically increase external access to the XRAM space. Table 11 details the selected ERAM size and address range.

Table 11. ERAM Size Selection

XRS1	XRS0	ERAM Size	Address
0	0	256 Bytes	0 to 00FFh
0	1	512 Bytes	0 to 01FFh
1	0	1K Byte	0 to 03FFh
1	1	2K Bytes	0 to 07FFh

Note: Lower 128 Bytes RAM, Upper 128 Bytes RAM, and expanded RAM are made of volatile memory cells. This means that the RAM content is indeterminate after power-up and must then be initialized properly.

8.2 External Space

8.2.1 Memory Interface

The external memory interface comprises the external bus (port 0 and port 2) as well as the bus control signals ($\overline{RD}$, $\overline{WR}$, and ALE).

Figure 8-3 shows the structure of the external address bus. P0 carries address A7:0 while P2 carries address A15:8. Data D7:0 is multiplexed with A7:0 on P0. Table 12 describes the external memory interface signals.

Figure 8-3. External Data Memory Interface Structure

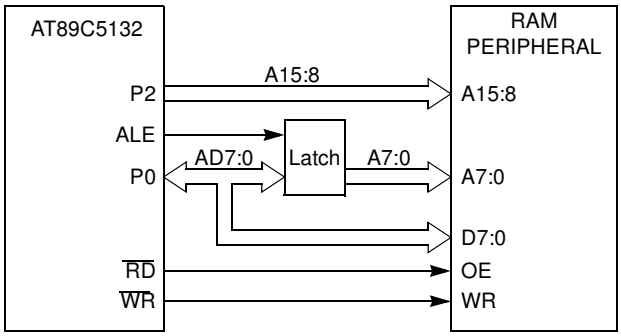


Table 12. External Data Memory Interface Signals

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

8.2.2 Page Access Mode

The AT89C5132 implement a feature called Page Access that disables the output of DPH on P2 when executing MOVX @DPTR instruction. Page Access is enable by setting the DPHDIS bit in AUXR register.

Page Access is useful when application uses both ERAM and 256 Bytes of XRAM. In this case, software modifies intensively EXTRAM bit to select access to ERAM or XRAM and must save it if used in interrupt service routine. Page Access allows external access above 00FFh address without generating DPH on P2. Thus ERAM is accessed using MOVX @Ri or MOVX @DPTR with DPTR < 0100h, < 0200h, < 0400h or < 0800h depending on the XRS1:0 bits value. Then XRAM is accessed using MOVX @DPTR with DPTR ≥ 0800h regardless of XRS1:0 bits value while keeping P2 for general I/O usage.

8.2.3 External Bus Cycles

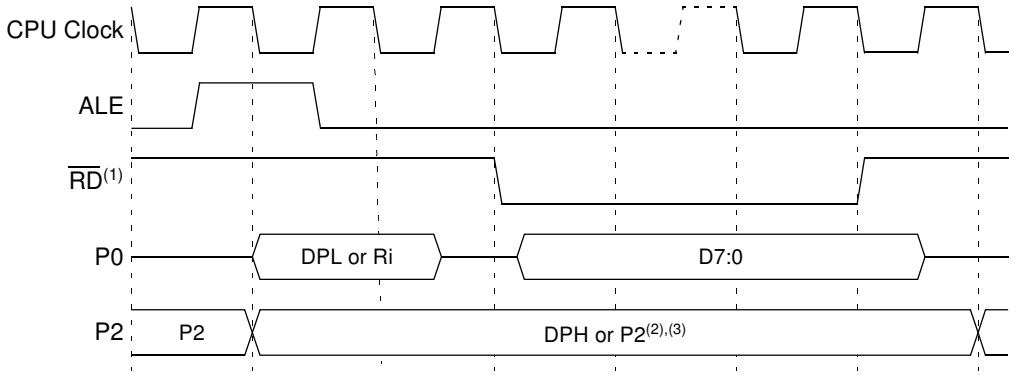
This section describes the bus cycles that AT89C5132 executes to read (see Figure 8-4), and write data (see Figure 8-5) in the external data memory.

External memory cycle takes 6 CPU clock periods. This is equivalent to 12 oscillator clock periods in standard mode or 6 oscillator clock periods in X2 mode. For further information on X2 mode, refer to the section “X2 Feature”, page 12.

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

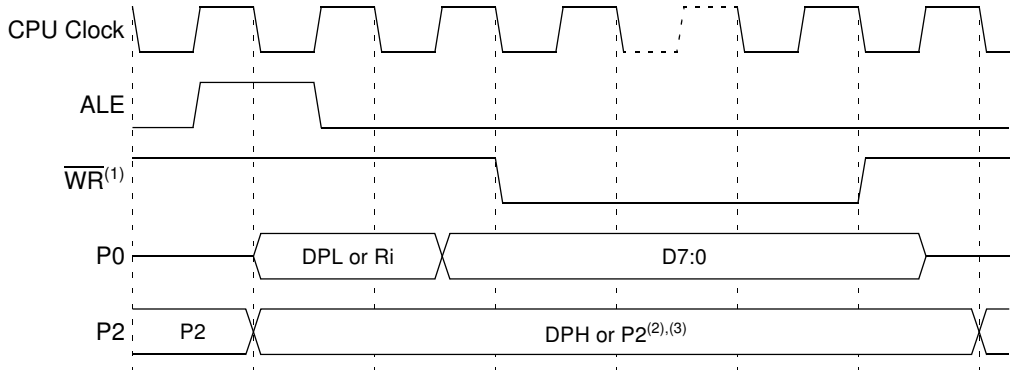
For simplicity, the accompanying figures depict the bus cycle waveforms in idealized form and do not provide precise timing information. For bus cycle timing parameters refer to the section “AC Characteristics”.

Figure 8-4. External Data Read Waveforms



Notes: 1. $\overline{RD}$ signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.
3. When executing MOVX @DPTR instruction, if DPHDIS is set (Page Access Mode), P2 outputs SFR content instead of DPH.

Figure 8-5. External Data Write Waveforms



Notes: 1. $\overline{WR}$ signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.
3. When executing MOVX @DPTR instruction, if DPHDIS is set (Page Access Mode), P2 outputs SFR content instead of DPH.

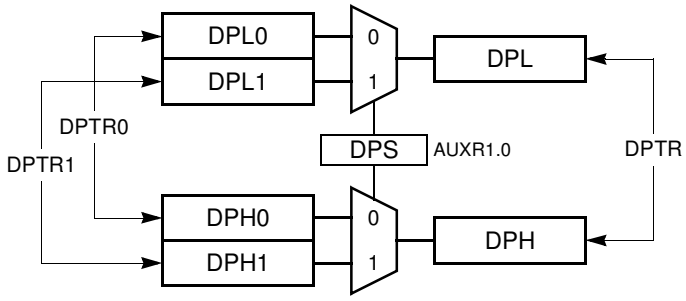
8.3 Dual Data Pointer

8.3.1 Description

The AT89C5132 implement a second data pointer for speeding up code execution and reducing code size in case of intensive usage of external memory accesses.

DPTR0 and DPTR1 are seen by the CPU as DPTR and are accessed using the SFR addresses 83h and 84h that are the DPH and DPL addresses. The DPS bit in AUXR1 register (see Table 15) is used to select whether DPTR is the data pointer 0 or the data pointer 1 (see Figure 8-6).

Figure 8-6. Dual Data Pointer Implementation



8.3.2 Application

Software can take advantage of the additional data pointers to both increase speed and reduce code size, for example, block operations (copy, compare, search ...) are well served by using one data pointer as a “source” pointer and the other one as a “destination” pointer.

Below is an example of block move implementation using the two pointers and coded in assembler. The latest C compiler also takes advantage of this feature by providing enhanced algorithm libraries.

The INC instruction is a short (2 Bytes) and fast (6 CPU clocks) way to manipulate the DPS bit in the AUXR1 register. However, note that the INC instruction does not directly forces the DPS bit to a particular state, but simply toggles it. In simple routines, such as the block move example, only the fact that DPS is toggled in the proper sequence matters, not its actual value. In other words, the block move routine works the same whether DPS is “0” or “1” on entry.

```
; ASCII block move using dual data pointers
; Modifies DPTR0, DPTR1, A and PSW
; Ends when encountering NULL character
; Note: DPS exits opposite of entry state unless an extra INC AUXR1 is added

AUXR1    EQU     0A2h

move:    mov     DPTR,#SOURCE   ; address of SOURCE
         inc     AUXR1          ; switch data pointers
         mov     DPTR,#DEST     ; address of DEST
mv_loop: inc     AUXR1          ; switch data pointers
         movx    A,@DPTR        ; get a byte from SOURCE
         inc     DPTR           ; increment SOURCE address
         inc     AUXR1          ; switch data pointers
         movx    @DPTR,A        ; write the byte to DEST
         inc     DPTR           ; increment DEST address
         jnz     mv_loop        ; check for NULL terminator
end_move:
```

8.4 Registers

Table 13. PSW Register
PSW (S:8Eh) – Program Status Word Register

7	6	5	4	3	2	1	0
CY	**AC**	**F0**	**RS1**	**RS0**	**OV**	**F1**	**P**

Bit Number	Bit Mnemonic	Description
7	CY	**Carry Flag** Carry out from bit 1 of ALU operands.
6	AC	**Auxiliary Carry Flag** Carry out from bit 1 of addition operands.
5	F0	**User Definable Flag 0.**
4 - 3	RS1:0	**Register Bank Select Bits** Refer to Table 10 for Bits description.
2	OV	**Overflow Flag** Overflow set by arithmetic operations.
1	F1	**User Definable Flag 1**
0	P	**Parity Bit** Set when ACC contains an odd number of 1's. Cleared when ACC contains an even number of 1's.

Reset Value = 0000 0000b

Table 14. AUXR Register
AUXR (S:8Eh) – Auxiliary Control Register

7	6	5	4	3	2	1	0
-	**EXT16**	**M0**	**DPHDIS**	**XRS1**	**XRS0**	**EXTRAM**	**AO**

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The values read from this bit is indeterminate. Do not set this bit.
6	EXT16	**External 16-bit Access Enable Bit** Set to enable 16-bit access mode during MOVX instructions. Clear to disable 16-bit access mode and enable standard 8-bit access mode during MOVX instructions.
5	M0	**External Memory Access Stretch Bit** Set to stretch $\overline{RD}$ or $\overline{WR}$ signals duration to 15 CPU clock periods. Clear not to stretch $\overline{RD}$ or $\overline{WR}$ signals and set duration to 3 CPU clock periods.
4	DPHDIS	**DPH Disable Bit** Set to disable DPH output on P2 when executing MOVX @DPTR instruction. Clear to enable DPH output on P2 when executing MOVX @DPTR instruction.
3 - 2	XRS1:0	**Expanded RAM Size Bits** Refer to Table 11 for ERAM size description.

Bit Number	Bit Mnemonic	Description
1	EXTRAM	**External RAM Enable Bit** Set to select the external XRAM when executing MOVX @Ri or MOVX @DPTR instructions. Clear to select the internal expanded RAM when executing MOVX @Ri or MOVX @DPTR instructions.
0	AO	**ALE Output Enable Bit** Set to output the ALE signal only during MOVX instructions. Clear to output the ALE signal at a constant rate of $F_{CPU}/3$.

Reset Value = X000 1101b

9. Special Function Registers

The Special Function Registers (SFRs) of the AT89C5132 derivatives fall into the categories detailed in Table 15 to Table 30. The relative addresses of these SFRs are provided together with their reset values in Table 31. In this table, the bit-addressable registers are identified by Note 1.

Table 15. C51 Core SFRs

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

Table 16. System Management SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
PCON	87h	Power Control	SMOD1	SMOD0	-	-	GF1	GF0	PD	IDL
AUXR	8Eh	Auxiliary Register 0	-	EXT16	M0	DPHDIS	XRS1	XRS0	EXTRAM	AO
AUXR1	A2h	Auxiliary Register 1	-	-	ENBOOT	-	GF3	0	-	DPS
NVERS	FBh	Version Number	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0

Table 17. PLL and System Clock SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CKCON	8Fh	Clock Control	-	WDX2	-	-	-	T1X2	T0X2	X2
PLLCON	E9h	PLL Control	R1	R0	-	-	PLLRES	-	PLLEN	PLOCK
PLLNDIV	EEh	PLL N Divider	-	N6	N5	N4	N3	N2	N1	N0
PLLRDIV	EFh	PLL R Divider	R9	R8	R7	R6	R5	R4	R3	R2

Table 18. Interrupt SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
IEN0	A8h	Interrupt Enable Control 0	EA	EAUD	-	ES	ET1	EX1	ET0	EX0
IEN1	B1h	Interrupt Enable Control 1	-	EUSB	-	EKB	EADC	ESPI	EI2C	EMMC
IPH0	B7h	Interrupt Priority Control High 0	-	IPHAUD	-	IPHS	IPHT1	IPHX1	IPHT0	IPHX0
IPL0	B8h	Interrupt Priority Control Low 0	-	IPLAUD	-	IPLS	IPLT1	IPLX1	IPLT0	IPLX0
IPH1	B3h	Interrupt Priority Control High 1	-	IPHUSB	-	IPHKB	IPHADC	IPHSPI	IPHI2C	IPHMMC
IPL1	B2h	Interrupt Priority Control Low 1	-	IPLUSB	-	IPLKB	IPLADC	IPLSPI	IPLI2C	IPLMMC

Table 19. Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P0	80h	8-bit Port 0								
P1	90h	8-bit Port 1								
P2	A0h	8-bit Port 2								
P3	B0h	8-bit Port 3								
P4	C0h	8-bit Port 4								
P5	D8h	4-bit Port 5	-	-	-	-				

Table 20. Flash Memory SFR

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
FCON	D1h	Flash Control	FPL3	FPL2	FPL1	FPL0	FPS	FMOD1	FMOD0	FBUSY

Table 21. Timer SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
TCON	88h	Timer/Counter 0 and 1 Control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89h	Timer/Counter 0 and 1 Modes	GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00
TL0	8Ah	Timer/Counter 0 Low Byte								
TH0	8Ch	Timer/Counter 0 High Byte								
TL1	8Bh	Timer/Counter 1 Low Byte								
TH1	8Dh	Timer/Counter 1 High Byte								
WDTRST	A6h	WatchDog Timer Reset								
WDTPRG	A7h	WatchDog Timer Program	-	-	-	-	-	WTO2	WTO1	WTO0

Table 22. Audio Interface SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
AUDCON0	9Ah	Audio Control 0	JUST4	JUST3	JUST2	JUST1	JUST0	POL	DSIZ	HLR
AUDCON1	9Bh	Audio Control 1	SRC	DRQEN	MSREQ	MUDRN	-	DUP1	DUP0	AUDEN
AUDSTA	9Ch	Audio Status	SREQ	UDRN	AUBUSY	-	-	-	-	-
AUDDAT	9Dh	Audio Data	AUD7	AUD6	AUD5	AUD4	AUD3	AUD2	AUD1	AUD0
AUDCLK	ECh	Audio Clock Divider	-	-	-	AUCD4	AUCD3	AUCD2	AUCD1	AUCD0

Table 23. USB Controller SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
USBCON	BCh	USB Global Control	USBE	SUSPCLK	SDRMWUP	-	UPRSM	RMWUPE	CONFG	FADDEN
USBADDR	C6h	USB Address	FEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
USBINT	BDh	USB Global Interrupt	-	-	WUPCPU	EORINT	SOFINT	-	-	SPINT
USBIEN	BEh	USB Global Interrupt Enable	-	-	EWUPCPU	EEORINT	ESOFINT	-	-	ESPINT
UEPNUM	C7h	USB Endpoint Number	-	-	-	-	-	-	EPNUM1	EPNUM0
UEPCONX	D4h	USB Endpoint X Control	EPEN	-	-	-	DTGL	EPDIR	EPTYPE1	EPTYPE0
UEPSTAX	CEh	USB Endpoint X Status	DIR	-	STALLRQ	TXRDY	STLCRC	RXSETUP	RXOUT	TXCMP
UEPRST	D5h	USB Endpoint Reset	-	-	-	-	EP3RST	EP2RST	EP1RST	EP0RST
UEPINT	F8h	USB Endpoint Interrupt	-	-	-	-	EP3INT	EP2INT	EP1INT	EP0INT
UEPIEN	C2h	USB Endpoint Interrupt Enable	-	-	-	-	EP3INTE	EP2INTE	EP1INTE	EP0INTE
UEPDATX	CFh	USB Endpoint X FIFO Data	FDAT7	FDAT6	FDAT5	FDAT4	FDAT3	FDAT2	FDAT1	FDAT0
UBYCTX	E2h	USB Endpoint X Byte Counter	–	BYCT6	BYCT5	BYCT4	BYCT3	BYCT2	BYCT1	BYCT0
UFNUML	BAh	USB Frame Number Low	FNUM7	FNUM6	FNUM5	FNUM4	FNUM3	FNUM2	FNUM1	FNUM0
UFNUMH	BBh	USB Frame Number High	-	-	CRCOK	CRCERR	-	FNUM10	FNUM9	FNUM8
USBCLK	EAh	USB Clock Divider	-	-	-	-	-	-	USBCD1	USBCD0

Table 24. MMC Controller SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
MMCON0	E4h	MMC Control 0	DRPTR	DTPTR	CRPTR	CTPTR	MBLOCK	DFMT	RFMT	CRCDIS
MMCON1	E5h	MMC Control 1	BLEN3	BLEN2	BLEN1	BLEN0	DATDIR	DATEN	RESPEN	CMDEN
MMCON2	E6h	MMC Control 2	MMCEN	DCR	CCR	-	-	DATD1	DATD0	FLOWC
MMSTA	DEh	MMC Control and Status	-	-	CBUSY	CRC16S	DATFS	CRC7S	RESPFS	CFLCK
MMINT	E7h	MMC Interrupt	MCBI	EORI	EOCI	EOFI	F2FI	F1FI	F2EI	F1EI
MMMSK	DFh	MMC Interrupt Mask	MCBM	EORM	EOCM	EOFM	F2FM	F1FM	F2EM	F1EM
MMCMD	DDh	MMC Command	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
MMDAT	DCh	MMC Data	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
MMCLK	EDh	MMC Clock Divider	MMCD7	MMCD6	MMCD5	MMCD4	MMCD3	MMCD2	MMCD1	MMCD0

Table 25. IDE Interface SFR

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
DAT16H	F9h	High Order Data Byte	D15	D14	D13	D12	D11	D10	D9	D8

Table 26. Serial I/O Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SCON	98h	Serial Control	FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF	99h	Serial Data Buffer								
SADEN	B9h	Slave Address Mask								
SADDR	A9h	Slave Address								
BDRCON	92h	Baud Rate Control	-	-	-	BRR	TBCK	RBCK	SPD	SRC
BRL	91h	Baud Rate Reload								

Table 27. SPI Controller SFRs

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

Table 28. Special Register

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

Table 29. Keyboard Interface SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
KBCON	A3h	Keyboard Control	KINL3	KINL2	KINL1	KINL0	KINM3	KINM2	KINM1	KINM0
KBSTA	A4h	Keyboard Status	KPDE	-	-	-	KINF3	KINF2	KINF1	KINF0

Table 30. A/D Controller SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
ADCON	F3h	ADC Control	-	ADIDL	ADEN	ADEOC	ADSST	-	-	ADCS
ADCLK	F2h	ADC Clock Divider	-	-	-	ADCD4	ADCD3	ADCD2	ADCD1	ADCD0
ADDL	F4h	ADC Data Low Byte	-	-	-	-	-	-	ADAT1	ADAT0
ADDH	F5h	ADC Data High Byte	ADAT9	ADAT8	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2

Table 31. SFR Addresses and Reset Values

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	UEPINT 0000 0000	DAT16H XXXX XXXX		NVERS(2) XXXX XXXX					FFh
F0h	B(1) 0000 0000		ADCLK 0000 0000	ADCON 0000 0000	ADDL 0000 0000	ADDH 0000 0000			F7h
E8h		PLLCON 0000 1000	USBCLK 0000 0000		AUDCLK 0000 0000	MMCLK 0000 0000	PLLNDIV 0000 0000	PLLRDIV 0000 0000	EFh
E0h	ACC(1) 0000 0000		UBYCTLX 0000 0000		MMCON0 0000 0000	MMCON1 0000 0000	MMCON2 0000 0000	MMINT 0000 0011	E7h
D8h	P5(1) XXXX 1111				MMDAT 1111 1111	MMCMD 1111 1111	MMSTA 0000 0000	MMMSK 1111 1111	DFh
D0h	PSW(1) 0000 0000	FCON(3) 1111 0000(4)			UEPCONX 0000 0000	UEPRST 0000 0000			D7h
C8h							UEPSTAX 0000 0000	UEPDATX 0000 0000	CFh
C0h	P4(1) 1111 1111		UEPIEN 0000 0000	SPCON 0001 0100	SPSTA 0000 0000	SPDAT XXXX XXXX	USBADDR 1000 0000	UEPNUM 0000 0000	C7h
B8h	IPL0(1) X000 0000	SADEN 0000 0000	UFNUML 0000 0000	UFNUMH 0000 0000	USBCON 0000 0000	USBINT 0000 0000	USBIEN 0001 0000		BFh
B0h	P3(1) 1111 1111	IEN1 0000 0000	IPL1 0000 0000	IPH1 0000 0000				IPH0 X000 0000	B7h
A8h	IEN0(1) 0000 0000	SADDR 0000 0000							AFh
A0h	P2(1) 1111 1111		AUXR1 XXXX 00X0	KBCON 0000 1111	KBSTA 0000 0000		WDTRST XXX XXXX	WDTPRG XXXX X000	A7h
98h	SCON 0000 0000	SBUF XXXX XXXX	AUDCON0 0000 1000	AUDCON1 1011 0010	AUDSTA 1100 0000	AUDDAT 1111 1111			9Fh
90h	P1(1) 1111 1111	BRL 0000 0000	BDRCON XXX0 0000	SSCON 0000 0000	SSSTA 1111 1000	SSDAT 1111 1111	SSADR 1111 1110		97h
88h	TCON(1) 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR X000 1101	CKCON 0000 000X(5)	8Fh
80h	P0(1) 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 00XX 0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Reserved

Notes:
1. SFR registers with least significant nibble address equal to 0 or 8 are bit-addressable.
2. NVERS reset value depends on the silicon version: 1000 0011 for AT89C5132 product
3. FCON register is only available in AT89C5132 product.
4. FCON reset value is 00h in case of reset with hardware condition.
5. CKCON reset value depends on the X2B bit (programmed or unprogrammed) in the Hardware Byte.

10. Interrupt System

The AT89C5132, like other control-oriented computer architectures, employ a program interrupt method. This operation branches to a subroutine and performs some service in response to the interrupt. When the subroutine terminates, execution resumes at the point where the interrupt occurred. Interrupts may occur as a result of internal AT89C5132 activity (e.g., timer overflow) or at the initiation of electrical signals external to the microcontroller (e.g., keyboard). In all cases, interrupt operation is programmed by the system designer, who determines priority of interrupt service relative to normal code execution and other interrupt service routines. All of the interrupt sources are enabled or disabled by the system designer and may be manipulated dynamically.

A typical interrupt event chain occurs as follows:

1. An internal or external device initiates an interrupt-request signal. The AT89C5132, latch this event into a flag buffer.
2. The priority of the flag is compared to the priority of other interrupts by the interrupt handler. A high priority causes the handler to set an interrupt flag.
3. This signals the instruction execution unit to execute a context switch. This context switch breaks the current flow of instruction sequences. The execution unit completes the current instruction prior to a save of the program counter (PC) and reloads the PC with the start address of a software service routine.
4. The software service routine executes assigned tasks and as a final activity performs a RETI (return from interrupt) instruction. This instruction signals completion of the interrupt, resets the interrupt-in-progress priority and reloads the program counter. Program operation then continues from the original point of interruption.

Table 32. Interrupt System Signals

Signal Name	Type	Description	Alternate Function
$\overline{INT0}$	I	**External Interrupt 0** See Section "External Interrupts", page 37.	P3.2
$\overline{INT1}$	I	**External Interrupt 1** See Section “External Interrupts”, page 37.	P3.3
KIN3:0	I	**Keyboard Interrupt Inputs** See Section “Keyboard Interface”, page 152.	P1.3:0

Six interrupt registers are used to control the interrupt system. Two 8-bit registers are used to enable separately the interrupt sources: IEN0 and IEN1 registers (see Table 35 and Table 36).

Four 8-bit registers are used to establish the priority level of the sources: IPH0, IPL0, IPH1 and IPL1 registers (see Table 10-1 to Table 39).

10.1 Interrupt System Priorities

Each of the interrupt sources on the AT89C5132 can be individually programmed to one of four priority levels. This is accomplished by one bit in the Interrupt Priority High registers (IPH0 and IPH1) and one bit in the Interrupt Priority Low registers (IPL0 and IPL1). This provides each interrupt source four possible priority levels according to Table 33.

Table 33. Priority Levels

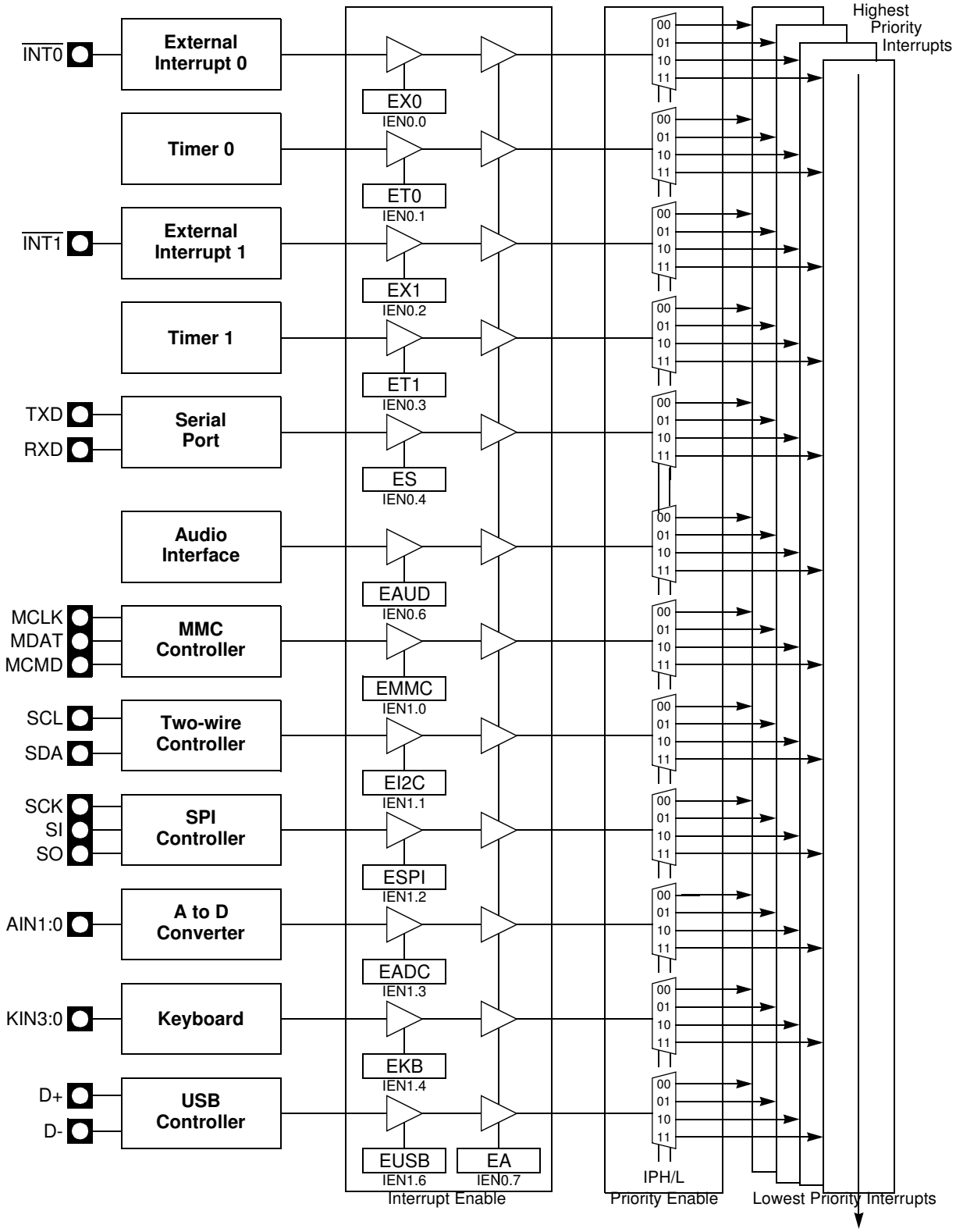
IPHxx	IPLxx	Priority Level
0	0	0 Lowest
0	1	1
1	0	2
1	1	3 Highest

A low-priority interrupt is always interrupted by a higher priority interrupt but not by another interrupt of lower or equal priority. Higher priority interrupts are serviced before lower priority interrupts. The response to simultaneous occurrence of equal priority interrupts is determined by an internal hardware polling sequence detailed in Table 34. Thus within each priority level there is a second priority structure determined by the polling sequence. The interrupt control system is shown in Figure 10-1.

Table 34. Priority Within Same Level

Interrupt Name	Priority Number	Interrupt Address Vectors	Interrupt Request Flag Cleared by Hardware (H) or by Software (S)
$\overline{\text{INT0}}$	0 (Highest Priority)	C:0003h	H if edge, S if level
Timer 0	1	C:000Bh	H
$\overline{\text{INT1}}$	2	C:0013h	H if edge, S if level
Timer 1	3	C:001Bh	H
Serial Port	4	C:0023h	S
Reserved	5		
Audio Interface	6	C:0033h	S
MMC Interface	7	C:003Bh	S
Two-wire Controller	8	C:0043h	S
SPI Controller	9	C:004Bh	S
A-to-D Converter	10	C:0053h	S
Keyboard	11	C:005Bh	S
Reserved	12	C:0063h	-
USB	13	C:006Bh	S
Reserved	14 (Lowest Priority)	C:0073h	-

Figure 10-1. Interrupt Control System



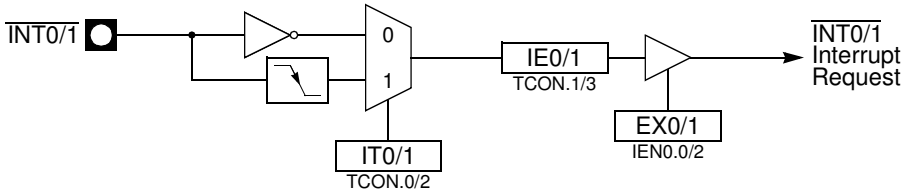
10.2 External Interrupts

10.2.1 $\overline{\text{INT1:0}}$ Inputs

External interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ ($\overline{\text{INTn}}$, n = 0 or 1) pins may each be programmed to be level-triggered or edge-triggered, dependent upon bits IT0 and IT1 (ITn, n = 0 or 1) in TCON register as shown in Figure 10-2. If ITn = 0, $\overline{\text{INTn}}$ is triggered by a low level at the pin. If ITn = 1, $\overline{\text{INTn}}$ is negative-edge triggered. External interrupts are enabled with bits EX0 and EX1 (EXn, n = 0 or 1) in IEN0. Events on $\overline{\text{INTn}}$ set the interrupt request flag IEn in TCON register. If the interrupt is edge-triggered, the request flag is cleared by hardware when vectoring to the interrupt service routine. If the interrupt is level-triggered, the interrupt service routine must clear the request flag and the interrupt must be deasserted before the end of the interrupt service routine.

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ inputs provide both the capability to exit from Power-down mode on low level signals as detailed in Section “Exiting Power-down Mode”, page 47.

Figure 10-2. INT1:0 Input Circuitry



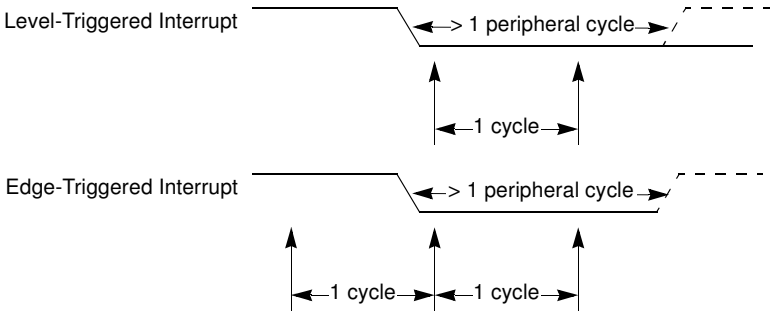
10.2.2 KIN3:0 Inputs

External interrupts KIN0 to KIN3 provide the capability to connect a matrix keyboard. For detailed information on these inputs, refer to Section “Keyboard Interface”, page 152.

10.2.3 Input Sampling

External interrupt pins ($\overline{\text{INT1:0}}$ and KIN3:0) are sampled once per peripheral cycle (6 peripheral clock periods) (see Figure 10-3). A level-triggered interrupt pin held low or high for more than 6 peripheral clock periods (12 oscillator in standard mode or 6 oscillator clock periods in X2 mode) guarantees detection. Edge-triggered external interrupts must hold the request pin low for at least 6 peripheral clock periods.

Figure 10-3. Minimum Pulse Timings



10.3 Registers

Table 35. IEN0 Register
IEN0 (S:A8h) – Interrupt Enable Register 0

7	6	5	4	3	2	1	0
EA	**EAUD**	**–**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

Bit Number	Bit Mnemonic	Description
7	EA	**Enable All Interrupt Bit** Set to enable all interrupts. Clear to disable all interrupts. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its interrupt enable bit.
6	EAUD	**Audio Interface Interrupt Enable Bit** Set to enable audio interface interrupt. Clear to disable audio interface interrupt.
5	–	**Reserved** The values read from this bit is always 0. Do not set this bit.
4	ES	**Serial Port Interrupt Enable Bit** Set to enable serial port interrupt. Clear to disable serial port interrupt.
3	ET1	**Timer 1 Overflow Interrupt Enable Bit** Set to enable Timer 1 overflow interrupt. Clear to disable Timer 1 overflow interrupt.
2	EX1	**External Interrupt 1 Enable bit** Set to enable external interrupt 1. Clear to disable external interrupt 1.
1	ET0	**Timer 0 Overflow Interrupt Enable Bit** Set to enable timer 0 overflow interrupt. Clear to disable timer 0 overflow interrupt.
0	EX0	**External Interrupt 0 Enable Bit** Set to enable external interrupt 0. Clear to disable external interrupt 0.

Reset Value = 0000 0000b

Table 36. IEN1 Register
IEN1 (S:B1h) – Interrupt Enable Register 1

7	6	5	4	3	2	1	0
-	**EUSB**	**–**	**EKB**	**EADC**	**ESPI**	**EI2C**	**EMMC**

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
6	EUSB	**USB Interface Interrupt Enable Bit** Set this bit to enable **USB** interrupts. Clear this bit to disable **USB** interrupts.
5	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
4	EKB	**Keyboard Interface Interrupt Enable Bit** Set to enable Keyboard interrupt. Clear to disable Keyboard interrupt.
3	EADC	**A to D Converter Interrupt Enable Bit** Set to enable ADC interrupt. Clear to disable ADC interrupt.
2	ESPI	**SPI Controller Interrupt Enable Bit** Set to enable SPI interrupt. Clear to disable SPI interrupt.
1	EI2C	**Two Wire Controller Interrupt Enable Bit** Set to enable Two Wire interrupt. Clear to disable Two Wire interrupt.
0	EMMC	**MMC Interface Interrupt Enable Bit** Set to enable MMC interrupt. Clear to disable MMC interrupt.

Reset Value = 0000 0000b

Table 10-1. IPH0 Register

IPH0 (S:B7h) – Interrupt Priority High Register 0

7	6	5	4	3	2	1	0
-	IPHAUD	–	IPHS	IPHT1	IPHX1	IPHT0	IPHX0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPHAUD	**Audio Interface Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
5	IPHMP3	**MP3 Decoder Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
4	IPHS	**Serial Port Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
3	IPHT1	**Timer 1 Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
2	IPHX1	**External Interrupt 1 Priority Level MSB** Refer to Table 33 for priority level description.
1	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	IPHX0	**External Interrupt 0 Priority Level MSB** Refer to Table 33 for priority level description.

Reset Value = X000 0000b

Table 37. IPH1 Register

IPH1 (S:B3h) – Interrupt Priority High Register 1

7	6	5	4	3	2	1	0
-	IPHUSB	–	IPHKB	IPHADC	IPHSPI	IPHI2C	IPHMMC

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
6	IPHUSB	**USB Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
5	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
4	IPHKB	**Keyboard Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
3	IPHADC	**A to D Converter Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
2	IPHSPI	**SPI Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
1	IPHI2C	**Two Wire Controller Interrupt Priority Level MSB** Refer to Table 33 for priority level description.
0	IPHMMC	**MMC Interrupt Priority Level MSB** Refer to Table 33 for priority level description.

Reset Value = 0000 0000b

Table 38. IPL0 Register

IPL0 (S:B8h) – Interrupt Priority Low Register 0

7	6	5	4	3	2	1	0
-	IPLAUD	–	IPLS	IPLT1	IPLX1	IPLT0	IPLX0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPLAUD	**Audio Interface Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
5	IPLMP3	**MP3 Decoder Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
4	IPLS	**Serial Port Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
3	IPLT1	**Timer 1 Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
2	IPLX1	**External Interrupt 1 Priority Level LSB** Refer to Table 33 for priority level description.
1	IPLT0	**Timer 0 Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
0	IPLX0	**External Interrupt 0 Priority Level LSB** Refer to Table 33 for priority level description.

Reset Value = X000 0000b

Table 39. IPL1 Register

IPL1 (S:B2h) – Interrupt Priority Low Register 1

7	6	5	4	3	2	1	0
-	IPLUSB	-	IPLKB	IPLADC	IPLSPI	IPLI2C	IPLMMC

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
6	IPLUSB	**USB Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
5	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
4	IPLKB	**Keyboard Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
3	IPLADC	**A to D Converter Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
2	IPLSPI	**SPI Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
1	IPLI2C	**Two Wire Controller Interrupt Priority Level LSB** Refer to Table 33 for priority level description.
0	IPLMMC	**MMC Interrupt Priority Level LSB** Refer to Table 33 for priority level description.

Reset Value = 0000 0000b

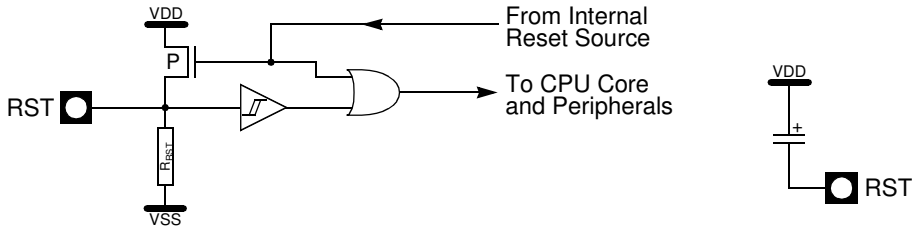
11. Power Management

2 power reduction modes are implemented in the AT89C5132: the Idle mode and the Power-down mode. These modes are detailed in the following sections. In addition to these power reduction modes, the clocks of the core and peripherals can be dynamically divided by 2 using the X2 mode detailed in Section “X2 Feature”, page 12.

11.1 Reset

In order to start-up (cold reset) or to restart (warm reset) properly the microcontroller, an high level has to be applied on the RST pin. A bad level leads to a wrong initialization of the internal registers like SFRs, Program Counter… and to unpredictable behavior of the microcontroller. A proper device reset initializes the AT89C5132 and vectors the CPU to address 0000h. RST input has a pull-down resistor allowing power-on reset by simply connecting an external capacitor to V_{DD} as shown in Figure 11-1. A warm reset can be applied either directly on the RST pin or indirectly by an internal reset source such as the watchdog timer. Resistor value and input characteristics are discussed in the Section “DC Characteristics” of the AT89C5132 datasheet. The status of the Port pins during reset is detailed in Table 16.

Figure 11-1. Reset Circuitry and Power-On Reset



RST input circuitry

Power-on Reset

Table 16. Pin Conditions in Special Operating Modes

Mode	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	MMC	Audio
Reset	Floating	High	High	High	High	High	Floating	[1]
Idle	Data	Data	Data	Data	Data	Data	Data	Data
Power-down	Data	Data	Data	Data	Data	Data	Data	Data

Note: 1. Refer to Section “Audio Output Interface”, page 75.

11.1.1 Cold Reset

2 conditions are required before enabling a CPU start-up:

- V_{DD} must reach the specified V_{DD} range
- The level on X1 input pin must be outside the specification (V_{IH}, V_{IL})

If one of these 2 conditions are not met, the microcontroller does not start correctly and can execute an instruction fetch from anywhere in the program space. An active level applied on the RST pin must be maintained till both of the above conditions are met. A reset is active when the level V_{IH1} is reached and when the pulse width covers the period of time where V_{DD} and the oscillator are not stabilized. 2 parameters have to be taken into account to determine the reset pulse width:

- V_{DD} rise time,
- Oscillator startup time.

To determine the capacitor value to implement, the highest value of these 2 parameters has to be chosen. Table 17 gives some capacitor values examples for a minimum R_{RST} of 50 KΩ and different oscillator startup and V_{DD} rise times.

Table 17. Minimum Reset Capacitor Value for a 50 kΩ Pull-down Resistor(1)

Oscillator Start-Up Time	VDD Rise Time		
	1 ms	10 ms	100 ms
5 ms	820 nF	1.2 µF	12 µF
20 ms	2.7 µF	3.9 µF	12 µF

Note: 1. These values assume V_{DD} starts from 0V to the nominal value. If the time between 2 on/off sequences is too fast, the power-supply de-coupling capacitors may not be fully discharged, leading to a bad reset sequence.

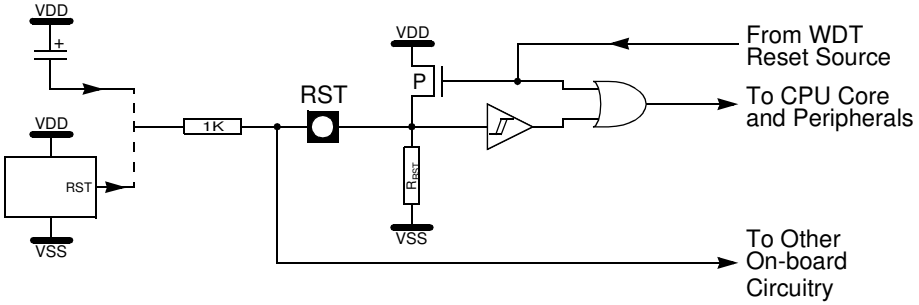
11.1.2 Warm Reset

To achieve a valid reset, the reset signal must be maintained for at least 2 machine cycles (24 oscillator clock periods) while the oscillator is running. The number of clock periods is mode independent (X2 or X1).

11.1.3 Watchdog Reset

As detailed in Section “Watchdog Timer”, page 61, the WDT generates a 96-clock period pulse on the RST pin. In order to properly propagate this pulse to the rest of the application in case of external capacitor or power-supply supervisor circuit, a 1 kΩ resistor must be added as shown in Figure 11-2.

Figure 11-2. Reset Circuitry for WDT Reset-out Usage



11.2 Reset Recommendation to Prevent Flash Corruption

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

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

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

11.3 Idle Mode

Idle mode is a power reduction mode that reduces the power consumption. In this mode, program execution halts. Idle mode freezes the clock to the CPU at known states while the peripherals continue to be clocked (refer to Section “Oscillator”, page 12). The CPU status before entering Idle mode is preserved, i.e., the program counter and program status word register retain their data for the duration of Idle mode. The contents of the SFRs and RAM are also retained. The status of the Port pins during Idle mode is detailed in Table 16.

11.3.1 Entering Idle Mode

To enter Idle mode, the user must set the IDL bit in PCON register (see Table 18). The AT89C5132 enters Idle mode upon execution of the instruction that sets IDL bit. The instruction that sets IDL bit is the last instruction executed.

Note: If IDL bit and PD bit are set simultaneously, the AT89C5132 enter Power-down mode. Then it does not go in Idle mode when exiting Power-down mode.

11.3.2 Exiting Idle Mode

There are 2 ways to exit Idle mode:

1. Generate an enabled interrupt.
   - Hardware clears IDL bit in PCON register which restores the clock to the CPU. Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Idle mode. The general-purpose flags (GF1 and GF0 in PCON register) may be used to indicate whether an interrupt occurred during normal operation or during Idle mode. When Idle mode is exited by an interrupt, the interrupt service routine may examine GF1 and GF0.
2. Generate a reset.
   - A logic high on the RST pin clears IDL bit in PCON register directly and asynchronously. This restores the clock to the CPU. Program execution momentarily resumes with the instruction immediately following the instruction that activated the Idle mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the AT89C5132 and vectors the CPU to address C:0000h.

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

11.4 Power-down Mode

The Power-down mode places the AT89C5132 in a very low power state. Power-down mode stops the oscillator and freezes all clocks at known states (refer to the Section "Oscillator", page 12). The CPU status prior to entering Power-down mode is preserved, i.e., the program counter, program status word register retain their data for the duration of Power-down mode. In addition, the SFRs and RAM contents are preserved. The status of the Port pins during Power-down mode is detailed in Table 16.

Note: V_{DD} may be reduced to as low as V_{RET} during Power-down mode to further reduce power dissipation. Notice, however, that V_{DD} is not reduced until Power-down mode is invoked.

11.4.1 Entering Power-down Mode

To enter Power-down mode, set PD bit in PCON register. The AT89C5132 enters the Power-down mode upon execution of the instruction that sets PD bit. The instruction that sets PD bit is the last instruction executed.

11.4.2 Exiting Power-down Mode

If V_{DD} was reduced during the Power-down mode, do not exit Power-down mode until V_{DD} is restored to the normal operating level.

There are 2 ways to exit the Power-down mode:

1. Generate an enabled external interrupt.
   - The AT89C5132 provides capability to exit from Power-down using $\overline{INT0}$, $\overline{INT1}$, and KIN3:0 inputs. In addition, using KIN input provides high or low level exit capability (see Section “Keyboard Interface”, page 181).
     Hardware clears PD bit in PCON register which starts the oscillator and restores the clocks to the CPU and peripherals. Using $\overline{INTn}$ input, execution resumes when the input is released (see Figure 11-3) while using KINx input, execution resumes after counting 1024 clock ensuring the oscillator is restarted properly (see Figure 11-4). This behavior is necessary for decoding the key while it is still pressed. In both cases, execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power-down mode.

Note: 1. The external interrupt used to exit Power-down mode must be configured as level sensitive ($\overline{INT0}$ and $\overline{INT1}$) and must be assigned the highest priority. In addition, the duration of the interrupt must be long enough to allow the oscillator to stabilize. The execution will only resume when the interrupt is deasserted.

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

Figure 11-3. Power-down Exit Waveform Using $\overline{INT1:0}$

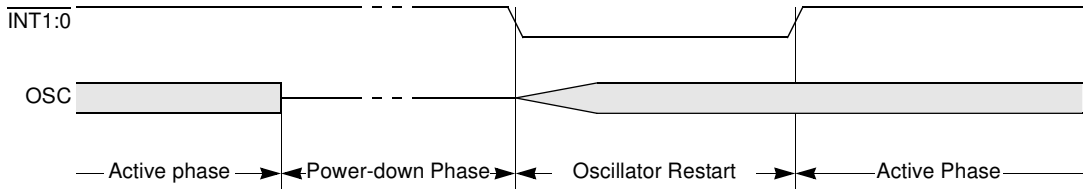
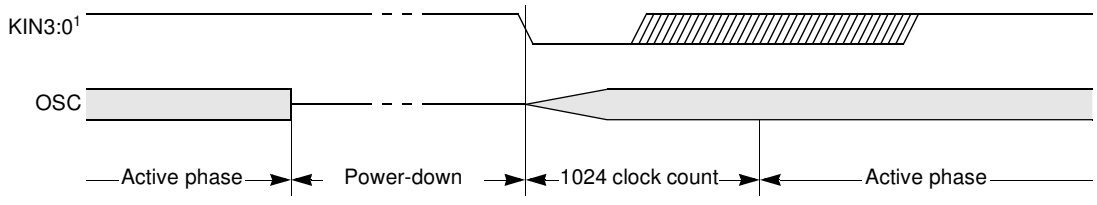


Figure 11-4. Power-down Exit Waveform Using KIN3:0



Note: 1. KIN3:0 can be high or low-level triggered.

2. Generate a reset.

– A logic high on the RST pin clears PD bit in PCON register directly and asynchronously. This starts the oscillator and restores the clock to the CPU and peripherals. Program execution momentarily resumes with the instruction immediately following the instruction that activated Power-down mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the AT89C5132 and vectors the CPU to address 0000h.

Notes: 1. During the time that execution resumes, the internal RAM cannot be accessed; however, it is possible for the Port pins to be accessed. To avoid unexpected outputs at the Port pins, the instruction immediately following the instruction that activated the Power-down mode should not write to a Port pin or to the external RAM.
2. Exit from power-down by reset redefines all the SFRs, but does not affect the internal RAM content.

11.5 Registers

Table 18. PCON Register

PCON (S:87h) – Power Configuration Register

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

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial Port Mode Bit 1** Set to select double baud rate in mode 1,2 or 3.
6	SMOD0	**Serial Port Mode Bit 0** Set to select FE bit in SCON register. Clear to select SM0 bit in SCON register.
5 - 4	-	**Reserved** The value read from these bits is indeterminate. Do not set these bits.
3	GF1	**General-Purpose Flag 1** One use is to indicate whether an interrupt occurred during normal operation or during Idle mode.
2	GF0	**General-Purpose Flag 0** One use is to indicate whether an interrupt occurred during normal operation or during Idle mode.
1	PD	**Power-Down Mode Bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Power-down mode. If IDL and PD are both set, PD takes precedence.
0	IDL	**Idle Mode Bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Idle mode. If IDL and PD are both set, PD takes precedence.

Reset Value = 00XX 0000b

12. Timers/Counters

The AT89C5132 implement two general-purpose, 16-bit Timers/Counters. They are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a Timer or as an event Counter. When operating as a Timer, the Timer/Counter runs for a programmed length of time, then issues an interrupt request. When operating as a Counter, the Timer/Counter counts negative transitions on an external pin. After a preset number of counts, the Counter issues an interrupt request.

The various operating modes of each Timer/Counter are described in the following sections.

12.1 Timer/Counter Operations

For instance, a basic operation is Timer registers THx and TLx (x = 0, 1) connected in cascade to form a 16-bit Timer. Setting the run control bit (TRx) in TCON register (see Table 40) turns the Timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the Timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx Timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the Timer/Counter is unpredictable.

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

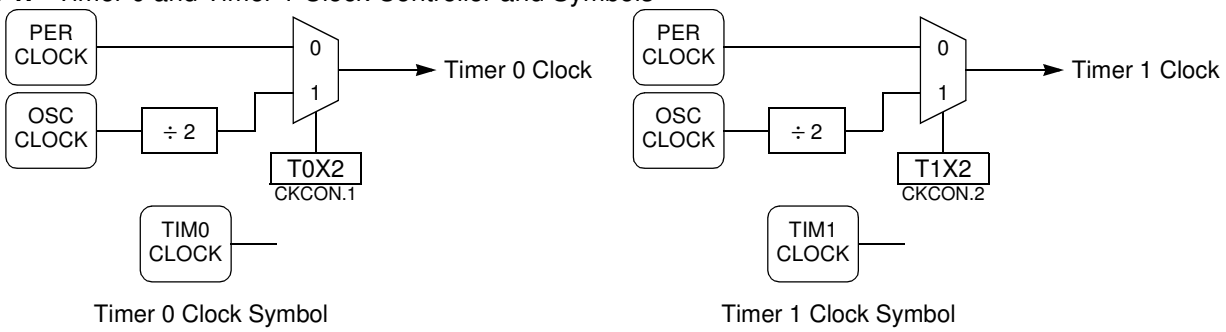
For Timer operation (C/Tx# = 0), the Timer register counts the divided-down peripheral clock. The Timer register is incremented once every peripheral cycle (6 peripheral clock periods). The Timer clock rate is $F_{PER}/6$, i.e., $F_{OSC}/12$ in standard mode or $F_{OSC}/6$ in X2 mode.

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

12.2 Timer Clock Controller

As shown in Figure 12-1, the Timer 0 (FT0) and Timer 1 (FT1) clocks are derived from either the peripheral clock (F_{PER}) or the oscillator clock (F_{OSC}) depending on the T0X2 and T1X2 Bits in CKCON register. These clocks are issued from the Clock Controller block as detailed in Section 'CKCON Register', page 14. When T0X2 or T1X2 bit is set, the Timer 0 or Timer 1 clock frequency is fixed and equal to the oscillator clock frequency divided by 2. When cleared, the Timer clock frequency is equal to the oscillator clock frequency divided by 2 in standard mode or to the oscillator clock frequency in X2 mode.

Figure 12-1. Timer 0 and Timer 1 Clock Controller and Symbols



12.3 Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 12-2 through Figure 12-8 show the logical configuration of each mode.

Timer 0 is controlled by the four lower Bits of TMOD register (see Table 41) and Bits 0, 1, 4 and 5 of TCON register (see Table 40). TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (C/T0#) and mode of operation (M10 and M00). TCON register provides Timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).

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

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

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

12.3.1 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 TL0 register (see Figure 12-2). The upper three Bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments TH0 register. Figure 12-3 gives the overflow period calculation formula.

Figure 12-2. Timer/Counter x (x = 0 or 1) in Mode 0

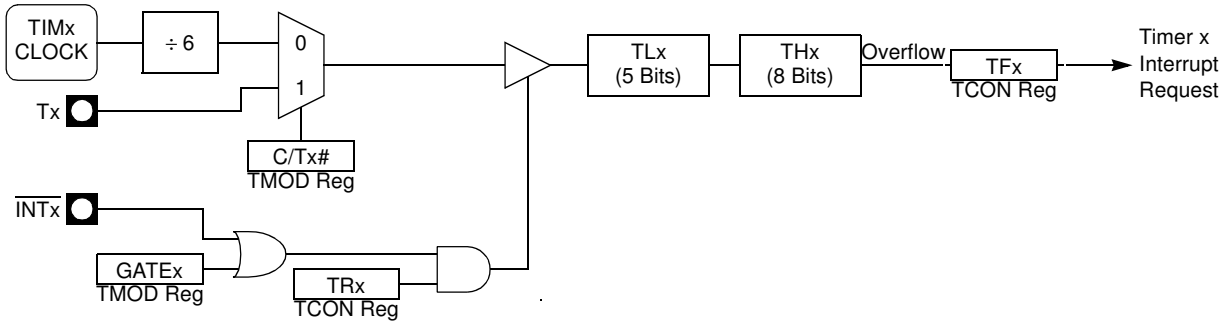


Figure 12-3. Mode 0 Overflow Period Formula

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

12.3.2 Mode 1 (16-bit Timer)

Mode 1 configures Timer 0 as a 16-bit Timer with TH0 and TL0 registers connected in cascade (see Figure 12-4). The selected input increments TL0 register. Figure 12-5 gives the overflow period calculation formula when in timer mode.

Figure 12-4. Timer/Counter x (x = 0 or 1) in Mode 1

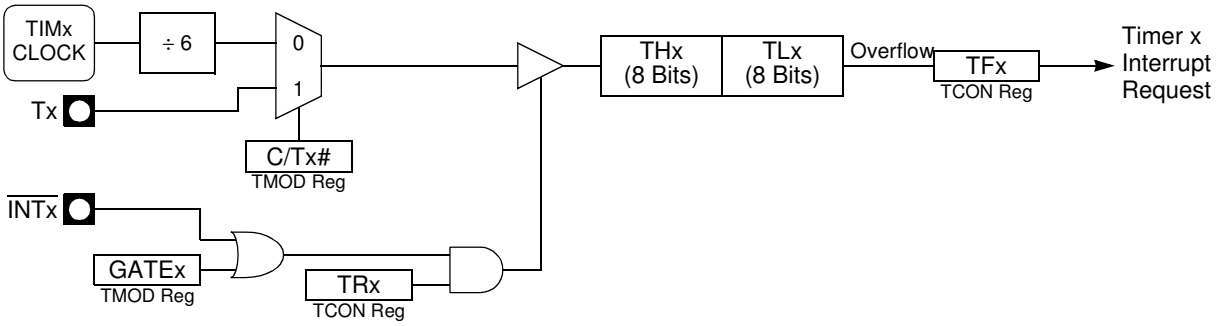


Figure 12-5. Mode 1 Overflow Period Formula

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

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

Mode 2 configures Timer 0 as an 8-bit Timer (TL0 register) that automatically reloads from TH0 register (see Table 42). TL0 overflow sets TF0 flag in TCON register and reloads TL0 with the contents of TH0, which is preset by software. When the interrupt request is serviced, hardware clears TF0. The reload leaves TH0 unchanged. The next reload value may be changed at any time by writing it to TH0 register. Figure 12-7 gives the autoreload period calculation formula when in timer mode.

Figure 12-6. Timer/Counter x (x = 0 or 1) in Mode 2

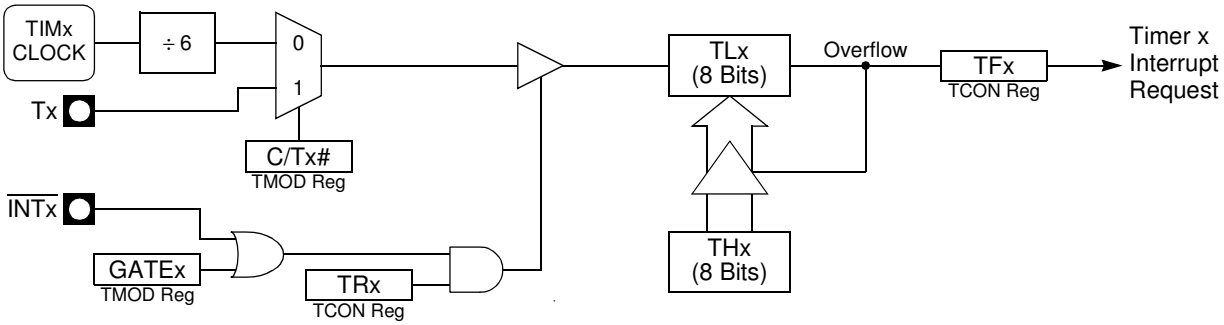


Figure 12-7. Mode 2 Autoreload Period Formula

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

12.3.4 Mode 3 (Two 8-bit Timers)

Mode 3 configures Timer 0 such that registers TL0 and TH0 operate as separate 8-bit Timers (see Figure 12-8). This mode is provided for applications requiring an additional 8-bit Timer or

Counter. TL0 uses the Timer 0 control Bits C/T0# and GATE0 in TMOD register, and TR0 and TF0 in TCON register in the normal manner. TH0 is locked into a Timer function (counting $F_{TF1}/6$) and takes over use of the Timer 1 interrupt (TF1) and run control (TR1) Bits. Thus, operation of Timer 1 is restricted when Timer 0 is in mode 3. Figure 12-7 gives the autoreload period calculation formulas for both TF0 and TF1 flags.

Figure 12-8. Timer/Counter 0 in Mode 3: Two 8-bit Counters

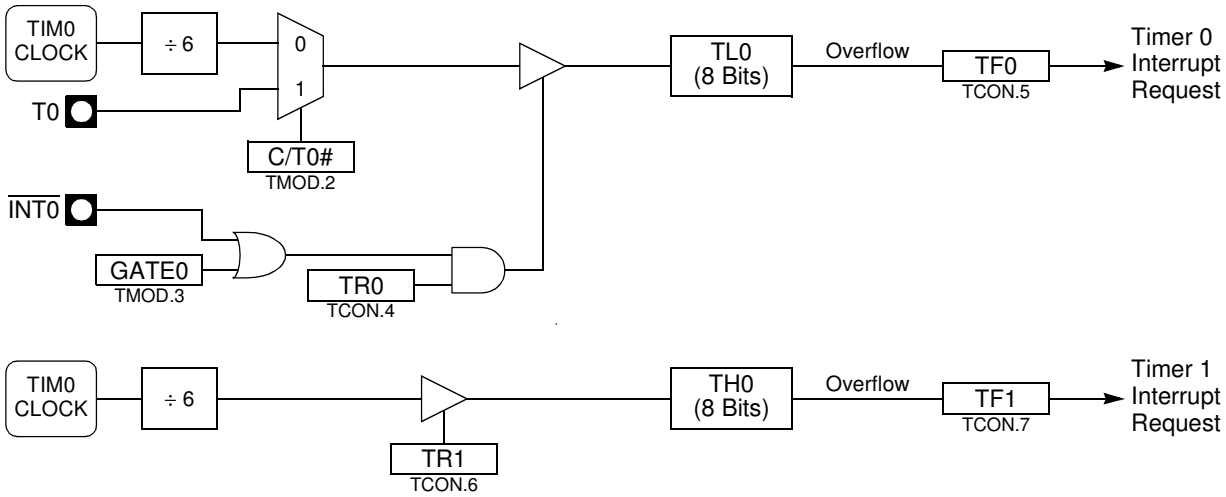


Figure 12-9. Mode 3 Overflow Period Formula

$$TF0_{PER} = \frac{6 \cdot (256 - TL0)}{F_{TIM0}} \qquad TF1_{PER} = \frac{6 \cdot (256 - TH0)}{F_{TIM0}}$$

12.4 Timer 1

Timer 1 is identical to Timer 0 excepted for Mode 3 which is a hold-count mode. Following comments help to understand the differences:

- Timer 1 functions as either a Timer or event Counter in three modes of operation. Figure 12-2 through Figure 12-6 show the logical configuration for modes 0, 1, and 2. Timer 1's mode 3 is a hold-count mode.
- Timer 1 is controlled by the four high-order Bits of TMOD register (see Table 41) and Bits 2, 3, 6 and 7 of TCON register (see Figure 40). TMOD register selects the method of Timer gating (GATE1), Timer or Counter operation (C/T1#) and mode of operation (M11 and M01). TCON register provides Timer 1 control functions: overflow flag (TF1), run control bit (TR1), interrupt flag (IE1) and interrupt type control bit (IT1).
- Timer 1 can serve as the Baud Rate Generator for the Serial Port. Mode 2 is best suited for this purpose.
- For normal Timer operation (GATE1 = 0), setting TR1 allows TL1 to be incremented by the selected input. Setting GATE1 and TR1 allows external pin $\overline{INT1}$ to control Timer operation.
- Timer 1 overflow (count rolls over from all 1s to all 0s) sets the TF1 flag generating an interrupt request.
- When Timer 0 is in mode 3, it uses Timer 1's overflow flag (TF1) and run control bit (TR1). For this situation, use Timer 1 only for applications that do not require an interrupt (such as a Baud Rate Generator for the Serial Port) and switch Timer 1 in and out of mode 3 to turn it off and on.
- It is important to stop the Timer/Counter before changing modes.

12.4.1 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 12-2). The upper 3 Bits of TL1 register are ignored. Prescaler overflow increments TH1 register.

12.4.2 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 12-4). The selected input increments TL1 register.

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

Mode 2 configures Timer 1 as an 8-bit Timer (TL1 register) with automatic reload from TH1 register on overflow (see Figure 12-6). TL1 overflow sets TF1 flag in TCON register and reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

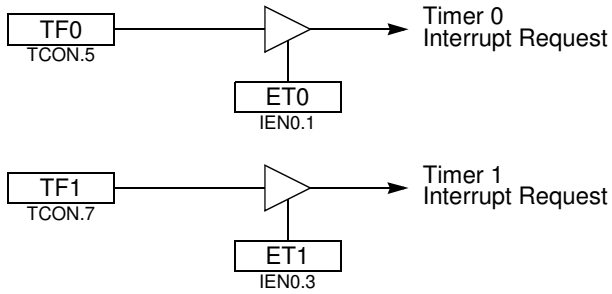
12.4.4 Mode 3 (Halt)

Placing Timer 1 in mode 3 causes it to halt and hold its count. This can be used to halt Timer 1 when TR1 run control bit is not available i.e. when Timer 0 is in mode 3.

12.5 Interrupt

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

Figure 12-10. Timer Interrupt System



12.6 Registers

Table 40. TCON Register

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

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

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

Reset Value = 0000 0000b

Table 41. TMOD Register

TMOD (89:h) - Timer/Counter 0 and 1 Modes

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

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

Reset Value = 0000 0000b

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

Table 42. TH0 Register

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

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

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

Reset Value = 0000 0000b

Table 43. TL0 Register

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

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

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

Reset Value = 0000 0000b

Table 44. TH1 Register

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

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

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

Reset Value = 0000 0000b

Table 45. TL1 Register

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

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

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

Reset Value = 0000 0000b

13. Watchdog Timer

The AT89C5132 implement a hardware Watchdog Timer (WDT) that automatically resets the chip if it is allowed to time out. The WDT provides a means of recovering from routines that do not complete successfully due to software or hardware malfunctions.

13.1 Description

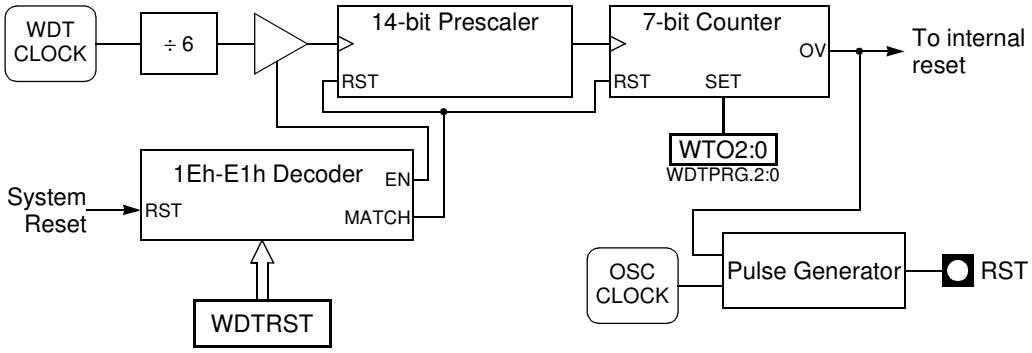
The WDT consists of a 14-bit prescaler followed by a 7-bit programmable counter. As shown in Figure 13-1, the 14-bit prescaler is fed by the WDT clock detailed in section "Watchdog Clock Controller", page 57.

The Watchdog Timer Reset register (WDTRST, see Table 47) provides control access to the WDT, while the Watchdog Timer Program register (WDTPRG, see Figure 48) provides time-out period programming.

Three operations control the WDT:

- Chip reset clears and disables the WDT.
- Programming the time-out value to the WDTPRG register.
- Writing a specific two-byte sequence to the WDTRST register clears and enables the WDT.

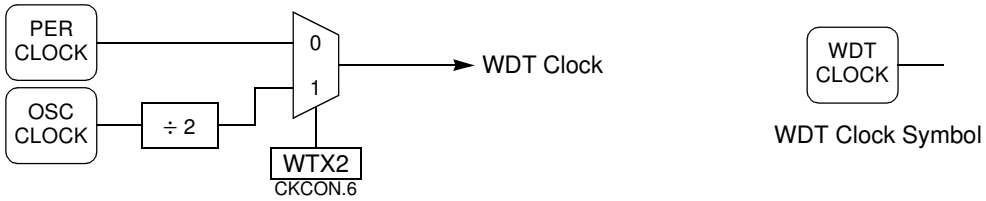
Figure 13-1. WDT Block Diagram



13.2 Watchdog Clock Controller

As shown in Figure 13-2 the WDT clock (F_{WDT}) is derived from either the peripheral clock (F_{PER}) or the oscillator clock (F_{OSC}) depending on the WTX2 bit in CKCON register. These clocks are issued from the Clock Controller block as detailed in section "Clock Controller", page 12. When WTX2 bit is set, the WDT clock frequency is fixed and equal to the oscillator clock frequency divided by 2. When cleared, the WDT clock frequency is equal to the oscillator clock frequency divided by 2 in standard mode or to the oscillator clock frequency in X2 mode.

Figure 13-2. WDT Clock Controller and Symbol



13.3 Watchdog Operation

After reset, the WDT is disabled. The WDT is enabled by writing the sequence 1Eh and E1h into the WDTRST register. As soon as it is enabled, there is no way except the chip reset to disable it. If it is not cleared using the previous sequence, the WDT overflows and forces a chip reset. This overflow generates a high level 96 oscillator periods pulse on the RST pin to globally reset the application. (refer to Section “Power Management”, page 48)

The WDT time-out period can be adjusted using WTO2:0 Bits located in the WDTPRG register accordingly to the formula shown in Figure 13-3. In this formula, WTOval represents the decimal value of WTO2:0 Bits. Table 48 reports the time-out period depending on the WDT frequency.

Figure 13-3. WDT Time-Out Formula

$$WDT_{TO} = \frac{6 \cdot (2^{14} \cdot 2^{WTOval})}{F_{WDT}}$$

Table 46. WDT Time-Out Computation

			F_{WDT} (ms)					
WTO2	**WTO1**	**WTO0**	**6 MHz(1)**	**8 MHz(1)**	**10 MHz(1)**	**12 MHz(2)**	**16 MHz(2)**	**20 MHz(2)**
0	0	0	16.38	12.28	9.83	8.19	6.14	4.92
0	0	1	32.77	24.57	19.66	16.38	12.28	9.83
0	1	0	65.54	49.14	39.32	32.77	24.57	19.66
0	1	1	131.07	98.28	78.64	65.54	49.14	39.32
1	0	0	262.14	196.56	157.29	131.07	98.28	78.64
1	0	1	524.29	393.1	314.57	262.14	196.56	157.29
1	1	0	1049	786.24	629.15	524.29	393.12	314.57
1	1	1	2097	1572	1258	1049	786.24	629.15

Notes: 1. These frequencies are achieved in X1 mode or in X2 mode when WTX2 = 1: $F_{WDT} = F_{OSC} \div 2$.
2. These frequencies are achieved in X2 mode when WTX2 = 0: $F_{WDT} = F_{OSC}$.

13.3.1 WDT Behavior During Idle and Power-down Modes

Operation of the WDT during power reduction modes deserves special attention.
The WDT continues to count while the AT89C5132 are in Idle mode. This means that the user must dedicate some internal or external hardware to service the WDT during Idle mode. One approach is to use a peripheral Timer to generate an interrupt request when the Timer overflows. The interrupt service routine then clears the WDT, reloads the peripheral Timer for the next service period and puts the AT89C5132 back into Idle mode.

The Power-down mode stops all phase clocks. This causes the WDT to stop counting and to hold its count. The WDT resumes counting from where it left off if the Power-down mode is terminated by $\overline{INT0}$, $\overline{INT1}$ or keyboard interrupt. To ensure that the WDT does not overflow shortly after exiting the Power-down mode, it is recommended to clear the WDT just before entering Power-down mode.

The WDT is cleared and disabled if the Power-down mode is terminated by a reset.

13.4 Registers

Table 47. WDTRST Register

WDTRST (S:A6h Write only) – Watchdog Timer Reset Register

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

Bit Number	Bit Mnemonic	Description
7-0	-	**Watchdog Control Value**.

Reset Value = XXXX XXXXb

Table 48. WDTPRG Register

WDTPRG (S:A7h) – Watchdog Timer Program Register

7	6	5	4	3	2	1	0
-	-	-	-	-	WTO2	WTO1	WTO0

Bit Number	Bit Mnemonic	Description
7-3	-	**Reserved** The values read from these Bits are indeterminate. Do not set these Bits.
2-0	WTO2:0	**Watchdog Timer Time-Out Selection Bits** Refer to Table 46 for time-out periods.

Reset Value = XXXX X000b

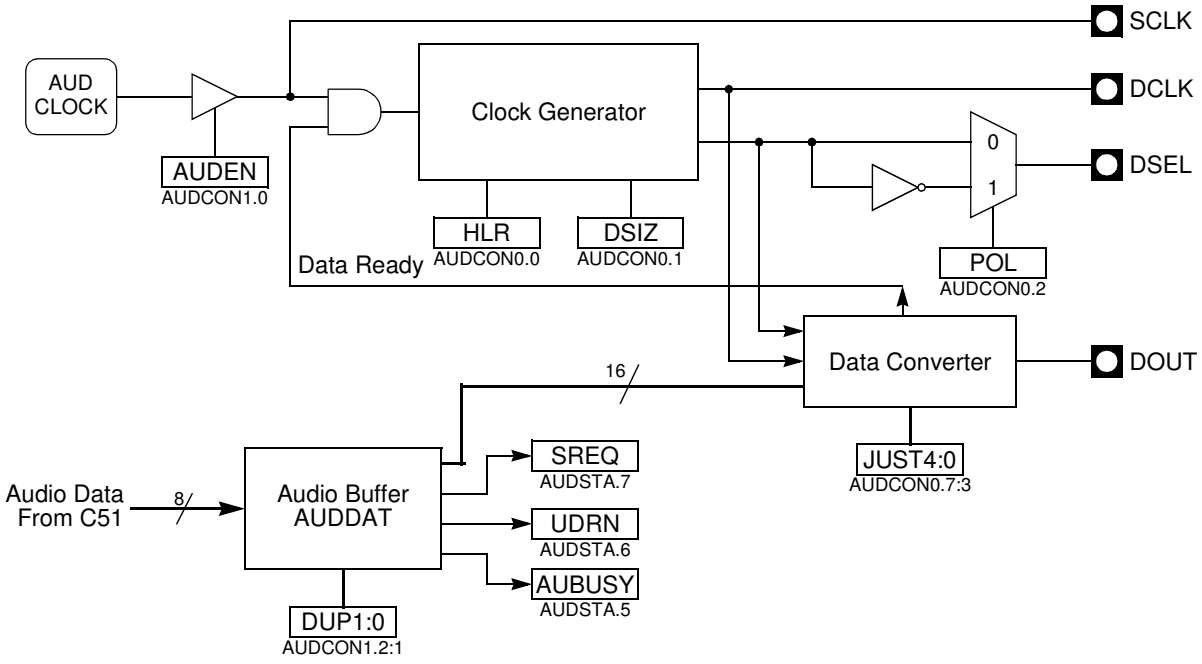
14. Audio Output Interface

The AT89C5132 implement an audio output interface allowing the audio bitstream to be output in various formats. It is compatible with right and left justification PCM and I^2S formats and thanks to the on-chip PLL (see Section “Clock Controller”, page 12) allows connection of almost all of the commercial audio DAC families available on the market.

14.1 Description

The C51 core interfaces to the audio interface through five special function registers: AUDCON0 and AUDCON1, the Audio Control registers (see Table 51 and Table 52); AUDSTA, the Audio Status register (see Table 53); AUDDAT, the Audio Data register (see Table 54); and AUDCLK, the Audio Clock Divider register (see Table 55).
Figure 14-1 shows the audio interface block diagram, blocks are detailed in the following sections.

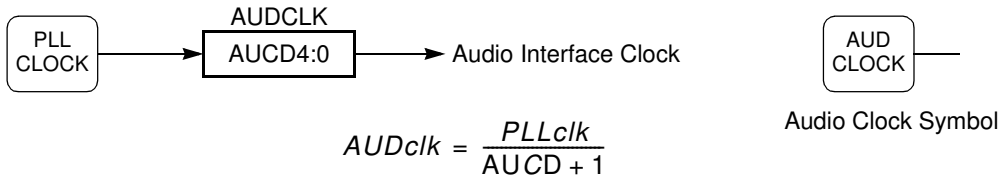
Figure 14-1. Audio Interface Block Diagram



14.2 Clock Generator

The audio interface clock is generated by division of the PLL clock. The division factor is given by AUCD4:0 bits in AUDCLK register. Figure 14-2 shows the audio interface clock generator and its calculation formula. The audio interface clock frequency depends on the audio DAC used.

Figure 14-2. Audio Clock Generator and Symbol

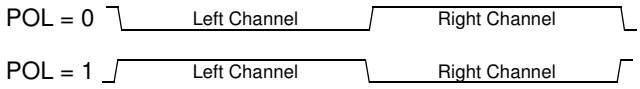


$$AUDclk = \frac{PLLclk}{AUCD + 1}$$

As soon as audio interface is enabled by setting AUDEN bit in AUDCON1 register, the master clock generated by the PLL is output on the SCLK pin which is the DAC system clock. This clock is output at 256 or 384 times the sampling frequency depending on the DAC capabilities. HLR bit in AUDCON0 register must be set according to this rate for properly generating the audio bit clock on the DCLK pin and the word selection clock on the DSEL pin. These clocks are not generated when no data is available at the data converter input.

For DAC compatibility, the bit clock frequency is programmable for outputting 16 bits or 32 bits per channel using the DSIZ bit in AUDCON0 register (see Section "Data Converter", page 61), and the word selection signal is programmable for outputting left channel on low or high level according to POL bit in AUDCON0 register as shown in Figure 14-3.

Figure 14-3. DSEL Output Polarity



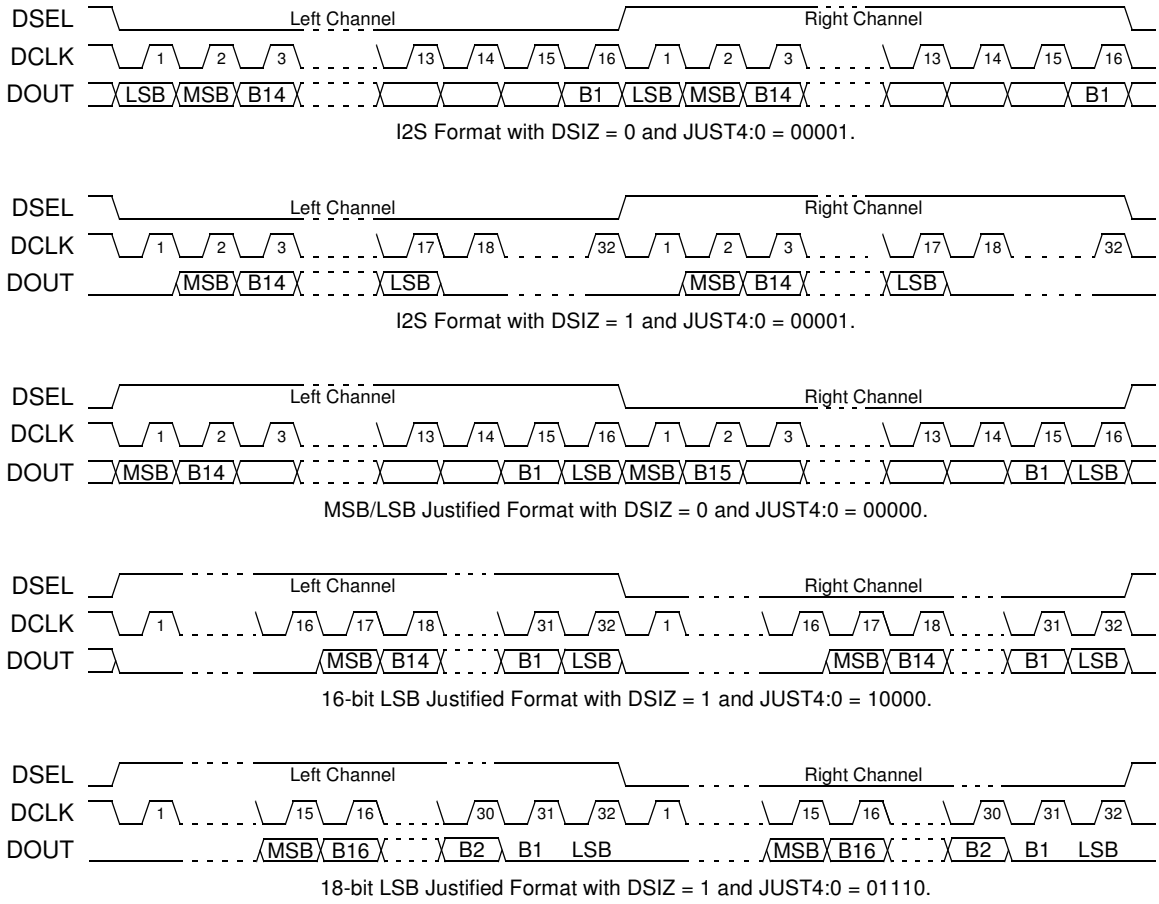
14.3 Data Converter

The data converter block converts the audio stream input from the 16-bit parallel format to a serial format. For accepting all PCM formats and I^2S format, JUST4:0 bits in AUDCON0 register are used to shift the data output point. As shown in Figure 14-4, these bits allow MSB justification by setting JUST4:0 = 00000, LSB justification by setting JUST4:0 = 10000, I^2S Justification by setting JUST4:0 = 00001, and more than 16-bit LSB justification by filling the low significant bits with logic 0.

Table 49. DAC Format Programing Examples

DAC Format	POL	DSIZ	JUST4:0
16-bit I^2S	0	0	00001
> 16-bit I^2S	0	1	00001
16-bit PCM	1	0	00000
18-bit PCM LSB justified	1	1	01110
20-bit PCM LSB justified	1	1	01100
20-bit PCM MSB justified	1	1	00000

Figure 14-4. Audio Output Format



As soon as first audio data is input to the data converter, it enables the clock generator for generating the bit and word clocks.

14.4 Audio Buffer

In voice or sound playing mode, the audio stream comes from the C51 core through an audio buffer. The data is in 8-bit format and is sampled at 8 kHz. The audio buffer adapts the sample format and rate. The sample format is extended to 16 bits by filling the LSB to 00h. Rate is adapted to the DAC rate by duplicating the data using DUP1:0 bits in AUDCON1 register according to Table 50.

The audio buffer interfaces to the C51 core through three flags: the sample request flag (SREQ in AUDSTA register), the under-run flag (UNDR in AUDSTA register) and the busy flag (AUBUSY in AUDSTA register). SREQ and UNDR can generate an interrupt request as explained in Section "Interrupt Request", page 63. The buffer size is 8 Bytes large. SREQ is set when the samples number switches from 4 to 3 and reset when the samples number switches from 4 to 5; UNDR is set when the buffer becomes empty signaling that the audio interface ran out of samples; and AUBUSY is set when the buffer is full.

Table 50. Sample Duplication Factor

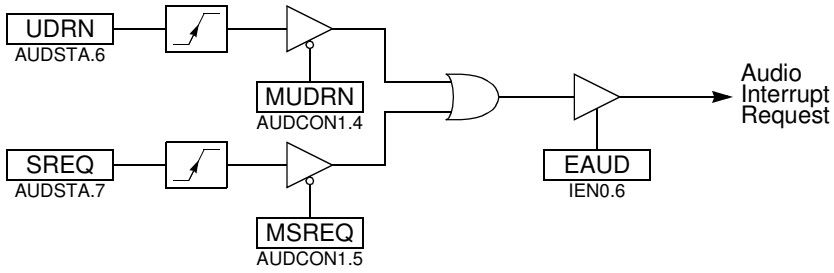
DUP1	DUP0	Factor
0	0	No sample duplication, DAC rate = 8 kHz (C51 rate).
0	1	One sample duplication, DAC rate = 16 kHz (2 x C51 rate).
1	0	Two samples duplication, DAC rate = 32 kHz (4 x C51 rate).
1	1	Three samples duplication, DAC rate = 48 kHz (6 x C51 rate).

14.5 Interrupt Request

The audio interrupt request can be generated by two sources when in C51 audio mode: a sample request when SREQ flag in AUDSTA register is set to logic 1, and an under-run condition when UDRN flag in AUDSTA register is set to logic 1. Both sources can be enabled separately by masking one of them using the MSREQ and MUDRN bits in AUDCON1 register. A global enable of the audio interface is provided by setting the EAUD bit in IEN0 register.

The interrupt is requested each time one of the two sources is set to one. The source flags are cleared by writing some data in the audio buffer through AUDDAT, but the global audio interrupt flag is cleared by hardware when the interrupt service routine is executed.

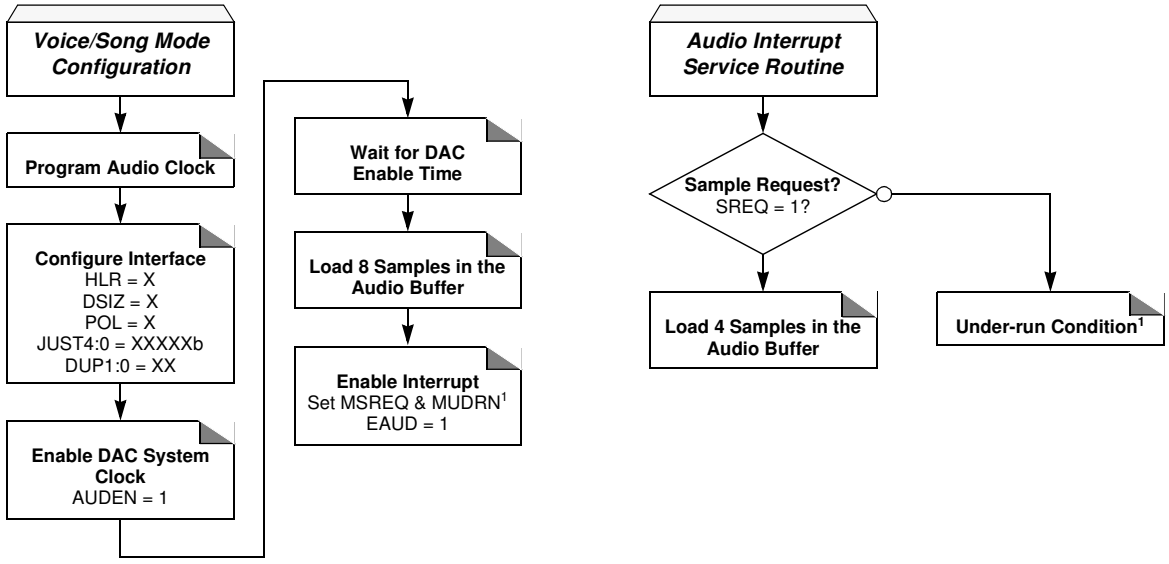
Figure 14-5. Audio Interface Interrupt System



14.6 Voice or Sound Playing

In voice or sound playing mode, the operations required are to configure the PLL and the audio interface according to the DAC selected. The audio clock is programmed to generate the 256·Fs or 384·Fs. The data flow sent by the C51 is then regulated by interrupt and data is loaded 4 Bytes by 4 Bytes. Figure 14-6 shows the configuration flow of the audio interface when in voice or sound mode.

Figure 14-6. Voice or Sound Mode Audio Flows



Note: 1. An under-run occurrence signifies that the C51 core did not respond to the previous sample request interrupt. It may never occur for a correct voice/sound generation. It is the user's responsibility to mask it or not.

14.7 Registers

Table 51. AUDCON0 Register

AUDCON0 (S:9Ah) – Audio Interface Control Register 0

7	6	5	4	3	2	1	0
JUST4	**JUST3**	**JUST2**	**JUST1**	**JUST0**	**POL**	**DSIZ**	**HLR**

Bit Number	Bit Mnemonic	Description
7-3	JUST4:0	**Audio Stream Justification Bits** Refer to Section "Data Converter", page 61 for bits description.
2	POL	**DSEL Signal Output Polarity** Set to output the left channel on high level of DSEL output (PCM mode). Clear to output the left channel on the low level of DSEL output (I^2S mode).
1	DSIZ	**Audio Data Size** Set to select 32-bit data output format. Clear to select 16-bit data output format.
0	HLR	**High/Low Rate Bit** Set by software when the PLL clock frequency is 384·Fs. Clear by software when the PLL clock frequency is 256·Fs.

Reset Value = 0000 1000b

Table 52. AUDCON1 Register

AUDCON1 (S:9Bh) – Audio Interface Control Register 1

7	6	5	4	3	2	1	0
–	**–**	**MSREQ**	**MUDRN**	**-**	**DUP1**	**DUP0**	**AUDEN**

Bit Number	Bit Mnemonic	Description
7-6	–	**Reserved** The value read from these bits is always 0. Do not set these bits.
5	MSREQ	**Audio Sample Request Flag Mask Bit** Set to prevent the SREQ flag from generating an audio interrupt. Clear to allow the SREQ flag to generate an audio interrupt.
4	MUDRN	**Audio Sample Under-run Flag Mask Bit** Set to prevent the UDRN flag from generating an audio interrupt. Clear to allow the UDRN flag to generate an audio interrupt.
3	–	**Reserved** The value read from this bit is always 0. Do not set this bit.
2-1	DUP1:0	**Audio Duplication Factor** Refer to Table 50 for bits description.
0	AUDEN	**Audio Interface Enable Bit** Set to enable the audio interface. Clear to disable the audio interface.

Reset Value = 1011 0010b

Table 53. AUDSTA Register

AUDSTA (S:9Ch Read Only) – Audio Interface Status Register

7	6	5	4	3	2	1	0
SREQ	**UDRN**	**AUBUSY**	**-**	**-**	**-**	**-**	**-**

Bit Number	Bit Mnemonic	Description
7	SREQ	**Audio Sample Request Flag** Set in C51 audio source mode when the audio interface request samples (buffer half empty). This bit generates an interrupt if not masked and if enabled in IEN0. Cleared by hardware when samples are loaded in AUDDAT.
6	UDRN	**Audio Sample Under-run Flag** Set in C51 audio source mode when the audio interface runs out of samples (buffer empty). This bit generates an interrupt if not masked and if enabled in IEN0. Cleared by hardware when samples are loaded in AUDDAT.
5	AUBUSY	**Audio Interface Busy Bit** Set in C51 audio source mode when the audio interface cannot accept more sample (buffer full). Cleared by hardware when buffer is no more full.
4-0	-	**Reserved** The value read from these bits is always 0. Do not set these bits.

Reset Value = 1100 0000b

Table 54. AUDDAT Register

AUDDAT (S:9Dh) – Audio Interface Data Register

7	6	5	4	3	2	1	0
AUD7	**AUD6**	**AUD5**	**AUD4**	**AUD3**	**AUD2**	**AUD1**	**AUD0**

Bit Number	Bit Mnemonic	Description
7-0	AUD7:0	**Audio Data** 8-bit sampling data for voice or sound playing.

Reset Value = 1111 1111b

Table 55. AUDCLK Register

AUDCLK (S:ECh) – Audio Clock Divider Register

7	6	5	4	3	2	1	0
-	-	-	AUCD4	AUCD3	AUCD2	AUCD1	AUCD0

Bit Number	Bit Mnemonic	Description
7-5	-	**Reserved** The value read from these bits is always 0. Do not set these bits.
4-0	AUCD4:0	**Audio Clock Divider** 5-bit divider for audio clock generation.

Reset Value = 0000 0000b

15. Universal Serial Bus

The AT89C5132 implement a USB device controller supporting Full-speed data transfer. In addition to the default control endpoint 0, it provides 3 other endpoints, which can be configured in Control, Bulk, Interrupt or Isochronous types.

This allows to develop firmware conforming to most USB device classes, for example the AT89C5132 support:

- 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 Device Firmware Upgrade Class, Revision 1.0 – May 13, 1999

15.0.1 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) but it is optional and may not be used (protocol 1).

The following AT89C5132 configuration adheres to that requirements:

- Endpoint 0: 32 Bytes, Control In-Out
- Endpoint 1: 64 Bytes, Bulk Out
- Endpoint 2: 64 Bytes, Bulk In
- Endpoint 3: 8 Bytes, Interrupt In

15.0.2 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 AT89C5132 configuration adheres to that requirements:

- Endpoint 0: 32 Bytes, Control In-Out
- Endpoint 1: 64 Bytes, Bulk Out
- Endpoint 2: 64 Bytes, Bulk In
- Endpoint 3: not used

15.0.3 USB Device Firmware Upgrade (DFU)

The USB Device Firmware Update (DFU) protocol can be used to upgrade the on-chip Flash memory of the AT89C5132. This allows installing product enhancements and patches to devices that are already in the field. Two different configurations and descriptor sets are used to support DFU functions. The Run-Time configuration co-exist with the usual functions of the device, which shall be USB Mass Storage for AT89C5132. It is used to initiate DFU from the normal operating mode. The DFU configuration is used to perform the firmware update after device reconfiguration 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 MaxPacketSize in the DFU configuration is 32 Bytes, which is the size of the FIFO implemented for endpoint 0.

15.1 Description

The USB device controller provides the hardware that the AT89C5132 need to interface a USB link to data flow stored in a double port memory.

It requires a 48 MHz reference clock provided by the clock controller as detailed in Section "Clock Controller", page 68. This clock is used to generate a 12 MHz Full Speed bit clock from the received USB differential data flow and to transmit data according to full speed USB device tolerance. Clock recovery is done by a Digital Phase Locked Loop (DPLL) block.

The Serial 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) controls the interface between the data flow and the Dual Port RAM, but also the interface with the C51 core itself.

Figure 15-3 shows how to connect the AT89C5132 to the USB connector. D+ and D- pins are connected through 2 termination resistors. A pull-up resistor is implemented on D+ to inform the host of a full speed device connection. Value of these resistors is detailed in the section “DC Characteristics”.

Figure 15-1. USB Device Controller Block Diagram

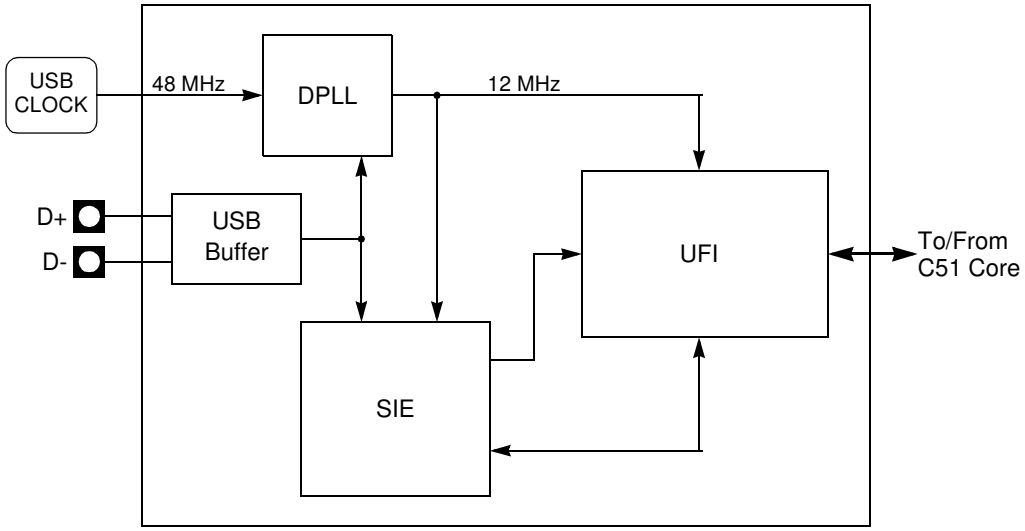
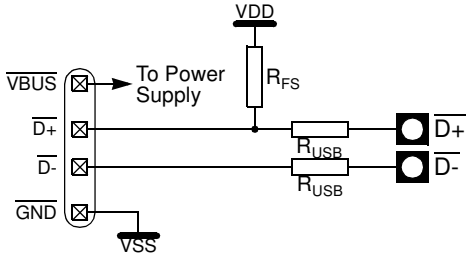


Figure 15-2. USB Connection

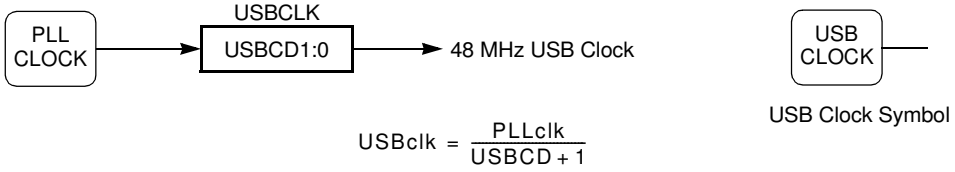


15.1.1 Clock Controller

The USB controller clock is generated by division of the PLL clock. The division factor is given by USBCD1:0 Bits in USBCLK register (see Table 70). Figure 15-3 shows the USB controller clock

generator and its calculation formula. The USB controller clock frequency must always be 48 MHz.

Figure 15-3. USB Clock Generator and Symbol

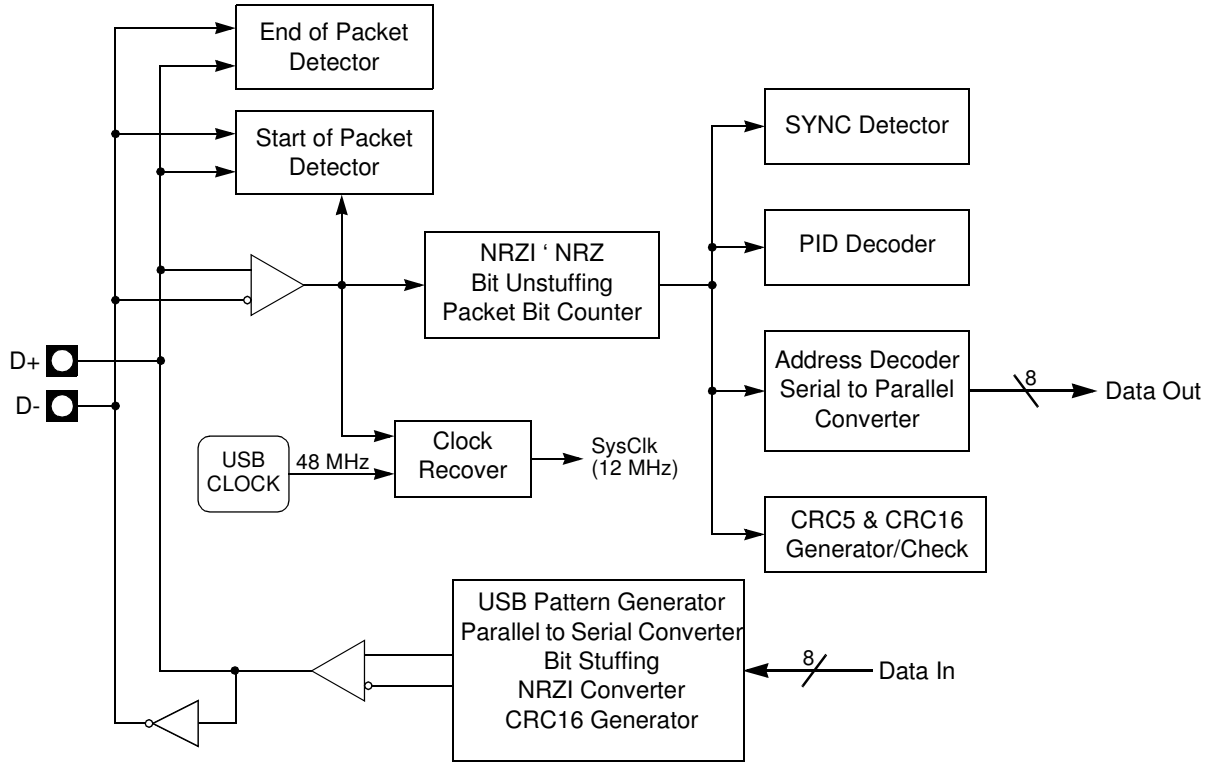


15.1.2 Serial Interface Engine (SIE)

The SIE performs the following functions:

- NRZI data encoding and decoding
- Bit stuffing and unstuffing
- CRC generation and checking
- ACKs and NACKs automatic generation
- TOKEN type identifying
- Address checking
- Clock recovery (using DPLL)

Figure 15-4. SIE Block Diagram



15.1.3 Function Interface Unit (UFI)

The Function Interface Unit provides the interface between the AT89C5132 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 15-6 shows typical USB IN and OUT transactions reporting the split in the hardware (UFI) and software (C51) load.

Figure 15-5. UFI Block Diagram

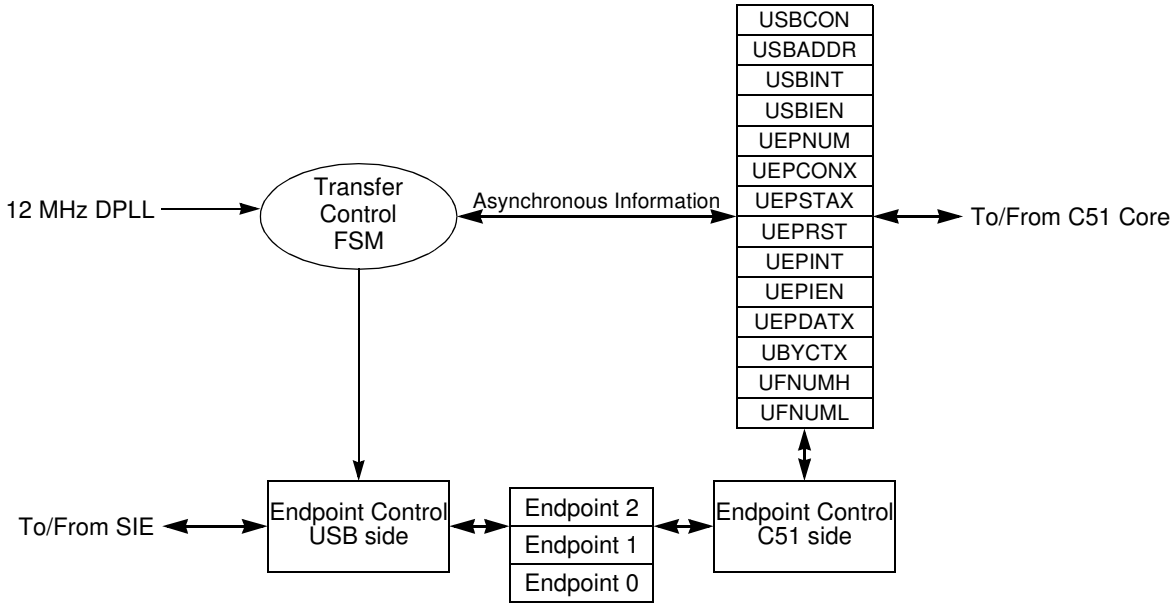
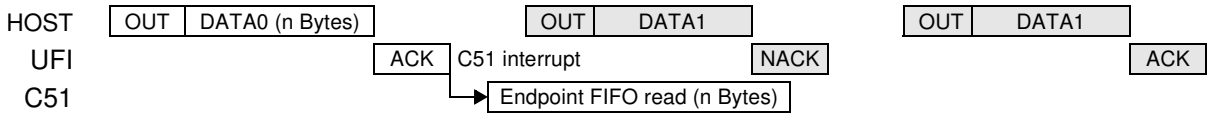


Figure 15-6. USB Typical Transaction Load

OUT Transactions:



IN Transactions:



15.2 USB Interrupt System

As shown in Figure 15-7, the USB controller of the AT89C5132 handle sixteen interrupt sources. These sources are separated in two groups: the endpoints interrupts and the controller interrupts, combined together to appear as single interrupt source for the C51 core. The USB interrupt is enabled by setting the EUSB bit in IEN1.

15.2.1 Controller Interrupt Sources

There are four controller interrupt sources which can be enabled separately in USBIEN:

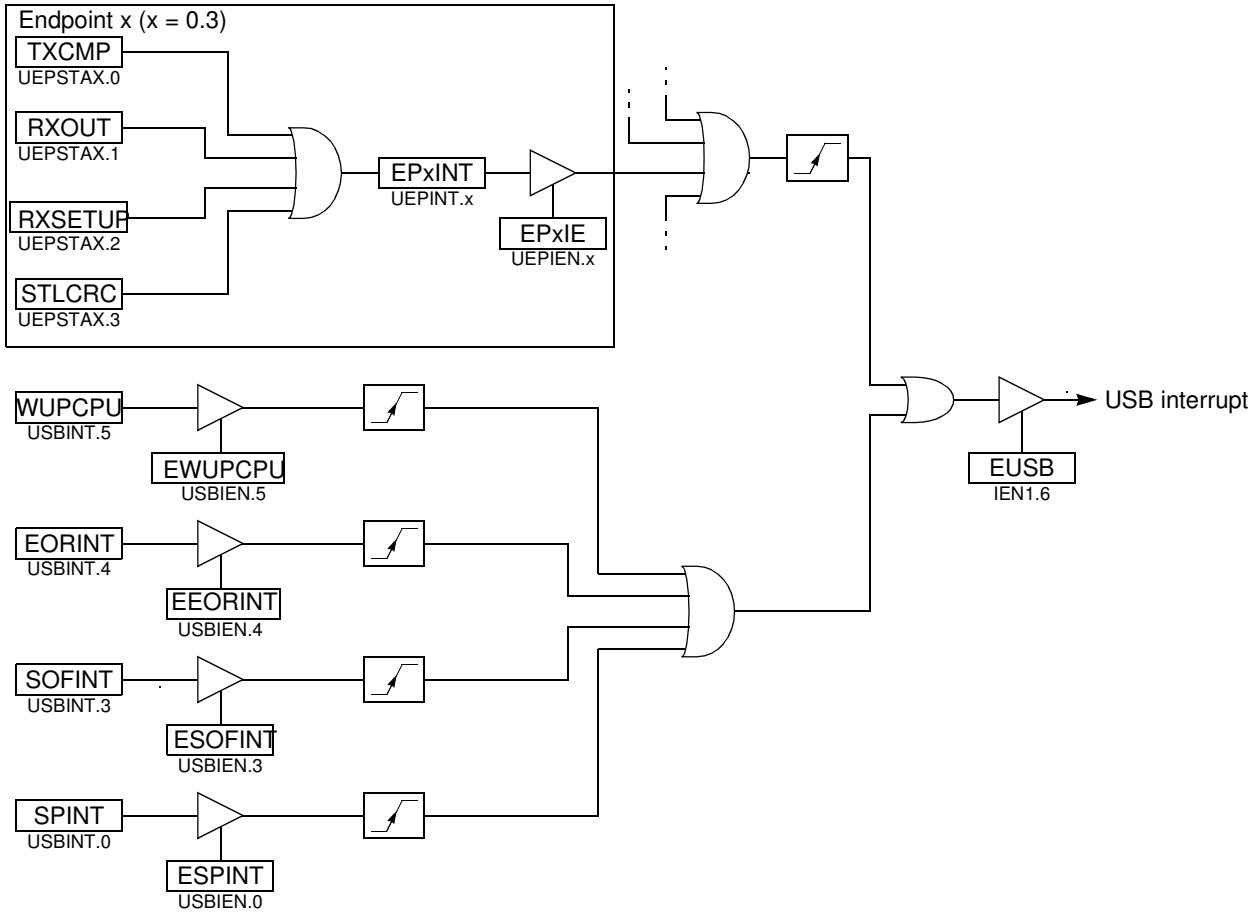
- SPINT: Suspend Interrupt Flag.
  This flag triggers an interrupt when a USB Suspend (Idle bus for three frame periods: a J state for 3 ms) is detected.
- SOFINT: Start Of Frame Interrupt Flag.
  This flag triggers an interrupt when a USB start of frame packet has been received.
- EORINT: End Of Reset Interrupt Flag.
  This flag triggers an interrupt when a End Of Reset has been detected by the USB controller.
- WUPCPU: Wake Up CPU Interrupt Flag.
  This flag triggers an interrupt when the USB controller is in SUSPEND state and is re-activated by a non-idle signal from USB line.

15.2.2 Endpoint Interrupt Sources

Each endpoint supports four interrupt sources reported in UEPSTAX and combined together to appear as a single endpoint interrupt source in UEPINT. Each endpoint interrupt can be enabled separately in UEPIEN.

- TXCMP: Transmitted In Data Interrupt Flag.
  This flag triggers an interrupt after an IN packet has been transmitted for Isochronous endpoints or after it has been accepted (ACK'ed) by the host for Control, Bulk and Interrupt endpoints.
- RXOUT: Received Out Data Interrupt Flag.
  This flag triggers an interrupt after a new packet has been received.
- RXSETUP: Receive Setup Interrupt Flag.
  This flag triggers an interrupt when a valid SETUP packet has been received from the host.
- STLCRC: Stall Sent Interrupt Flag/CRC Error Interrupt Flag.
  This flag triggers an interrupt after a STALL handshake has been sent on the bus, for Control, Bulk and Interrupt endpoints.
  This flag triggers an interrupt when the last data received is corrupted for Isochronous endpoints.

Figure 15-7. USB Interrupt Control Block Diagram



15.3 Registers

Table 56. USBCON Register

USBCON (S:BCh) – USB Global Control Register

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

Bit Number	Bit Mnemonic	Description
7	USBE	**USB Enable Bit** Set to enable the USB controller. Clear to disable and reset the USB controller.
6	SUSPCLK	**Suspend USB Clock Bit** Set to disable the 48 MHz clock input (Resume Detection is still active). Clear to enable the 48 MHz clock input.
5	SDRMWUP	**Send Remote Wake-up Bit** Set 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. Cleared by software.
4	-	**Reserved** The values read from this bit is always 0. Do not set this bit.
3	UPRSM	**Upstream Resume Bit (read only)** Set by hardware when SDRMWUP has been set and if RMWUPE is enabled. Cleared by hardware after the upstream resume has been sent.
2	RMWUPE	**Remote Wake-up Enable Bit** Set to enable request an upstream resume signalling to the host. Clear after the upstream resume has been indicated by RSMINPR. Note: Do not set this bit if the host has not set the DEVICE_REMOTE_WAKEUP feature for the device.
1	CONFG	**Configuration Bit** Set after a SET_CONFIGURATION request with a non-zero value has been correctly processed. Cleared by software when a SET_CONFIGURATION request with a zero value is received. Cleared by hardware on hardware reset or when an USB reset is detected on the bus.
0	FADDEN	**Function Address Enable Bit** Set by the device firmware after a successful status phase of a SET_ADDRESS transaction. It shall not be cleared afterwards by the device firmware. Cleared by hardware on hardware reset or when an USB reset is received. When this bit is cleared, the default function address is used (0).

Reset Value = 0000 0000b

Table 57. USBADDR Register

USBADDR (S:C6h) – USB Address Register

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 Bit** Set to enable the function. The device firmware shall set this bit after it has received a USB reset and participate in the following configuration process with the default address (FEN is reset to 0). Cleared by hardware at power-up, should not be cleared by the device firmware once set.
6-0	UADD6:0	**USB Address Bits** This field contains the default address (0) after power-up or USB bus reset. It shall be written with the value set by a SET_ADDRESS request received by the device firmware.

Reset Value = 0000 0000b

Table 58. USBINT Register

USBINT (S:BDh) – USB Global Interrupt Register

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

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
5	WUPCPU	**Wake Up CPU Interrupt Flag** 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 USBIEN. Cleared by software after re-enabling all USB clocks.
4	EORINT	**End of Reset Interrupt Flag** 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 USBIEN. Cleared by software.
3	SOFINT	**Start of Frame Interrupt Flag** Set by hardware when a USB Start of Frame packet (SOF) has been properly received. This triggers a USB interrupt when ESOFINT is set in USBIEN. Cleared by software.
2 - 1	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
0	SPINT	**Suspend Interrupt Flag** 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 USBIEN. Cleared by software.

Reset Value = 0000 0000b

Table 59. USBIEN Register

USBIEN (S:BEh) – USB Global Interrupt Enable Register

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

Bit Number	Bit Mnemonic	Description
7-6	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
5	EWUPCPU	**Wake up CPU Interrupt Enable Bit** Set to enable the Wake Up CPU interrupt. Clear to disable the Wake Up CPU interrupt.
4	EEOFINT	**End Of Reset Interrupt Enable Bit** Set to enable the End Of Reset interrupt. This bit is set after reset. Clear to disable End Of Reset interrupt.
3	ESOFINT	**Start Of Frame Interrupt Enable Bit** Set to enable the SOF interrupt. Clear to disable the SOF interrupt.
2-1	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
0	ESPINT	**Suspend Interrupt Enable Bit** Set to enable Suspend interrupt. Clear to disable Suspend interrupt.

Reset Value = 0001 0000b

Table 60. UEPNUM Register

UEPNUM (S:C7h) – USB Endpoint Number

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

Bit Number	Bit Mnemonic	Description
7 - 2	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
1 - 0	EPNUM1:0	**Endpoint Number Bits** Set this field with the number of the endpoint which shall be accessed when reading or writing to registers UEPSTAX, UEPDATX, UBYCTLX or UEPCONX.

Reset Value = 0000 0000b

Table 61. UEPCONX Register

UEPCONX (S:D4h) – USB Endpoint X Control Register (X = EPNUM set in UEPNUM)

7	6	5	4	3	2	1	0
EPEN	-	-	-	DTGL	EPDIR	EPTYPE1	EPTYPE0

Bit Number	Bit Mnemonic	Description
7	EPEN	**Endpoint Enable Bit** Set to enable the endpoint according to the device configuration. Endpoint 0 shall always be enabled after a hardware or USB bus reset and participate in the device configuration. Clear to disable the endpoint according to the device configuration.
6 - 4	-	**Reserved** The values read from this bit is always 0. Do not set this bit.
3	DTGL	**Data Toggle Status Bit (Read-only)** Set by hardware when a DATA1 packet is received. Cleared by hardware when a DATA0 packet is received. Note: When a new data packet is received without DTGL toggling from 1 to 0 or 0 to 1, a packet may have been lost. When this occurs for a Bulk endpoint, the device firmware shall consider the host has retried transmitting a properly received packet because the host has not received a valid ACK, then the firmware shall discard the new packet (N.B. The endpoint resets to DATA0 only upon configuration). For interrupt endpoints, data toggling is managed as for Bulk endpoints when used. For Control endpoints, each SETUP transaction starts with a DATA0 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 DATA1 (for a control read transfer). For Isochronous endpoints, the device firmware shall retrieve every new data packet and may ignore this bit.
2	EPDIR	**Endpoint Direction Bit** Set to configure IN direction for Bulk, Interrupt and Isochronous endpoints. Clear to configure OUT direction for Bulk, Interrupt and Isochronous endpoints. This bit has no effect for Control endpoints.
1 - 0	EPTYPE1: 0	**Endpoint Type Bits** Set this field according to the endpoint configuration (Endpoint 0 shall always be configured as Control): 0 0 Control endpoint 0 1 Isochronous endpoint 1 0 Bulk endpoint 1 1 Interrupt endpoint

Reset Value = 0000 0000b

Table 62. UEPSTAX Register

UEPSTAX (Soh) – USB Endpoint X Status and Control Register (X = EPNUM set in UEPNUM)

7	6	5	4	3	2	1	0
DIR	**-**	**STALLRQ**	**TXRDY**	**STLCRC**	**RXSETUP**	**RXOUT**	**TXCMP**

Bit Number	Bit Mnemonic	Description
7	DIR	**Control Endpoint Direction Bit** This bit is relevant only if the endpoint is configured in Control type. Set for the data stage. Clear otherwise. Note: This bit should be configured on RXSETUP interrupt before any other bit is changed. This also determines the status phase (IN for a control write and OUT for a control read). This bit should be cleared for status stage of a Control Out transaction.
6	-	**Reserved** The values read from this Bits are always 0. Do not set this bit.
5	STALLRQ	**Stall Handshake Request Bit** Set to send a STALL answer to the host for the next handshake.Clear otherwise.
4	TXRDY	**TX Packet Ready Control Bit** Set after a packet has been written into the endpoint FIFO for IN data transfers. Data shall 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, which is generally recommended and may be required to terminate a transfer when the length of the last data packet is equal to MaxPacketSize (e.g., for control read transfers). 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.
3	STLCRC	**Stall Sent Interrupt Flag/CRC Error Interrupt Flag** **For Control, Bulk and Interrupt Endpoints:** Set by hardware after a STALL handshake has been sent as requested by STALLRQ. Then, the endpoint interrupt is triggered if enabled in UEPIEN. Cleared by hardware when a SETUP packet is received (see RXSETUP). **For Isochronous Endpoints:** Set by hardware if the last data received is corrupted (CRC error on data). Then, the endpoint interrupt is triggered if enabled in UEPIEN. Cleared by hardware when a non corrupted data is received.
2	RXSETUP	**Received SETUP Interrupt Flag** 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 in UEPIEN. Clear by software after reading the SETUP data from the endpoint FIFO.
1	RXOUT	**Received OUT Data Interrupt Flag** Set by hardware after an OUT packet has been received. Then, the endpoint interrupt is triggered if enabled in UEPIEN and all the following OUT packets to the endpoint are rejected (NACK'ed) until this bit is cleared. 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. Clear by software after reading the OUT data from the endpoint FIFO.
0	TXCMP	**Transmitted IN Data Complete Interrupt Flag** 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 in UEPIEN. Clear by software before setting again TXRDY.

Reset Value = 0000 0000b

Table 63. UEPRST Register

UEPRST (S:D5h) – USB Endpoint FIFO Reset Register

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

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
3	EP3RST	**Endpoint 3 FIFO Reset** Set and clear to reset the endpoint 3 FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received.
2	EP2RST	**Endpoint 2 FIFO Reset** Set and clear to reset the endpoint 2 FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received.
1	EP1RST	**Endpoint 1 FIFO Reset** Set and clear to reset the endpoint 1 FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received.
0	EP0RST	**Endpoint 0 FIFO Reset** Set and clear to reset the endpoint 0 FIFO prior to any other operation, upon hardware reset or when an USB bus reset has been received.

Reset Value = 0000 0000b

Table 64. UEPINT Register

UEPINT (S:F8h Read-only) – USB Endpoint Interrupt Register

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

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
3	EP3INT	**Endpoint 3 Interrupt Flag** Set by hardware when an interrupt is triggered in UEPSTAX and the endpoint 3 interrupt is enabled in UEPIEN. Must be cleared by software.
2	EP2INT	**Endpoint 2 Interrupt Flag** Set by hardware when an interrupt is triggered in UEPSTAX and the endpoint 2 interrupt is enabled in UEPIEN. Must be cleared by software.
1	EP1INT	**Endpoint 1 Interrupt Flag** Set by hardware when an interrupt is triggered in UEPSTAX and the endpoint 1 interrupt is enabled in UEPIEN. Must be cleared by software.
0	EP0INT	**Endpoint 0 Interrupt Flag** Set by hardware when an interrupt is triggered in UEPSTAX and the endpoint 0 interrupt is enabled in UEPIEN. Must be cleared by software.

Reset Value = 0000 0000b

Table 65. UEPIEN Register

UEPIEN (S:C2h) – USB Endpoint Interrupt Enable Register

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

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
3	EP3INTE	**Endpoint 3 Interrupt Enable Bit** Set to enable the interrupts for endpoint 3. Clear to disable the interrupts for endpoint 3.
2	EP2INTE	**Endpoint 2 Interrupt Enable Bit** Set to enable the interrupts for endpoint 2. Clear this bit to disable the interrupts for endpoint 2.
1	EP1INTE	**Endpoint 1 Interrupt Enable Bit** Set to enable the interrupts for the endpoint 1. Clear to disable the interrupts for the endpoint 1.
0	EP0INTE	**Endpoint 0 Interrupt Enable Bit** Set to enable the interrupts for the endpoint 0. Clear to disable the interrupts for the endpoint 0.

Reset Value = 0000 0000b

Table 66. UEPDATX Register

UEPDATX (S:CFh) – USB Endpoint X FIFO Data Register (X = EPNUM set in UEPNUM)

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

Bit Number	Bit Mnemonic	Description
7 - 0	FDAT7:0	**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 = XXh

Table 67. UBYCTLX Register

UBYCTX (S:E2h) – USB Endpoint X Byte Count Register (X = EPNUM set in UEPNUM)

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The values read from this Bits are always 0. Do not set this bit.
6-0	BYCT7:0	**Byte Count** 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. UFNUML Register

UFNUML (S:BAh, Read-only) – USB Frame Number Low Register

7	6	5	4	3	2	1	0
FNUM7	**FNUM6**	**FNUM5**	**FNUM4**	**FNUM3**	**FNUM2**	**FNUM1**	**FNUM0**

Bit Number	Bit Mnemonic	Description
7 - 0	FNUM7:0	**Frame Number** Lower 8 Bits of the 11-bit Frame Number.

Reset Value = 00h

Table 69. UFNUMH Register

UFNUMH (S:BBh, Read-only) – USB Frame Number High Register

7	6	5	4	3	2	1	0
-	**-**	**CRCOK**	**CRCERR**	**-**	**FNUM10**	**FNUM9**	**FNUM8**

Bit Number	Bit Mnemonic	Description
7 - 3	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
5	CRCOK	**Frame Number CRC OK Bit** Set by hardware after a non corrupted Frame Number in Start of Frame Packet is received. Updated after every Start Of Frame packet reception. Note: The Start Of Frame interrupt is generated just after the PID receipt.
4	CRCERR	**Frame Number CRC Error Bit** Set by hardware after a corrupted Frame Number in Start of Frame Packet is received. Updated after every Start Of Frame packet reception. Note: The Start Of Frame interrupt is generated just after the PID receipt.
3	-	**Reserved** The values read from this Bits are always 0. Do not set this bit.
2 - 0	FNUM10:8	**Frame Number** Upper 3 Bits of the 11-bit Frame Number. It is provided in the last received SOF packet. FNUM does not change if a corrupted SOF is received.

Reset Value = 00h

Table 70. USBCLK Register

USBCLK (S:EAh) – USB Clock Divider Register

7	6	5	4	3	2	1	0
-	**-**	**-**	**-**	**-**	**-**	**USBCD1**	**USBCD0**

Bit Number	Bit Mnemonic	Description
7 - 2	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.

Bit Number	Bit Mnemonic	Description
1 - 0	USBCD1:0	**USB Controller Clock Divider** 2-bit divider for USB controller clock generation.

Reset Value = 0000 0000b

16. MultiMedia Card Controller

The AT89C5132 implements a MultiMedia Card (MMC) controller. The MMC is used to store files in removable Flash memory cards that can be easily plugged or removed from the application.

16.1 Card Concept

The basic MultiMedia Card concept is based on transferring data via a minimal number of signals.

16.1.1 Card Signals

The communication signals are:

- CLK: with each cycle of this signal an one bit transfer on the command and data lines is done. The frequency may vary from zero to the maximum clock frequency.
- CMD: is a bidirectional command channel used for card initialization and data transfer commands. The CMD signal has two operation modes: open-drain for initialization mode and push-pull for fast command transfer. Commands are sent from the MultiMedia Card bus master to the card and responses from the cards to the host.
- DAT: is a bidirectional data channel. The DAT signal operates in push-pull mode. Only one card or the host is driving this signal at a time.

16.1.2 Card Registers

Within the card interface five registers are defined: OCR, CID, CSD, RCA and DSR. These can be accessed only by corresponding commands.

The 32-bit Operation Conditions Register (OCR) stores the V_{DD} voltage profile of the card. The register is optional and can be read only.

The 128-bit wide CID register carries the card identification information (Card ID) used during the card identification procedure.

The 128-bit wide Card-Specific Data register (CSD) provides information on how to access the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, and whether the DSR register can be used.
The 16-bit Relative Card Address register (RCA) carries the card address assigned by the host during the card identification. This address is used for the addressed host-card communication after the card identification procedure

The 16-bit Driver Stage Register (DSR) can be optionally used to improve the bus performance for extended operating conditions (depending on parameters like bus length, transfer rate or number of cards).

16.2 Bus Concept

The MultiMedia Card bus is designed to connect either solid-state mass-storage memory or I/O-devices in a card format to multimedia applications. The bus implementation allows the coverage of application fields from low-cost systems to systems with a fast data transfer rate. It is a single master bus with a variable number of slaves. The MultiMedia Card bus master is the bus controller and each slave is either a single mass storage card (with possibly different technologies such as ROM, OTP, Flash etc.) or an I/O-card with its own controlling unit (on card) to perform the data transfer.

The MultiMedia Card bus also includes power connections to supply the cards.

The bus communication uses a special protocol (MultiMedia Card bus protocol) which is applicable for all devices. Therefore, the payload data transfer between the host and the cards can be bidirectional.

16.2.1 Bus Lines

The MultiMedia Card bus architecture requires all cards to be connected to the same set of lines. No card has an individual connection to the host or other devices, which reduces the connection costs of the MultiMedia Card system.

The bus lines can be divided into three groups:

- Power supply: V_{SS1} and V_{SS2}, V_{DD} – used to supply the cards.
- Data transfer: MCMD, MDAT – used for bidirectional communication.
- Clock: MCLK – used to synchronize data transfer across the bus.

16.2.2 Bus Protocol

After a Power-on reset, the host must initialize the cards by a special message-based MultiMedia Card bus protocol. Each message is represented by one of the following tokens:

- Command: a command is a token which starts an operation. A command is transferred serially from the host to the card on the MCMD line.
- Response: a response is a token which is sent from an addressed card (or all connected cards) to the host as an answer to a previously received command. It is transferred serially on the MCMD line.
- Data: data can be transferred from the card to the host or vice-versa. Data is transferred serially on the MDAT line.

Card addressing is implemented using a session address assigned during the initialization phase, by the bus controller to all currently connected cards. Individual cards are identified by their CID number. This method requires that every card will have an unique CID number. To ensure uniqueness of CIDs the CID register contains 24 Bits (MID and OID fields) which are defined by the MMCA. Every card manufacturers is required to apply for an unique MID (and optionally OID) number.

MultiMedia Card bus data transfers are composed of these tokens. One data transfer is a bus operation. There are different types of operations. Addressed operations always contain a command and a response token. In addition, some operations have data token, the others transfer their information directly within the command or response structure. In this case no data token is present in an operation. The Bits on the MDAT and the MCMD lines are transferred synchronous to the host clock.

Two types of data transfer commands are defined:

- Sequential commands: These commands initiate a continuous data stream, they are terminated only when a stop command follows on the MCMD line. This mode reduces the command overhead to an absolute minimum.
- Block-oriented commands: These commands send data block succeeded by CRC Bits. Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the MCMD line similarly to the stream read.

Figure 16-1 to Figure 16-5 show the different types of operations, on these figures, grayed tokens are from host to card(s) while white tokens are from card(s) to host.

Figure 16-1. Sequential Read Operation

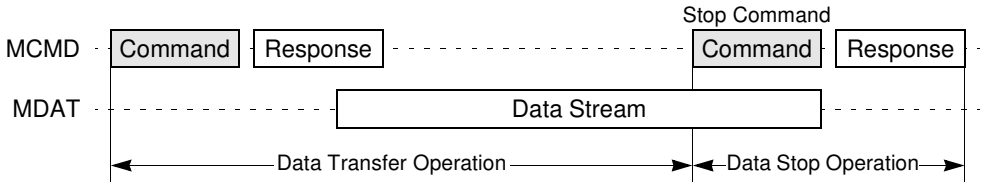
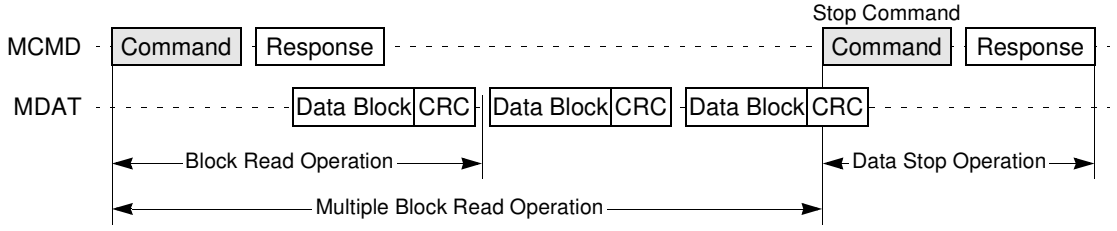


Figure 16-2. (Multiple) Block Read Operation



As shown in Figure 16-3 and Figure 16-4 the data write operation uses a simple busy signalling of the write operation duration on the data line (MDAT).

Figure 16-3. Sequential Write Operation

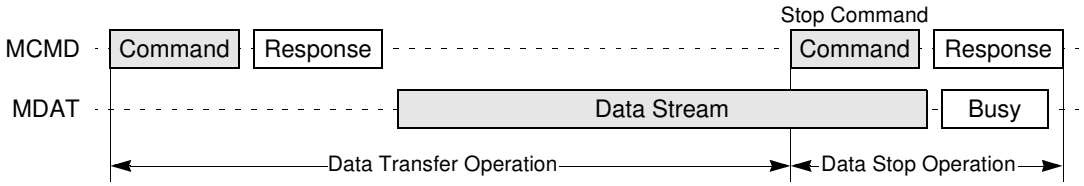


Figure 16-4. (Multiple) Block Write Operation

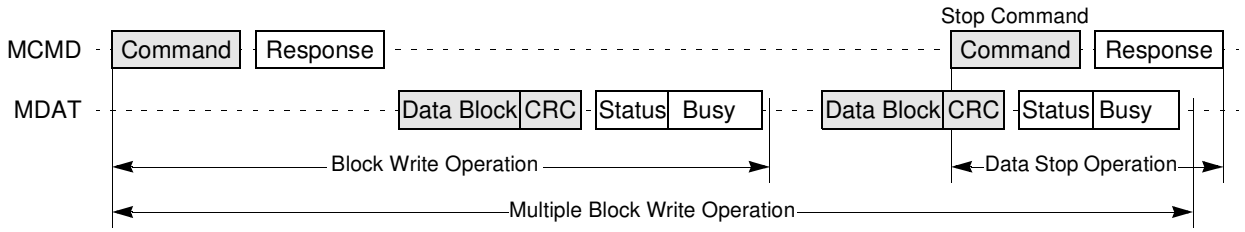
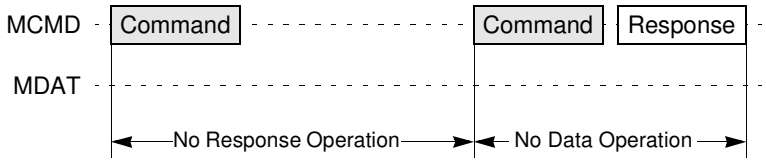


Figure 16-5. No Response and No Data Operation



16.2.3 Command Token Format

As shown in Figure 16-6, commands have a fixed code length of 48 Bits. Each command token is preceded by a Start bit: a low level on MCMD line and succeeded by an End bit: a high level on MCMD line. The command content is preceded by a Transmission bit: a high level on MCMD line for a command token (host to card) and succeeded by a 7-bit CRC so that transmission errors can be detected and the operation may be repeated.
Command content contains the command index and address information or parameters.

Figure 16-6. Command Token Format

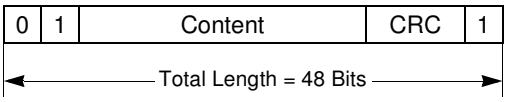


Table 71. Command Token Format

Bit Position	47	46	45:40	39:8	7:1	0
Width (Bits)	1	1	6	32	7	1
Value	'0'	'1'	-	-	-	'1'
Description	Start bit	Transmission bit	Command Index	Argument	CRC7	End bit

16.2.4 Response Token Format

There are five types of response tokens (R1 to R5). As shown in Figure 16-7, responses have a code length of 48 Bits or 136 Bits. A response token is preceded by a Start bit: a low level on MCMD line and succeeded by an End bit: a high level on MCMD line. The command content is preceded by a Transmission bit: a low level on MCMD line for a response token (card to host) and succeeded (R1,R2,R4,R5) or not (R3) by a 7-bit CRC.

Response content contains mirrored command and status information (R1 response), CID register or CSD register (R2 response), OCR register (R3 response), or RCA register (R4 and R5 response).

Figure 16-7. Response Token Format

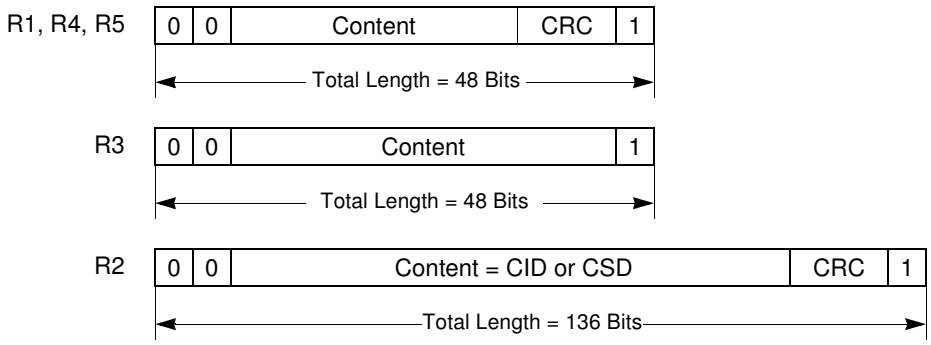


Table 72. R1 Response Format (Normal Response)

Bit Position	47	46	45:40	39:8	7:1	0
Width (Bits)	1	1	6	32	7	1
Value	'0'	'0'	-	-	-	'1'
Description	Start bit	Transmission bit	Command Index	Card Status	CRC7	End bit

Table 73. R2 Response Format (CID and CSD registers)

Bit Position	135	134	[133:128]	[127:1]	0
Width (Bits)	1	1	6	32	1
Value	'0'	'0'	'111111'	-	'1'
Description	Start bit	Transmission bit	Reserved	Argument	End bit

Table 74. R3 Response Format (OCR Register)

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (Bits)	1	1	6	32	7	1
Value	'0'	'0'	'111111'	-	'1111111'	'1'
Description	Start bit	Transmission bit	Reserved	OCR register	Reserved	End bit

Table 75. R4 Response Format (Fast I/O)

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (Bits)	1	1	6	32	7	1
Value	'0'	'0'	'100111'	-	-	'1'
Description	Start bit	Transmission bit	Command Index	Argument	CRC7	End bit

Table 76. R5 Response Format

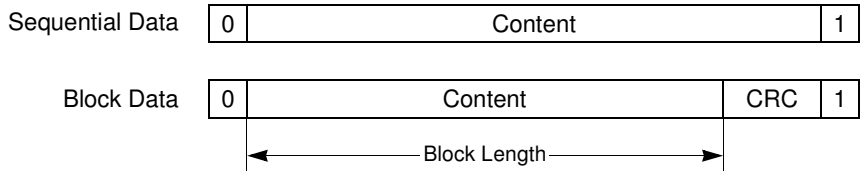
Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (Bits)	1	1	6	32	7	1
Value	'0'	'0'	'101000'	-	-	'1'
Description	Start bit	Transmission bit	Command Index	Argument	CRC7	End bit

16.2.5 Data Packet Format

There are two types of data packets: stream and block. As shown in Figure 16-8, stream data packets have an indeterminate length while block packets have a fixed length depending on the block length. Each data packet is preceded by a Start bit: a low level on MCMD line and succeeded by an End bit: a high level on MCMD line. Due to the fact that there is no predefined end

in stream packets, CRC protection is not included in this case. The CRC protection algorithm for block data is a 16-bit CCITT polynomial.

Figure 16-8. Data Token Format



16.2.6 Clock Control

The MMC bus clock signal can be used by the host to turn the cards into energy saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to lower the clock frequency or shut it down.

There are a few restrictions the host must follow:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the cards, and the identification frequency defined by the specification document).
- It is an obvious requirement that the clock must be running for the card to output data or response tokens. After the last MultiMedia Card bus transaction, the host is required, to provide 8 (eight) clock cycles for the card to complete the operation before shutting down the clock. Following is a list of the various bus transactions:
- A command with no response. 8 clocks after the host command End bit.
- A command with response. 8 clocks after the card command End bit.
- A read data transaction. 8 clocks after the End bit of the last data block.
- A write data transaction. 8 clocks after the CRC status token.
- The host is allowed to shut down the clock of a “busy” card. The card will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge the card (unless previously disconnected by a deselect command-CMD7) will force the MDAT line down, forever.

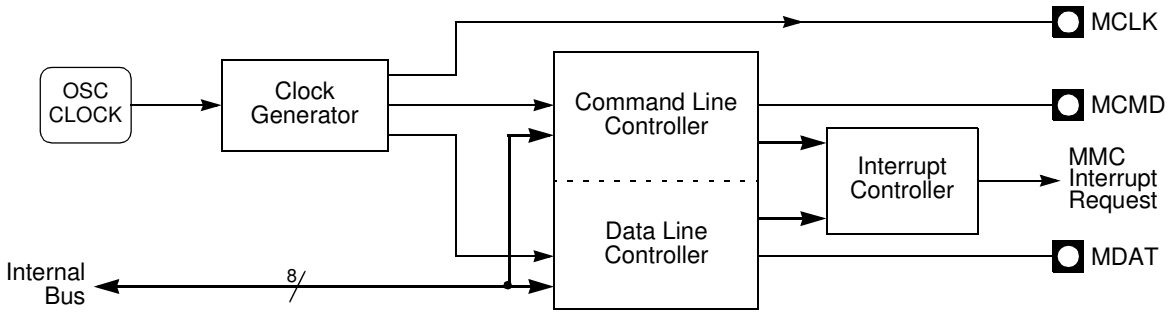
16.3 Description

The MMC controller interfaces to the C51 core through the following eight special function registers:

MMCON0, MMCON1, MMCON2, the three MMC control registers (see Figure 78 to Figure); MMSTA, the MMC status register (see Figure 81); MMINT, the MMC interrupt register (see Figure); MMMSK, the MMC interrupt mask register (see Figure 83); MMCMD, the MMC command register (see Figure 84); MMDAT, the MMC data register (see Figure); and MMCLK, the MMC clock register (see Figure 86).

As shown in Figure 16-9, the MMC controller is divided in four blocks: the clock generator that handles the MCLK (formally the MMC CLK) output to the card, the command line controller that handles the MCMD (formally the MMC CMD) line traffic to or from the card, the data line controller that handles the MDAT (formally the MMC DAT) line traffic to or from the card, and the interrupt controller that handles the MMC controller interrupt sources. These blocks are detailed in the following sections.

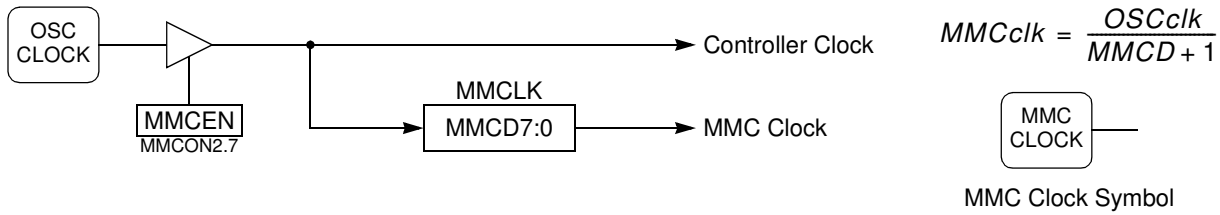
Figure 16-9. MMC Controller Block Diagram



16.4 Clock Generator

The MMC clock is generated by division of the oscillator clock (F_{OSC}) issued from the Clock Controller block as detailed in Section "Oscillator", page 12. The division factor is given by MMCD7:0 Bits in MMCLK register. Figure 16-10 shows the MMC clock generator and its output clock calculation formula.

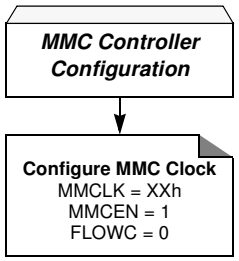
Figure 16-10. MMC Clock Generator and Symbol



As soon as MMCEN bit in MMCON2 is set, the MMC controller receives its system clock. The MMC command and data clock is generated on MCLK output and sent to the command line and data line controllers. Figure 16-11 shows the MMC controller configuration flow.

As exposed in Section “Clock Control”, MMCD7:0 Bits can be used to dynamically increase or reduce the MMC clock.

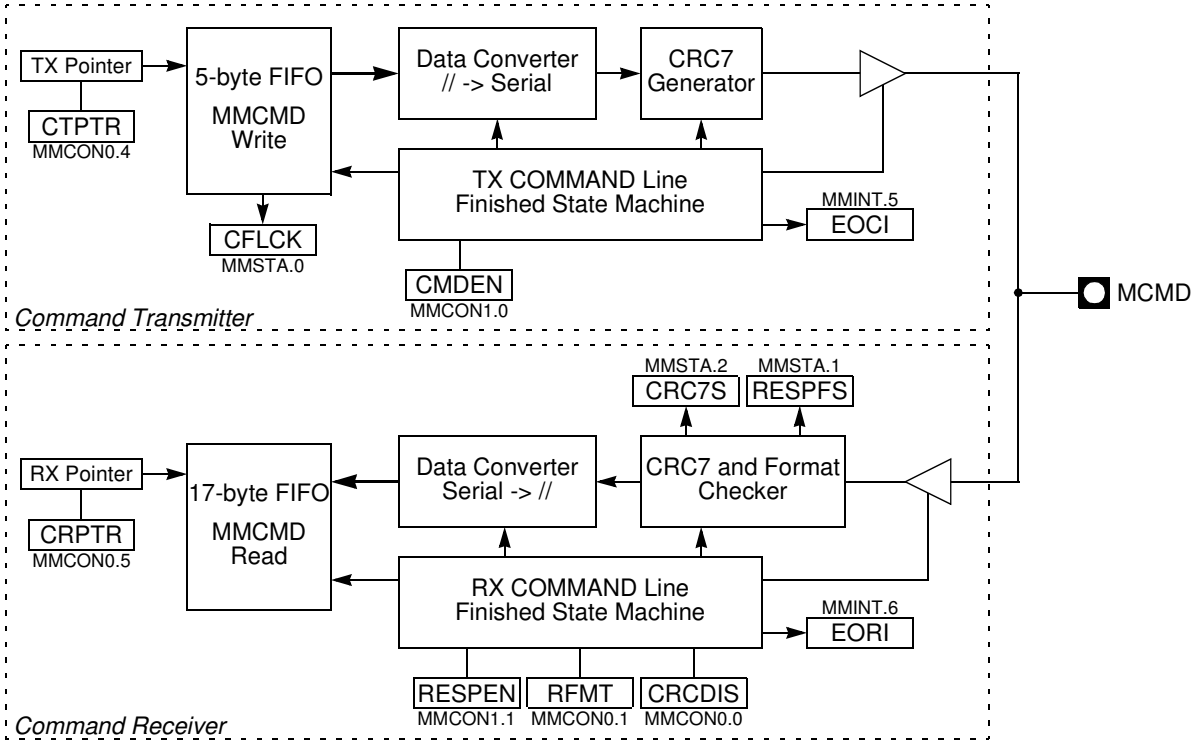
Figure 16-11. Configuration Flow



16.5 Command Line Controller

As shown in Figure 16-12, the command line controller is divided in two channels: the command transmitter channel that handles the command transmission to the card through the MCMD line and the command receiver channel that handles the response reception from the card through the MCMD line. These channels are detailed in the following sections.

Figure 16-12. Command Line Controller Block Diagram



16.5.1 Command Transmitter

To send a command to the card, the user must load the command index (1 byte) and argument (4 Bytes) in the command transmit FIFO using the MMCMD register. Before starting transmission by setting and clearing the CMDEN bit in MMCON1 register, the user must first configure:

- RESPEN bit in MMCON1 register to indicate whether a response is expected or not.
- RFMT bit in MMCON0 register to indicate the response size expected.
- CRCDIS bit in MMCON0 register to indicate whether the CRC7 included in the response will be computed or not. In order to avoid CRC error, CRCDIS may be set for responses that do not include CRC7.

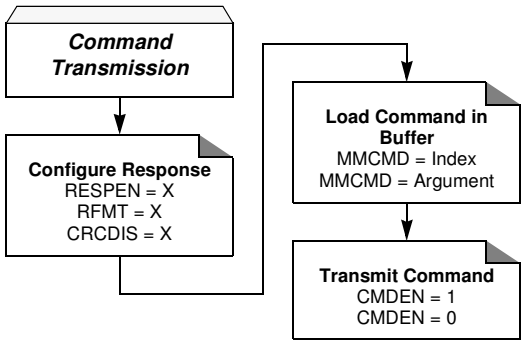
Figure 16-13 summarizes the command transmission flow.

As soon as command transmission is enabled, the CFLCK flag in MMSTA is set indicating that write to the FIFO is locked. This mechanism is implemented to avoid command over-run.

The end of the command transmission is signalled by the EOCI flag in MMINT register becoming set. This flag may generate an MMC interrupt request as detailed in Section "Interrupt", page 96. The end of the command transmission also resets the CFLCK flag.

The user may abort command loading by setting and clearing the CTPTR bit in MMCON0 register which resets the write pointer to the transmit FIFO.

Figure 16-13. Command Transmission Flow



16.5.2 Command Receiver

The end of the response reception is signalled by the EORI flag in MMINT register. This flag may generate an MMC interrupt request as detailed in Section "Interrupt", page 96. When this flag is set, two other flags in MMSTA register: RESPFS and CRC7S give a status on the response received. RESPFS indicates if the response format is correct or not: the size is the one expected (48 Bits or 136 Bits) and a valid End bit has been received, and CRC7S indicates if the CRC7 computation is correct or not. These Flags are cleared when a command is sent to the card and updated when the response has been received.

The user may abort response reading by setting and clearing the CRPTR bit in MMCON0 register which resets the read pointer to the receive FIFO.

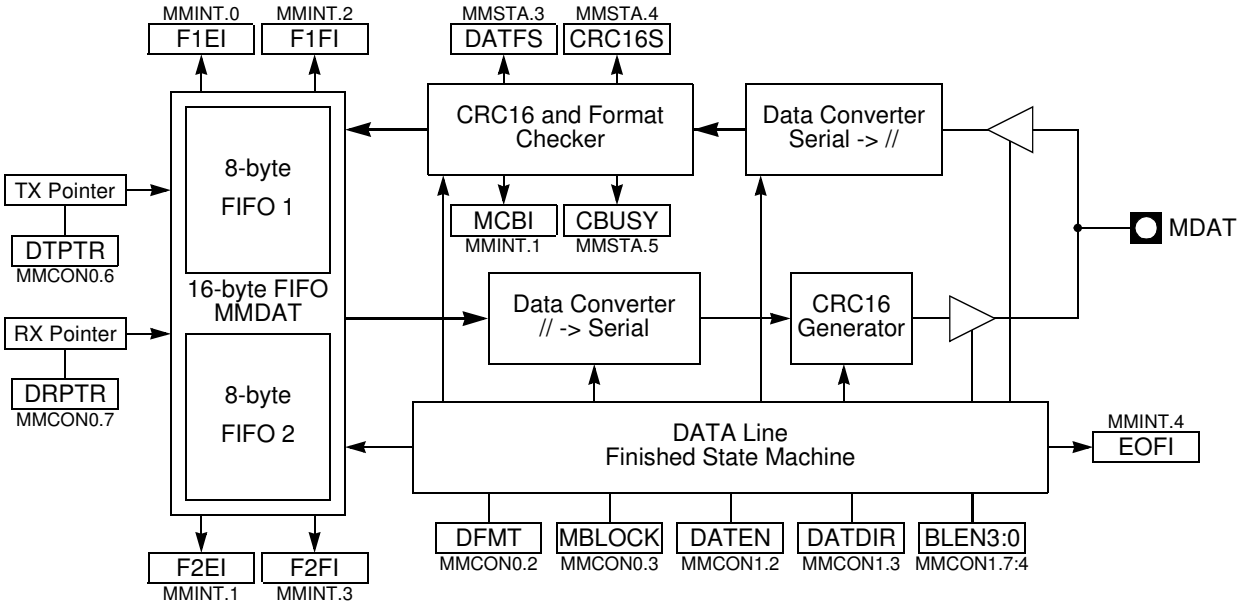
According to the MMC specification delay between a command and a response (formally N_{CR} parameter) cannot exceed 64 MMC clock periods. To avoid any locking of the MMC controller when card does not send its response (e.g. physically removed from the bus), user must launch a timeout period to exit from such situation. In case of timeout user may reset the command controller and its internal state machine by setting and clearing the CCR bit in MMCON2 register.

This timeout may be disarmed when receiving the response.

16.6 Data Line Controller

The data line controller is based on a 16-byte FIFO used both by the data transmitter channel and by the data receiver channel.

Figure 16-14. Data Line Controller Block Diagram



16.6.1 FIFO Implementation

The 16-byte FIFO is based on a dual 8-byte FIFO managed using two pointers and four flags indicating the status full and empty of each FIFO.

Pointers are not accessible to user but can be reset at any time by setting and clearing DRPTR and DTPTR Bits in MMCON0 register. Resetting the pointers is equivalent to abort the writing or reading of data.

F1EI and F2EI flags in MMINT register signal when set that respectively FIFO1 and FIFO2 are empty. F1FI and F2FI flags in MMINT register signal when set that respectively FIFO1 and FIFO2 are full. These flags may generate an MMC interrupt request as detailed in Section “Interrupt”.

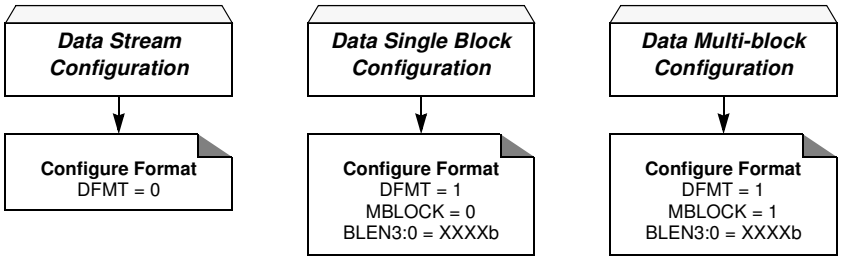
16.6.2 Data Configuration

Before sending or receiving any data, the data line controller must be configured according to the type of the data transfer considered. This is achieved using the Data Format bit: DFMT in MMCON0 register. Clearing DFMT bit enables the data stream format while setting DFMT bit enables the data block format. In data block format, user must also configure the single or multi-block mode by clearing or setting the MBLOCK bit in MMCON0 register and the block length using BLEN3:0 Bits in MMCON1 according to Table 77. Figure 16-15 summarizes the data modes configuration flows.

Table 77. Block Length Programming

BLEN3:0	Block Length (Byte)
BLEN = 0000 to 1011	Length = 2^{BLEN}: 1 to 2048
> 1011	Reserved: do not program BLEN3:0 > 1011

Figure 16-15. Data Controller Configuration Flows



16.6.3 Data Transmitter

16.6.3.1 Configuration

For transmitting data to the card, user must first configure the data controller in transmission mode by setting the DATDIR bit in MMCON1 register.

Figure 16-16 summarizes the data stream transmission flows in both polling and interrupt modes while Figure 16-17 summarizes the data block transmission flows in both polling and interrupt modes, these flows assume that block length is greater than 16 data.

16.6.3.2 Data Loading

Data is loaded in the FIFO by writing to MMDAT register. Number of data loaded may vary from 1 to 16 Bytes. Then if necessary (more than 16 Bytes to send) user must wait that one FIFO becomes empty (F1EI or F2EI set) before loading 8 new data.

16.6.3.3 Data Transmission

Transmission is enabled by setting and clearing DATEN bit in MMCON1 register.

Data is transmitted immediately if the response has already been received, or is delayed after the response reception if its status is correct. In both cases transmission is delayed if a card sends a busy state on the data line until the end of this busy condition.

According to the MMC specification, the data transfer from the host to the card may not start sooner than 2 MMC clock periods after the card response was received (formally N_{WR} parameter). To address all card types, this delay can be programmed using DATD1:0 Bits in MMCON2 register from 3 MMC clock periods when DATD1:0 Bits are cleared to 9 MMC clock periods when DATD2:0 Bits are set, by step of 2 MMC clock periods.

16.6.3.4 End of Transmission

The end of data frame (block or stream) transmission is signalled by the EOFI flag in MMINT register. This flag may generate an MMC interrupt request as detailed in Section "Interrupt", page 96.

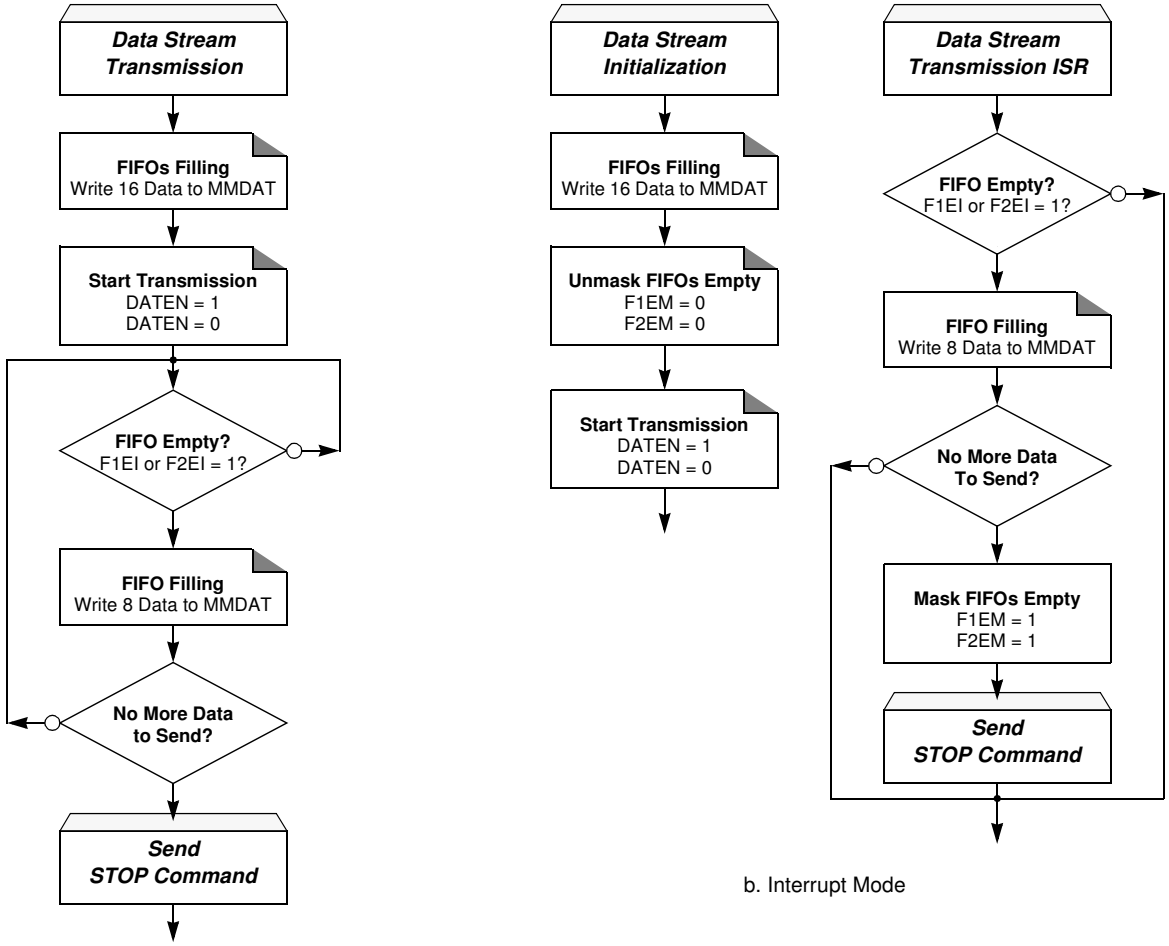
In data stream mode, EOFI flag is set, after reception of the End bit. This assumes user has previously sent the STOP command to the card, which is the only way to stop stream transfer.

In data block mode, EOFI flag is set, after reception of the CRC status token (see Figure 16-4). Two other flags in MMSTA register: DATFS and CRC16S report a status on the frame sent. DATFS indicates if the CRC status token format is correct or not, and CRC16S indicates if the card has found the CRC16 of the block correct or not.

16.6.3.5 Busy Status

As shown in Figure 16-4 the card uses a busy token during a block write operation. This busy status is reported by the CBUSY flag in MMSTA register and by the MCBI flag in MMINT which is set every time CBUSY toggles, i.e. when the card enters and exits its busy state. This flag may generate an MMC interrupt request as detailed in Section "Interrupt", page 96.

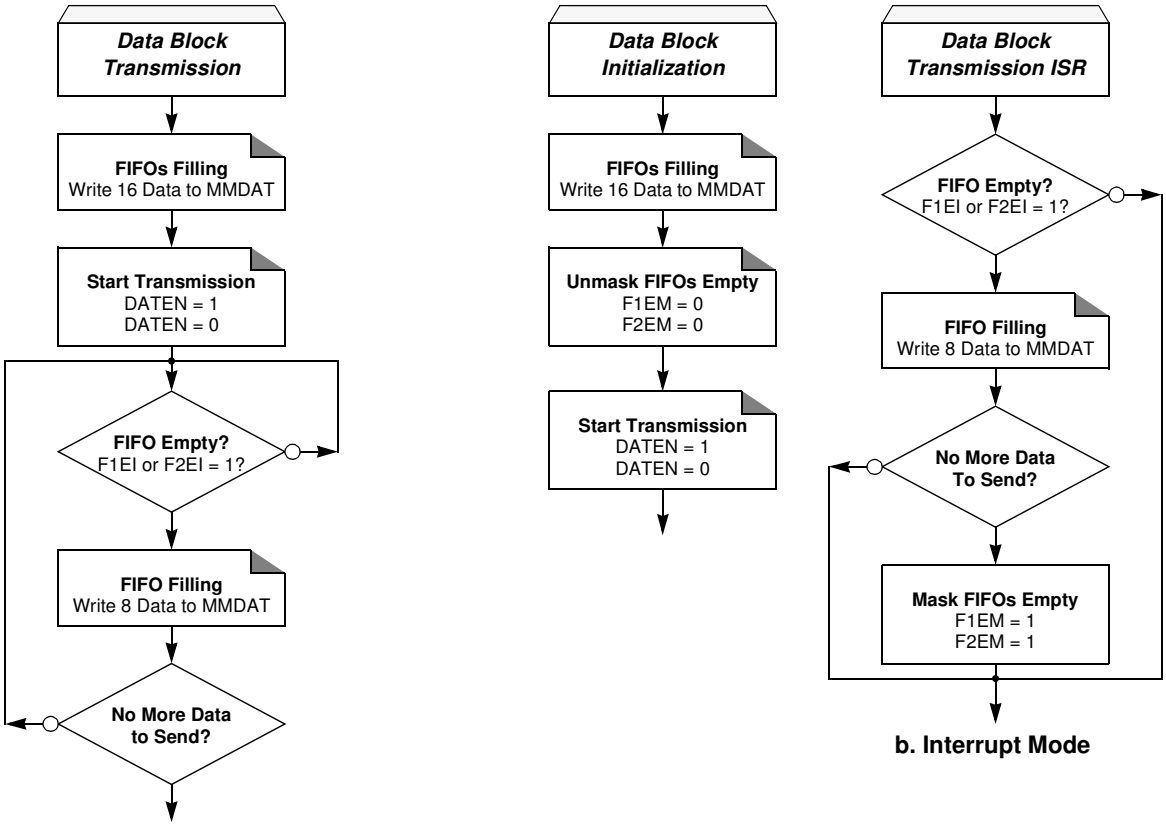
Figure 16-16. Data Stream Transmission Flows



a. Polling Mode

b. Interrupt Mode

Figure 16-17. Data Block Transmission Flows



16.6.4 Data Receiver

16.6.4.1 Configuration

To receive data from the card, the user must first configure the data controller in reception mode by clearing the DATDIR bit in MMCON1 register.

Figure 16-18 summarizes the data stream reception flows in both polling and interrupt modes while Figure 16-19 summarizes the data block reception flows in both polling and interrupt modes, these flows assume that block length is greater than 16 Bytes.

16.6.4.2 Data Reception

The end of data frame (block or stream) reception is signalled by the EOFI flag in MMINT register. This flag may generate an MMC interrupt request as detailed in Section "Interrupt", page 96. When this flag is set, two other flags in MMSTA register: DATFS and CRC16S give a status on the frame received. DATFS indicates if the frame format is correct or not: a valid End bit has been received, and CRC16S indicates if the CRC16 computation is correct or not. In case of data stream CRC16S has no meaning and stays cleared.

According to the MMC specification data transmission, the card starts after the access time delay (formally N_{AC} parameter) beginning from the End bit of the read command. To avoid any locking of the MMC controller when card does not send its data (e.g. physically removed from the bus), the user must launch a time-out period to exit from such situation. In case of time-out

the user may reset the data controller and its internal state machine by setting and clearing the DCR bit in MMCON2 register.

This time-out may be disarmed after receiving 8 data (F1FI flag set) or after receiving end of frame (EOFI flag set) in case of block length less than 8 data (1, 2 or 4).

16.6.4.3 Data Reading

Data is read from the FIFO by reading to MMDAT register. Each time one FIFO becomes full (F1FI or F2FI set), user is requested to flush this FIFO by reading 8 data.

Figure 16-18. Data Stream Reception Flows

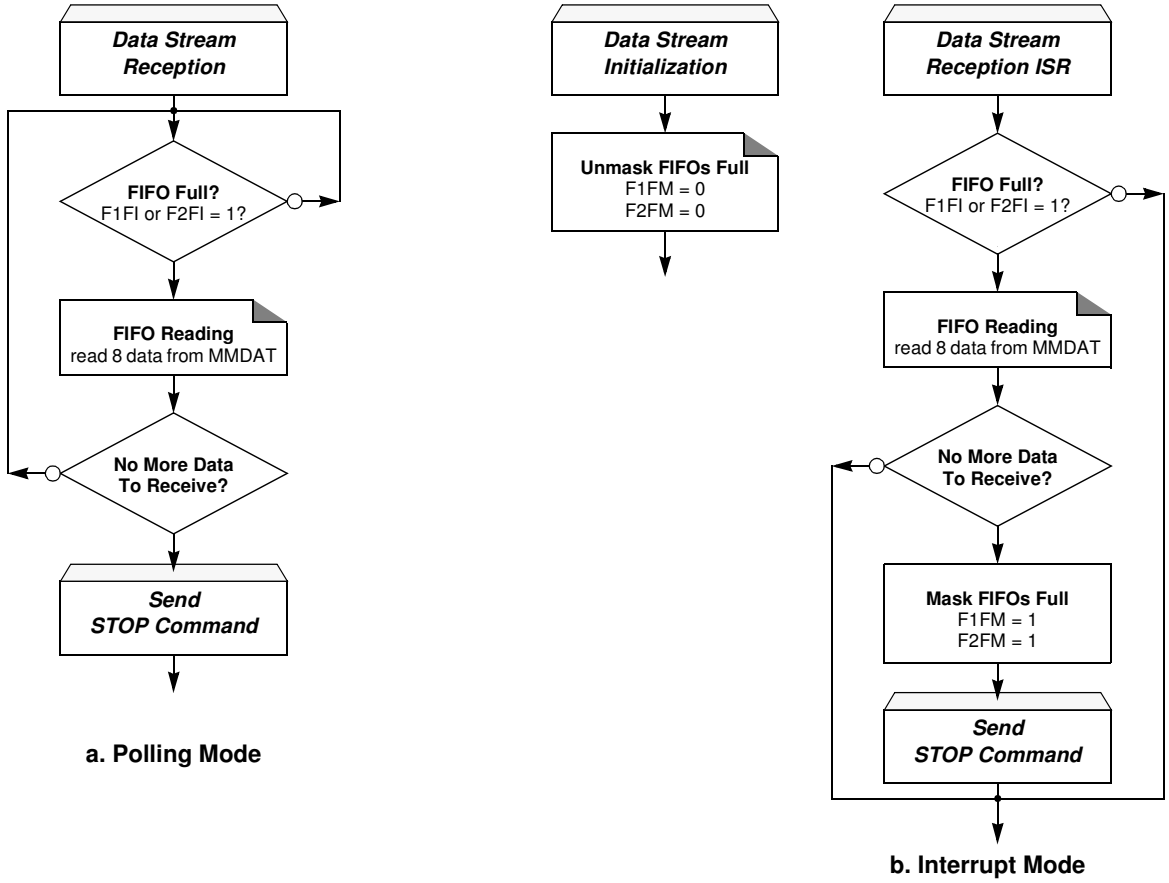
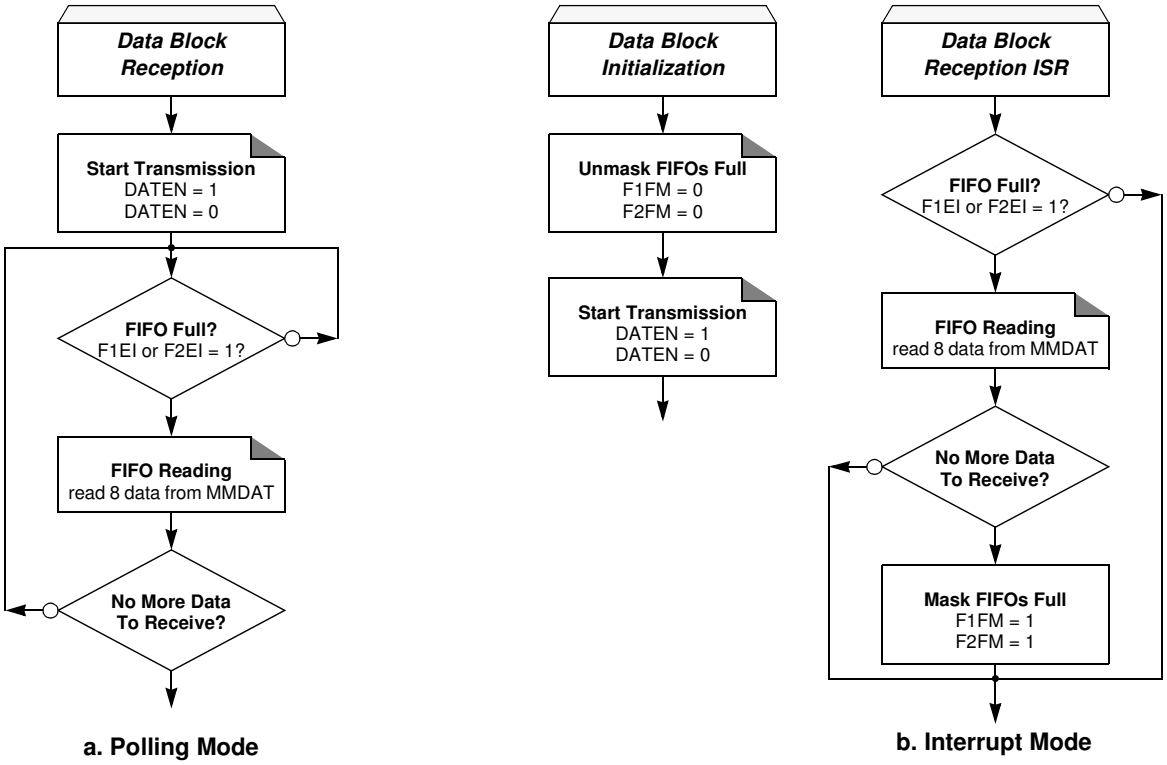


Figure 16-19. Data Block Reception Flows



a. Polling Mode **b. Interrupt Mode**

16.6.5 Flow Control

To allow transfer at high speed without taking care of CPU oscillator frequency, the FLOWC bit in MMCON2 allows control of the data flow in both transmission and reception.

During transmission, setting the FLOWC bit has the following effects:

- MMCLK is stopped when both FIFOs become empty: F1EI and F2EI set.
- MMCLK is restarted when one of the FIFOs becomes full: F1EI or F2EI cleared.

During reception, setting the FLOWC bit has the following effects:

- MMCLK is stopped when both FIFOs become full: F1FI and F2FI set.
- MMCLK is restarted when one of the FIFOs becomes empty: F1FI or F2FI cleared.

As soon as the clock is stopped, the MMC bus is frozen and remains in its state until the clock is restored by writing or reading data in MMDAT.

16.7 Interrupt

16.7.1 Description

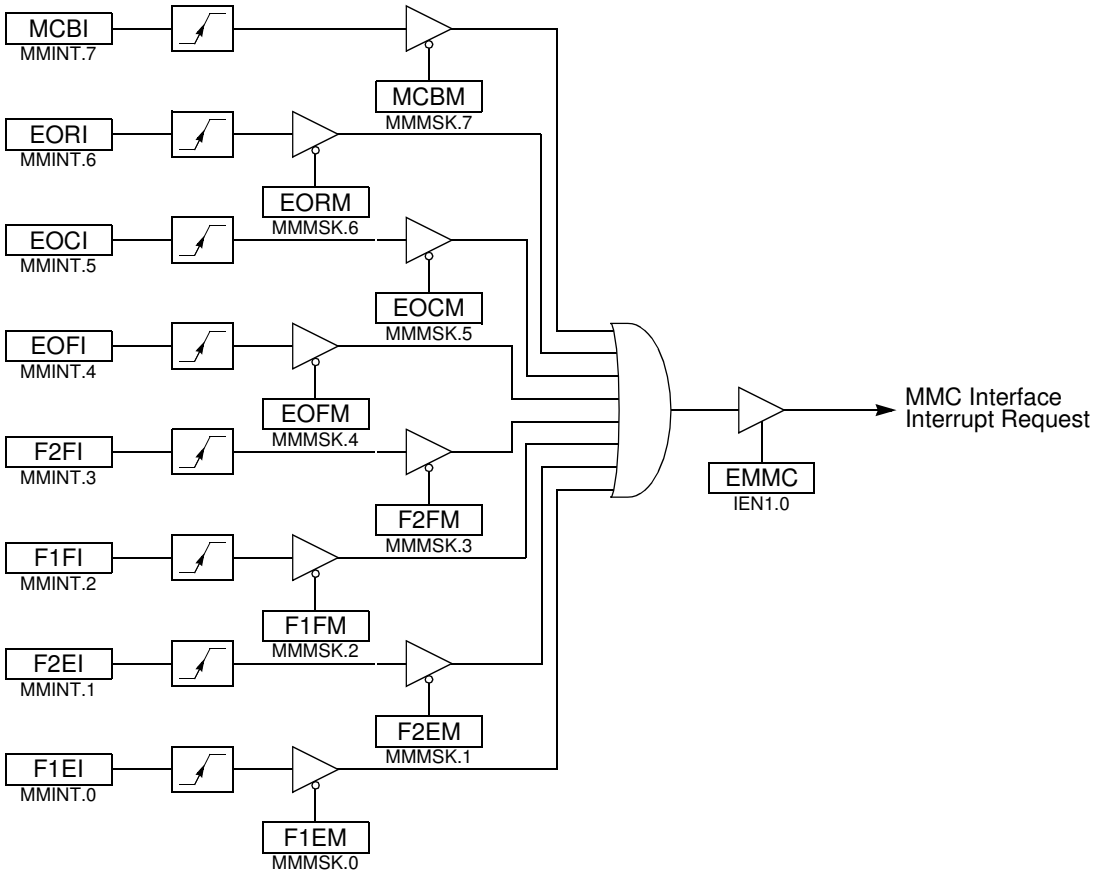
As shown in Figure 16-20, the MMC controller implements eight interrupt sources reported in MCBI, EORI, EOCI, EOFI, F2FI, F1FI, and F2EI flags in MMCINT register. These flags were detailed in the previous sections.

All of these sources are maskable separately using MCBM, EORM, EOCM, EOFM, F2FM, F1FM, and F2EM mask bits, respectively, in MMMSK register.

The interrupt request is generated each time an unmasked flag is set, and the global MMC controller interrupt enable bit is set (EMMC in IEN1 register).

Reading the MMINT register automatically clears the interrupt flags (acknowledgment). This implies that register content must be saved and tested interrupt flag by interrupt flag to be sure not to overlook any interrupts.

Figure 16-20. MMC Controller Interrupt System



16.8 Registers

Table 78. MMCON0 Register

MMCON0 (S:E4h) – MMC Control Register 0

7	6	5	4	3	2	1	0
DRPTR	**DTPTR**	**CRPTR**	**CTPTR**	**MBLOCK**	**DFMT**	**RFMT**	**CRCDIS**

Bit Number	Bit Mnemonic	Description
7	DRPTR	**Data Receive Pointer Reset Bit** Set to reset the read pointer of the data FIFO. Clear to release the read pointer of the data FIFO.

Bit Number	Bit Mnemonic	Description
6	DTPTR	**Data Transmit Pointer Reset Bit** Set to reset the write pointer of the data FIFO. Clear to release the write pointer of the data FIFO.
5	CRPTR	**Command Receive Pointer Reset Bit** Set to reset the read pointer of the receive command FIFO. Clear to release the read pointer of the receive command FIFO.
4	CTPTR	**Command Transmit Pointer Reset Bit** Set to reset the write pointer of the transmit command FIFO. Clear to release the read pointer of the transmit command FIFO.
3	MBLOCK	**Multi-block Enable Bit** Set to select multi-block data format. Clear to select single block data format.
2	DFMT	**Data Format Bit** Set to select the block-oriented data format. Clear to select the stream data format.
1	RFMT	**Response Format Bit** Set to select the 48-bit response format. Clear to select the 136-bit response format.
0	CRCDIS	**CRC7 Disable Bit** Set to disable the CRC7 computation when receiving a response. Clear to enable the CRC7 computation when receiving a response.

Reset Value = 0000 0000b

Table 79. MMCON1 Register

MMCON1 (S:E5h) – MMC Control Register 1

7	6	5	4	3	2	1	0
BLEN3	**BLEN2**	**BLEN1**	**BLEN0**	**DATDIR**	**DATEN**	**RESPEN**	**CMDEN**

Bit Number	Bit Mnemonic	Description
7 - 4	BLEN3:0	**Block Length Bits** Refer to Table 77 for Bits description. Do not program value > 1011b.
3	DATDIR	**Data Direction Bit** Set to select data transfer from host to card (write mode). Clear to select data transfer from card to host (read mode).
2	DATEN	**Data Transmission Enable Bit** Set and clear to enable data transmission immediately or after response has been received.
1	RESPEN	**Response Enable Bit** Set and clear to enable the reception of a response following a command transmission.
0	CMDEN	**Command Transmission Enable Bit** Set and clear to enable transmission of the command FIFO to the card.

Reset Value = 0000 0000b

Table 80. MMCON2 Register

MMCON2 (S:E6h) – MMC Control Register 2

7	6	5	4	3	2	1	0
MMCEN	**DCR**	**CCR**	**-**	**-**	**DATD1**	**DATD0**	**FLOWC**

Bit Number	Bit Mnemonic	Description
7	MMCEN	**MMC Clock Enable Bit** Set to enable the MCLK clocks and activate the MMC controller. Clear to disable the MMC clocks and freeze the MMC controller.
6	DCR	**Data Controller Reset Bit** Set and clear to reset the data line controller in case of transfer abort.
5	CCR	**Command Controller Reset Bit** Set and clear to reset the command line controller in case of transfer abort.
4 - 3	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
2 - 1	DATD1:0	**Data Transmission Delay Bits** Used to delay the data transmission after a response from 3 MMC clock periods (all Bits cleared) to 9 MMC clock periods (all Bits set) by step of 2 MMC clock periods.
0	FLOWC	**MMC Flow Control Bit** Set to enable the flow control during data transfers. Clear to disable the flow control during data transfers.

Reset Value = 0000 0000b

Table 81. MMSTA Register

MMSTA (S:DEh Read Only) – MMC Control and Status Register

7	6	5	4	3	2	1	0
-	**-**	**CBUSY**	**CRC16S**	**DATFS**	**CRC7S**	**RESPFS**	**CFLCK**

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
5	CBUSY	**Card Busy Flag** Set by hardware when the card sends a busy state on the data line. Cleared by hardware when the card no more sends a busy state on the data line.

Bit Number	Bit Mnemonic	Description
4	CRC16S	**CRC16 Status Bit** **Transmission mode** Set by hardware when the token response reports a good CRC. Cleared by hardware when the token response reports a bad CRC. **Reception mode** Set by hardware when the CRC16 received in the data block is correct. Cleared by hardware when the CRC16 received in the data block is not correct.
3	DATFS	**Data Format Status Bit** **Transmission mode** Set by hardware when the format of the token response is correct. Cleared by hardware when the format of the token response is not correct. **Reception mode** Set by hardware when the format of the frame is correct. Cleared by hardware when the format of the frame is not correct.
2	CRC7S	**CRC7 Status Bit** Set by hardware when the CRC7 computed in the response is correct. Cleared by hardware when the CRC7 computed in the response is not correct. This bit is not relevant when CRCDIS is set.
1	RESPFS	**Response Format Status Bit** Set by hardware when the format of a response is correct. Cleared by hardware when the format of a response is not correct.
0	CFLCK	**Command FIFO Lock Bit** Set by hardware to signal user not to write in the transmit command FIFO: busy state. Cleared by hardware to signal user the transmit command FIFO is available: idle state.

Reset Value = 0000 0000b

Table 82. MMINT Register

MMINT (S:E7h Read Only) – MMC Interrupt Register

7	6	5	4	3	2	1	0
MCBI	**EORI**	**EOCI**	**EOFI**	**F2FI**	**F1FI**	**F2EI**	**F1EI**

Bit Number	Bit Mnemonic	Description
7	MCBI	**MMC Card Busy Interrupt Flag** Set by hardware when the card enters or exits its busy state (when the busy signal is asserted or deasserted on the data line). Cleared when reading MMINT.

Bit Number	Bit Mnemonic	Description
6	EORI	**End of Response Interrupt Flag** Set by hardware at the end of response reception. Cleared when reading MMINT.
5	EOCI	**End of Command Interrupt Flag** Set by hardware at the end of command transmission. Clear when reading MMINT.
4	EOFI	**End of Frame Interrupt Flag** Set by hardware at the end of frame (stream or block) transfer. Clear when reading MMINT.
3	F2FI	**FIFO 2 Full Interrupt Flag** Set by hardware when second FIFO becomes full. Cleared by hardware when second FIFO becomes empty.
2	F1FI	**FIFO 1 Full Interrupt Flag** Set by hardware when first FIFO becomes full. Cleared by hardware when first FIFO becomes empty.
1	F2EI	**FIFO 2 Empty Interrupt Flag** Set by hardware when second FIFO becomes empty. Cleared by hardware when second FIFO becomes full.
0	F1EI	**FIFO 1 Empty Interrupt Flag** Set by hardware when first FIFO becomes empty. Cleared by hardware when first FIFO becomes full.

Reset Value = 0000 0011b

Table 83. MMMSK Register

MMMSK (S:DFh) – MMC Interrupt Mask Register

7	6	5	4	3	2	1	0
MCBM	**EORM**	**EOCM**	**EOFM**	**F2FM**	**F1FM**	**F2EM**	**F1EM**

Bit Number	Bit Mnemonic	Description
7	MCBM	**MMC Card Busy Interrupt Mask Bit** Set to prevent MCBI flag from generating an MMC interrupt. Clear to allow MCBI flag to generate an MMC interrupt.
6	EORM	**End Of Response Interrupt Mask Bit** Set to prevent EORI flag from generating an MMC interrupt. Clear to allow EORI flag to generate an MMC interrupt.
5	EOCM	**End Of Command Interrupt Mask Bit** Set to prevent EOCI flag from generating an MMC interrupt. Clear to allow EOCI flag to generate an MMC interrupt.
4	EOFM	**End Of Frame Interrupt Mask Bit** Set to prevent EOFI flag from generating an MMC interrupt. Clear to allow EOFI flag to generate an MMC interrupt.
3	F2FM	**FIFO 2 Full Interrupt Mask Bit** Set to prevent F2FI flag from generating an MMC interrupt. Clear to allow F2FI flag to generate an MMC interrupt.

Bit Number	Bit Mnemonic	Description
2	F1FM	**FIFO 1 Full Interrupt Mask Bit** Set to prevent F1FI flag from generating an MMC interrupt. Clear to allow F1FI flag to generate an MMC interrupt.
1	F2EM	**FIFO 2 Empty Interrupt Mask Bit** Set to prevent F2EI flag from generating an MMC interrupt. Clear to allow F2EI flag to generate an MMC interrupt.
0	F1EM	**FIFO 1 Empty Interrupt Mask Bit** Set to prevent F1EI flag from generating an MMC interrupt. Clear to allow F1EI flag to generate an MMC interrupt.

Reset Value = 1111 1111b

Table 84. MMCMD Register

MMCMD (S:DDh) – MMC Command Register

7	6	5	4	3	2	1	0
MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0

Bit Number	Bit Mnemonic	Description
7 - 0	MC7:0	**MMC Command Receive Byte** Output (read) register of the response FIFO. **MMC Command Transmit Byte** Input (write) register of the command FIFO.

Reset Value = 1111 1111b

Table 85. MMDAT Register

MMDAT (S:DCh) – MMC Data Register

7	6	5	4	3	2	1	0
MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

Bit Number	Bit Mnemonic	Description
7 - 0	MD7:0	**MMC Data Byte** Input (write) or output (read) register of the data FIFO.

Reset Value = 1111 1111b

Table 86. MMCLK Register

MMCLK (S:EDh) – MMC Clock Divider Register

7	6	5	4	3	2	1	0
MMCD7	MMCD6	MMCD5	MMCD4	MMCD3	MMCD2	MMCD1	MMCD0

Bit Number	Bit Mnemonic	Description
7 - 0	MMCD7:0	**MMC Clock Divider** 8-bit divider for MMC clock generation.

Reset Value = 0000 0000b

17. IDE/ATAPI Interface

The AT89C5132 provide an IDE/ATAPI interface allowing connection of devices such as CD-ROM reader, CompactFlash cards, hard disk drive, etc. It consists of a 16-bit data transfer (read or write) between the AT89C5132 and the IDE devices.

17.1 Description

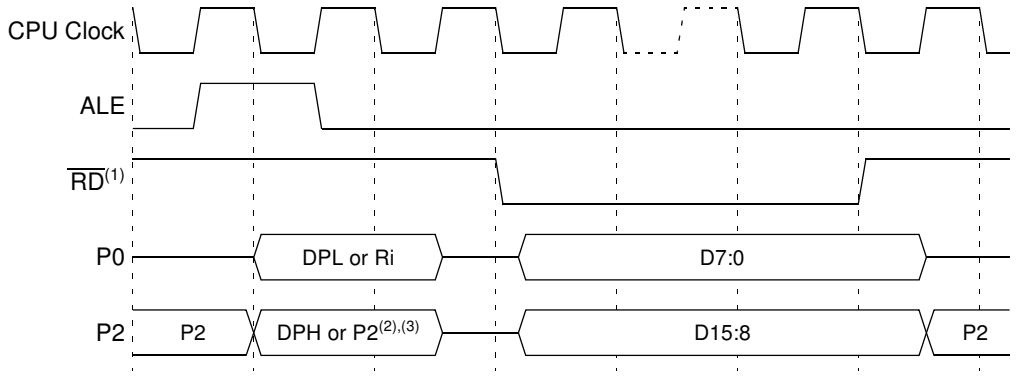
The IDE interface mode is enabled by setting the EXT16 bit in AUXR (see Table 14 on page 27). As soon as this bit is set, all MOVX instructions read or write are done in a 16-bit mode compare to the standard 8-bit mode. P0 carries the low order multiplexed address and data bus (A7:0, D7:0) while P2 carries the high order multiplexed address and data bus (A15:8, D15:8). When writing data in IDE mode, the ACC contains D7:0 data (as in 8-bit mode) while DAT16H register (see Table 88) contains D15:8 data. When reading data in IDE mode, D7:0 data is returned in ACC while D15:8 data is returned in DAT16H.

Figure 17-1 shows the IDE read bus cycle while Figure 17-2 shows the IDE write bus cycle. For simplicity, these figures depict the bus cycle waveforms in idealized form and do not provide precise timing information. For IDE bus cycle timing parameters refer to the Section “AC Characteristics”.

IDE cycle takes 6 CPU clock periods which is equivalent to 12 oscillator clock periods in standard mode or 6 oscillator clock periods in X2 mode. For further information on X2 mode, refer to the Section “X2 Feature”, page 12.

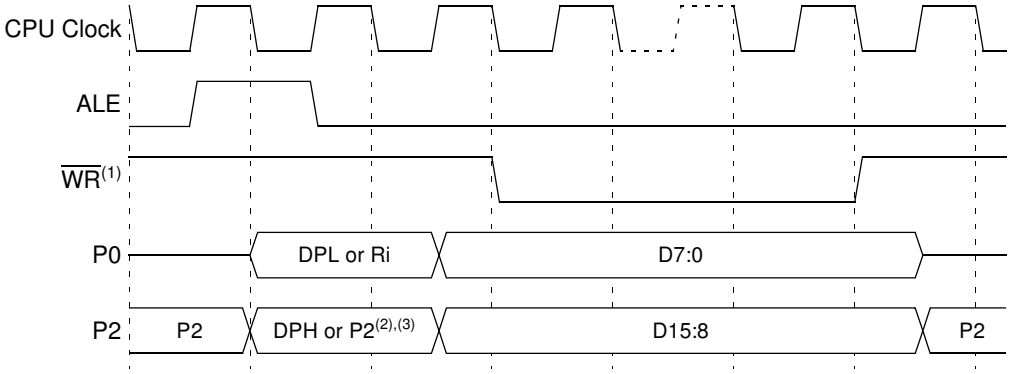
Slow IDE devices can be accessed by stretching the read and write cycles. This is done using the M0 bit in AUXR. Setting this bit changes the width of the $\overline{RD}$ and $\overline{WR}$ signals from 3 to 15 CPU clock periods.

Figure 17-1. IDE Read Waveforms



Notes:
1. $\overline{RD}$ signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.
3. When executing MOVX @DPTR instruction, if DPHDIS is set (Page Access Mode), P2 outputs SFR content instead of DPH.

Figure 17-2. IDE Write Waveforms



Notes: 1. $\overline{WR}$ signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.
3. When executing MOVX @DPTR instruction, if DPHDIS is set (Page Access Mode), P2 outputs SFR content instead of DPH.

17.1.1 IDE Device Connection

Figure 17-3 and Figure 17-4 show two examples on how to interface up to two IDE devices to the AT89C5132. In both examples P0 carries IDE low order data bits D7:0, P2 carries IDE high order data bits D15:8, while $\overline{RD}$ and $\overline{WR}$ signals are respectively connected to the IDE nIOR and nIOW signals. Other IDE control signals are generated by the external address latch outputs in the first example while they are generated by some port I/Os in the second one. Using an external latch will achieve higher transfer rate.

Figure 17-3. IDE Device Connection Example 1

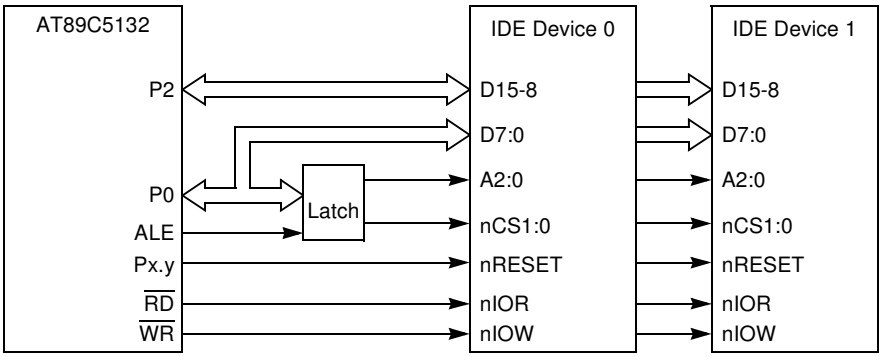


Figure 17-4. IDE Device Connection Example 2

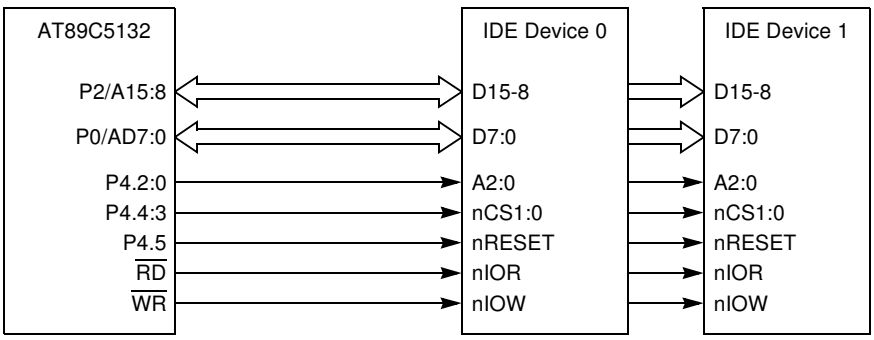


Table 87. External Data Memory Interface Signals

Signal Name	Type	Description	Alternate Function
A15:8	I/O	**Address Lines** Upper address lines for the external bus. Multiplexed higher address and data lines for the IDE interface.	P2.7:0
AD7:0	I/O	**Address/Data Lines** Multiplexed lower address and data lines for the IDE interface.	P0.7:0
ALE	O	**Address Latch Enable** ALE signals indicates that valid address information is available on lines AD7:0.	-
$\overline{RD}$	O	**Read** Read signal output to external data memory.	P3.7
$\overline{WR}$	O	**Write** Write signal output to external memory.	P3.6

17.2 Registers

Table 88. DAT16H Register

DAT16H (S:F9h) – Data 16 High Order Byte

7	6	5	4	3	2	1	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**

Bit Number	Bit Mnemonic	Description
7 - 0	D15:8	**Data 16 High Order Byte** When EXT16 bit is set, DAT16H is set by software with the high order data byte prior any MOVX write instruction. When EXT16 bit is set, DAT16H contains the high order data byte after any MOVX read instruction.

Reset Value = 0000 0000b

18. Serial I/O Port

The serial I/O port in the AT89C5132 provides both synchronous and asynchronous communication modes. It operates as a Synchronous Receiver and Transmitter in one single mode (Mode 0) and operates as an Universal Asynchronous Receiver and Transmitter (UART) in three full-duplex modes (modes 1, 2 and 3). Asynchronous modes support framing error detection and multiprocessor communication with automatic address recognition.

18.1 Mode Selection

SM0 and SM1 Bits in SCON register (see Figure 91) are used to select a mode among the single synchronous and the three asynchronous modes according to Table 89.

Table 89. Serial I/O Port Mode Selection

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Synchronous Shift Register	Fixed/Variable
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fixed
1	1	3	9-bit UART	Variable

18.2 Baud Rate Generator

Depending on the mode and the source selection, the baud rate can be generated from either the Timer 1 or the Internal Baud Rate Generator. The Timer 1 can be used in Modes 1 and 3 while the Internal Baud Rate Generator can be used in Modes 0, 1
and 3.

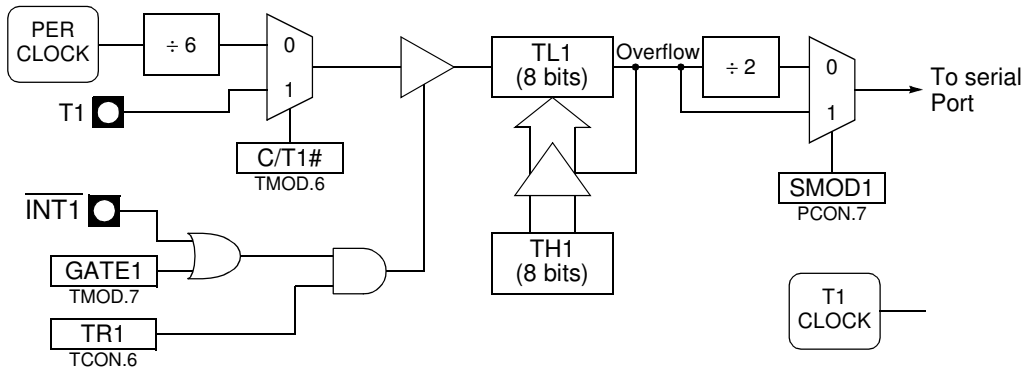
The addition of the Internal Baud Rate Generator allows freeing of the Timer 1 for other purposes in the application. It is highly recommended to use the Internal Baud Rate Generator as it allows higher and more accurate baud rates than Timer 1.

Baud rate formulas depend on the modes selected and are given in the following mode sections.

18.2.1 Timer 1

When using Timer 1, the Baud Rate is derived from the overflow of the timer. As shown in Figure 18-1 Timer 1 is used in its 8-bit auto-reload mode (detailed in Section "Mode 2 (8-bit Timer with Auto-Reload)", page 51). SMOD1 bit in PCON register allows doubling of the generated baud rate.

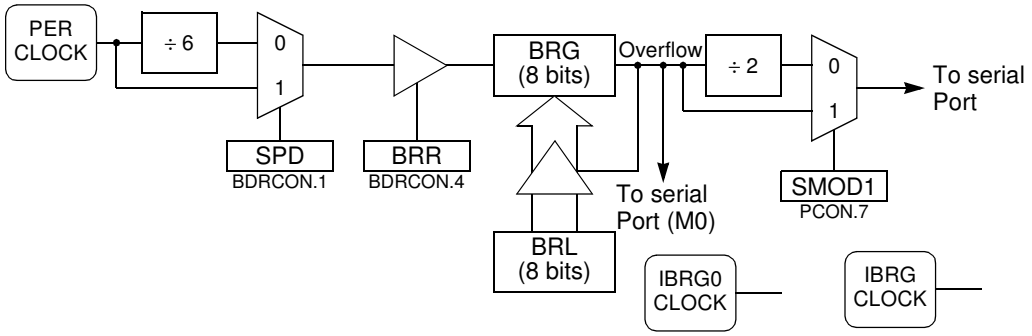
Figure 18-1. Timer 1 Baud Rate Generator Block Diagram



18.2.2 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 18-2, 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 95). 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.

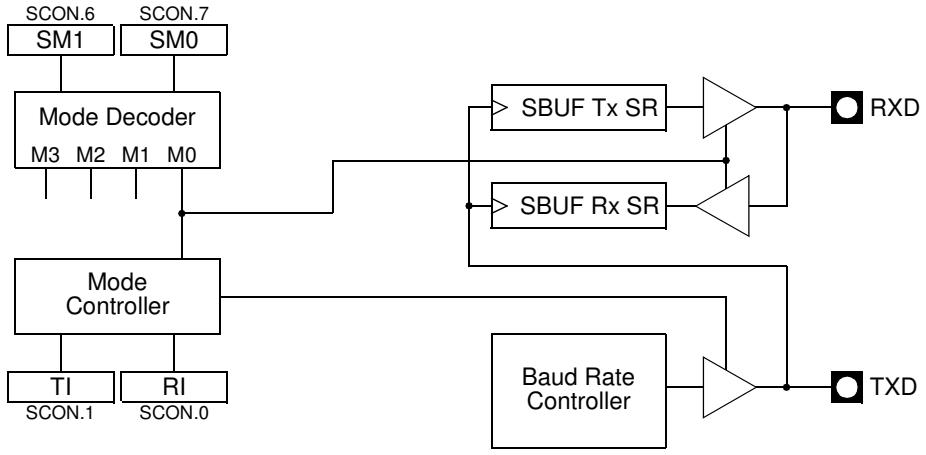
Figure 18-2. Internal Baud Rate Generator Block Diagram



18.3 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)", page 110). Figure 18-3 shows the serial port block diagram in Mode 0.

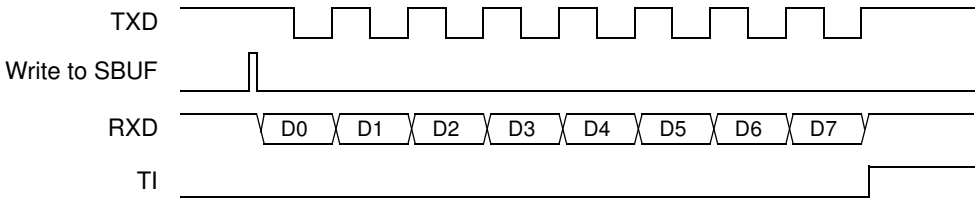
Figure 18-3. Serial I/O Port Block Diagram (Mode 0)



18.3.1 Transmission (Mode 0)

To start a transmission mode 0, write to SCON register clearing Bits SM0, SM1.
As shown in Figure 18-4, 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 18-4. Transmission Waveforms (Mode 0)

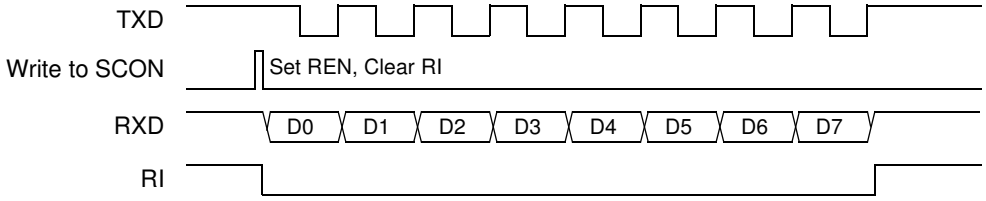


18.3.2 Reception (Mode 0)

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 18-5, 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 18-5. Reception Waveforms (Mode 0)



18.3.3 Baud Rate Selection (Mode 0)

In mode 0, the baud rate can be either fixed or variable.
As shown in Figure 18-6, the selection is done using M0SRC bit in BDRCON register.
Figure 18-7 gives the baud rate calculation formulas for each baud rate source.

Figure 18-6. Baud Rate Source Selection (mode 0)

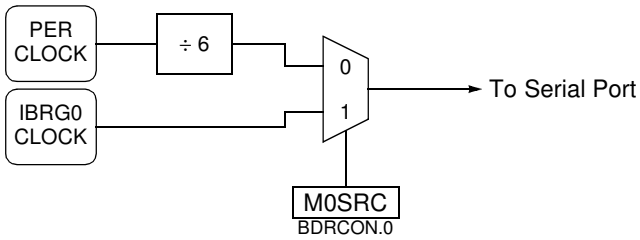


Figure 18-7. Baud Rate Formulas (Mode 0)

$$\text{Baud_Rate} = \frac{F_{PER}}{6}$$

a. Fixed Formula

$$\text{Baud_Rate} = \frac{F_{PER}}{6^{(1-SPD)} \cdot 16 \cdot (256 - BRL)}$$

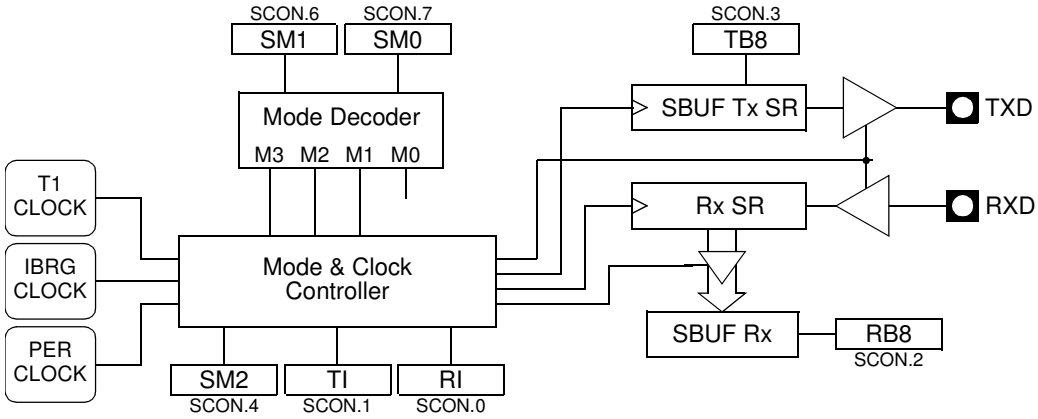
$$BRL = 256 - \frac{F_{PER}}{6^{(1-SPD)} \cdot 16 \cdot \text{Baud_Rate}}$$

b. Variable Formula

18.4 Asynchronous Modes (Modes 1, 2 and 3)

The Serial Port has one 8-bit and two 9-bit asynchronous modes of operation. Figure 18-8 shows the Serial Port block diagram in asynchronous modes.

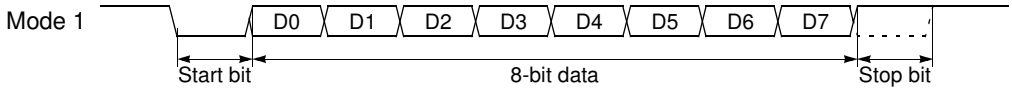
Figure 18-8. Serial I/O Port Block Diagram (Modes 1, 2 and 3)



18.4.0.1 Mode 1

Mode 1 is a full-duplex, asynchronous mode. The data frame (see Figure 18-9) 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 data is received, the stop bit is read in the RB8 bit in SCON register.

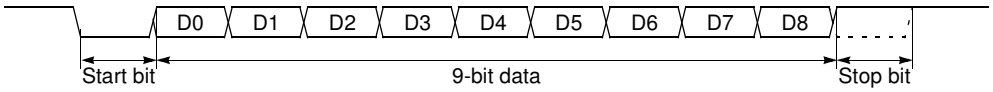
Figure 18-9. Data Frame Format (Mode 1)



18.4.0.2 Modes 2 and 3

Modes 2 and 3 are full-duplex, asynchronous modes. The data frame (see Figure 18-10) 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, the ninth bit can be used as a command/data flag.

Figure 18-10. Data Frame Format (Modes 2 and 3)



18.4.1 Transmission (Modes 1, 2 and 3)

To initiate a transmission, write to SCON register, setting SM0 and SM1 Bits according to Table 89, and setting the ninth bit by writing to TB8 bit. Then, writing the byte to be transmitted to SBUF register starts the transmission.

18.4.2 Reception (Modes 1, 2 and 3)

To prepare for reception, write to SCON register, setting SM0 and SM1 Bits according to Table 89, and set the REN bit. The actual reception is then initiated by a detected high-to-low transition on the RXD pin.

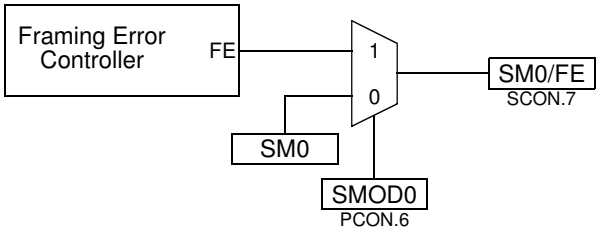
18.4.3 Framing Error Detection (Modes 1, 2 and 3)

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 18-11.

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 clears 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 18-17.

Figure 18-11. Framing Error Block Diagram



18.4.4 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 18-12, the selection is done using RBCK and TBCK Bits in BDRCON register.

Figure 18-13 gives the baud rate calculation formulas for each baud rate source. Table 90 details Internal Baud Rate Generator configuration for different peripheral clock frequencies and gives baud rates closer to the standard baud rates.

Figure 18-12. Baud Rate Source Selection (Modes 1 and 3)

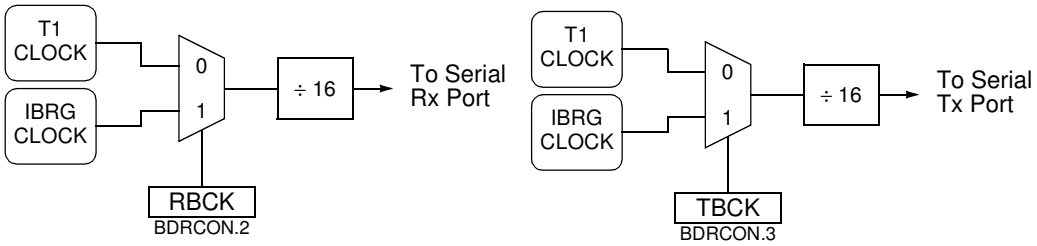


Figure 18-13. Baud Rate Formulas (Modes 1 and 3)

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

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

a. IBRG Formula

$$Baud_Rate = \frac{2^{SMOD1} \cdot F_{PER}}{6 \cdot 32 \cdot (256 - TH1)}$$

$$TH1 = 256 - \frac{2^{SMOD1} \cdot F_{PER}}{192 \cdot Baud_Rate}$$

b. T1 Formula

Table 90. Baud Rate Generator Configuration

Baud Rate	F_{PER} = 6 MHz(1)				F_{PER} = 8 MHz(1)				F_{PER} = 10 MHz(1)			
	SPD	SMOD 1	BRL	Error %	SPD	SMOD 1	BRL	Error %	SPD	SMOD 1	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_{PER} = 12 MHz(2)				F_{PER} = 16 MHz(2)				F_{PER} = 20 MHz(2)			
	SPD	SMOD 1	BRL	Error %	SPD	SMOD 1	BRL	Error %	SPD	SMOD 1	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, $F_{PER} = F_{OSC} \div 2$.
2. These frequencies are achieved in X2 mode, $F_{PER} = F_{OSC}$.

18.4.5 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 18-14 the selection is done using SMOD1 bit in PCON register. Figure 18-15 gives the baud rate calculation formula depending on the selection.

Figure 18-14. Baud Rate Generator Selection (mode 2)

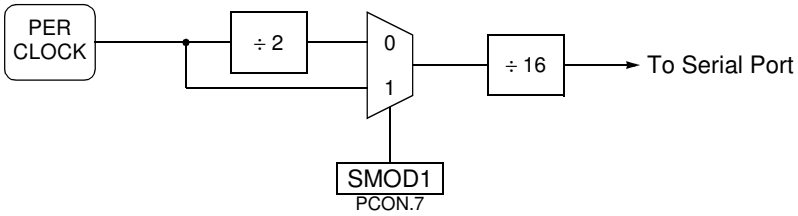


Figure 18-15. Baud Rate Formula (Mode 2)

$$\text{Baud_Rate} = \frac{2^{SMOD1} \cdot F_{PER}}{32}$$

18.5 Multiprocessor Communication (Modes 2 and 3)

Modes 2 and 3 provide a ninth-bit mode to facilitate multiprocessor communication. To enable this feature, set SM2 bit in SCON register. When the multiprocessor communication feature is enabled, the Serial Port can differentiate between data frames (ninth bit clear) and address frames (ninth bit set). This allows the AT89C5132 to function as a slave processor in an environment where multiple slave processors share a single serial line.

When the multiprocessor communication feature is enabled, the receiver ignores frames with the ninth bit clear. The receiver examines frames with the ninth bit set for an address match. If the received address matches the slaves address, the receiver hardware sets RB8 and RI bits in SCON register, generating an interrupt.

The addressed slave's software then clears SM2 bit in SCON register and prepares to receive the data Bytes. The other slaves are unaffected by these data Bytes because they are waiting to respond to their own addresses.

18.6 Automatic Address Recognition

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

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

If desired, the automatic address recognition feature in mode 1 may be enabled. 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).

18.6.1 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:

```
SADDR = 0101 0110b
SADEN = 1111 1100b
Given = 0101 01XXb
```

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

```
Slave A:  SADDR = 1111 0001b
          SADEN = 1111 1010b
          Given = 1111 0X0Xb
Slave B:  SADDR = 1111 0011b
          SADEN = 1111 1001b
          Given = 1111 0XX1b
Slave C:  SADDR = 1111 0011b
          SADEN = 1111 1101b
          Given = 1111 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 0; for slaves B and C, bit 1 is a don't care bit. To communicate with slaves A and B, but not slave C, 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).

18.6.2 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.:

```
SADDR = 0101 0110b
SADEN = 1111 1100b
(SADDR | SADEN)=1111 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:  SADDR = 1111 0001b
          SADEN = 1111 1010b
          Given = 1111 1X11b,

Slave B:  SADDR = 1111 0011b
          SADEN = 1111 1001b
          Given = 1111 1X11b,

Slave C:  SADDR = 1111 0010b
          SADEN = 1111 1101b
          Given = 1111 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 the address FFh.

To communicate with slaves A and B, but not slave C, the master must send the address FBh.

18.6.3 Reset Address

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 is backwards compatible with the 80C51 microcontrollers that do not support automatic address recognition.

18.7 Interrupt

The Serial I/O Port handles two interrupt sources that are the "end of reception" (RI in SCON) and "end of transmission" (TI in SCON) flags. As shown in Figure 18-16 these flags are combined together to appear as a single interrupt source for the C51 core. Flags must be cleared by software when executing the serial interrupt service routine.
The serial interrupt is enabled by setting ES bit in IEN0 register. This assumes interrupts are globally enabled by setting EA bit in IEN0 register.

Depending on the selected mode and whether the framing error detection is enabled or not, RI flag is set during the stop bit or during the ninth bit as detailed in Figure 18-17.

Figure 18-16. Serial I/O Interrupt System

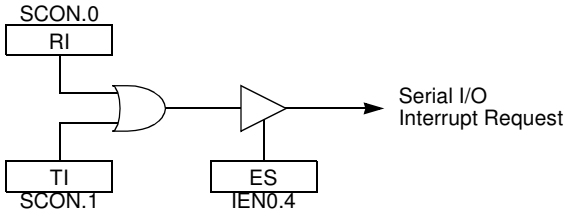
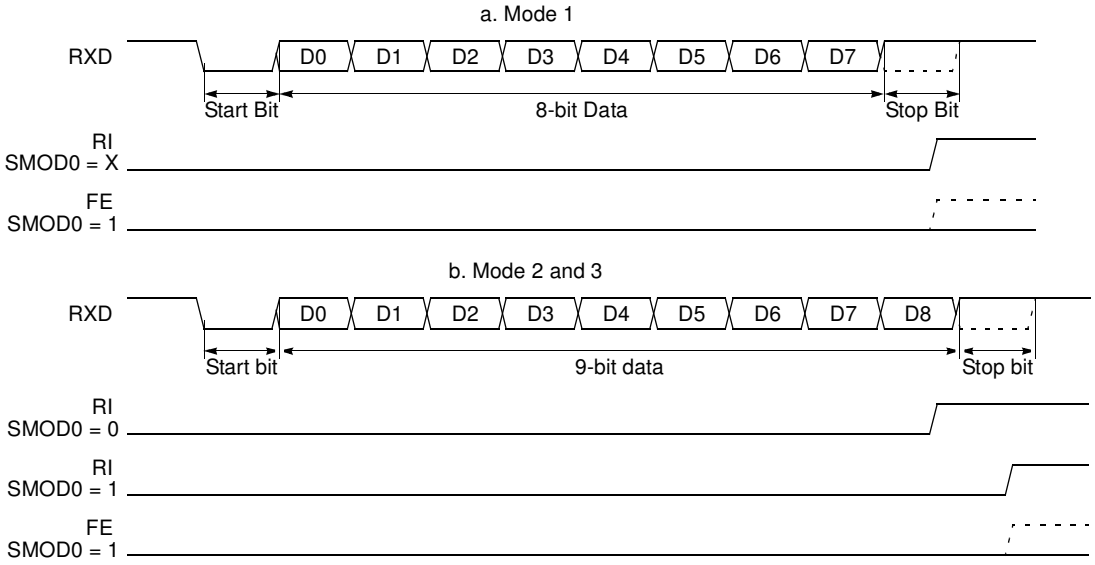


Figure 18-17. Interrupt Waveforms



18.8 Registers

Table 91. SCON Register

SCON (S:98h) – Serial Control Register

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

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error Bit** To select this function, set SMOD0 bit in PCON register. Set by hardware to indicate an invalid stop bit. Must be cleared by software.
	SM0	**Serial Port Mode Bit 0** Refer to Table 89 for mode selection.
6	SM1	**Serial Port Mode Bit 1** Refer to Table 89 for mode selection.
5	SM2	**Serial Port Mode Bit 2** Set to enable the multiprocessor communication and automatic address recognition features. Clear to disable the multiprocessor communication and automatic address recognition features.
4	REN	**Receiver Enable Bit** Set to enable reception. Clear to disable reception.
3	TB8	**Transmit Bit 8** Modes 0 and 1: Not used. Modes 2 and 3: Software writes the ninth data bit to be transmitted to TB8.
2	RB8	**Receiver Bit 8** Mode 0: Not used. Mode 1 (SM2 cleared): Set or cleared by hardware to reflect the stop bit received. Modes 2 and 3 (SM2 set): Set or cleared by hardware to reflect the ninth bit received.
1	TI	**Transmit Interrupt Flag** Set by the transmitter after the last data bit is transmitted. Must be cleared by software.
0	RI	**Receive Interrupt Flag** Set by the receiver after the stop bit of a frame has been received. Must be cleared by software.

Reset Value = 0000 0000b

Table 92. SBUF Register

SBUF (S:99h) – Serial Buffer Register

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

Bit Number	Bit Mnemonic	Description
7 - 0	SD7:0	**Serial Data Byte** Read the last data received by the Serial I/O Port. Write the data to be transmitted by the Serial I/O Port.

Reset value = XXXX XXXXb

Table 93. SADDR Register

SADDR (S:A9h) – Slave Individual Address Register

7	6	5	4	3	2	1	0
SAD7	SAD6	SAD5	SAD4	SAD3	SAD2	SAD1	SAD0

Bit Number	Bit Mnemonic	Description
7 - 0	SAD7:0	**Slave Individual Address.**

Reset Value = 0000 0000b

Table 94. SADEN Register

SADEN (S:B9h) – Slave Individual Address Mask Byte Register

7	6	5	4	3	2	1	0
SAE7	SAE6	SAE5	SAE4	SAE3	SAE2	SAE1	SAE0

Bit Number	Bit Mnemonic	Description
7 - 0	SAE7:0	**Slave Address Mask Byte.**

Reset Value = 0000 0000b

Table 95. BDRCON Register

BDRCON (S:92h) – Baud Rate Generator Control Register

7	6	5	4	3	2	1	0
-	-	-	BRR	TBCK	RBCK	SPD	M0SRC

Bit Number	Bit Mnemonic	Description
7-5	-	**Reserved** The value read from these bits is indeterminate. Do not set these bits.
4	BRR	**Baud Rate Run Bit** Set to enable the baud rate generator. Clear to disable the baud rate generator.
3	TBCK	**Transmission Baud Rate Selection Bit** Set to select the baud rate generator as transmission baud rate generator. Clear to select the Timer 1 as transmission baud rate generator.
2	RBCK	**Reception Baud Rate Selection Bit** Set to select the baud rate generator as reception baud rate generator. Clear to select the Timer 1 as reception baud rate generator.
1	SPD	**Baud Rate Speed Bit** Set to select high speed baud rate generation. Clear to select low speed baud rate generation.
0	M0SRC	**Mode 0 Baud Rate Source Bit** Set to select the variable baud rate generator in Mode 0. Clear to select fixed baud rate in Mode 0.

Reset Value = XXX0 0000b

Table 96. BRL Register

BRL (S:91h) – Baud Rate Generator Reload Register

7	6	5	4	3	2	1	0
BRL7	**BRL6**	**BRL5**	**BRL4**	**BRL3**	**BRL2**	**BRL1**	**BRL0**

Bit Number	Bit Mnemonic	Description
7-0	BRL7:0	**Baud Rate Reload Value.**

Reset Value = 0000 0000b

19. Synchronous Peripheral Interface

The AT89C5132 implement a Synchronous Peripheral Interface with master and slave modes capability.

Figure 19-1 shows an SPI bus configuration using the AT89C5132 as master connected to slave peripherals. Figure 19-2 shows an SPI bus configuration using the AT89C5132 as slave of an other master.
The bus is made of three wires connecting all the devices together:

- Master Output Slave Input (MOSI): it is used to transfer data in series from the master to a slave. It is driven by the master.
- Master Input Slave Output (MISO): it is used to transfer data in series from a slave to the master. It is driven by the selected slave.
- Serial Clock (SCK): it is used to synchronize the data transmission both in and out of the devices through their MOSI and MISO lines. It is driven by the master for eight clock cycles which allows to exchange one byte on the serial lines.

Each slave peripheral is selected by one Slave Select pin ($\overline{SS}$). If there is only one slave, it may be continuously selected with $\overline{SS}$ tied to a low level. Otherwise, the AT89C5132 may select each device by software through port pins (Pn.x). Special care should be taken not to select two slaves at the same time to avoid bus conflicts.

Figure 19-1. Typical Master SPI Bus Configuration

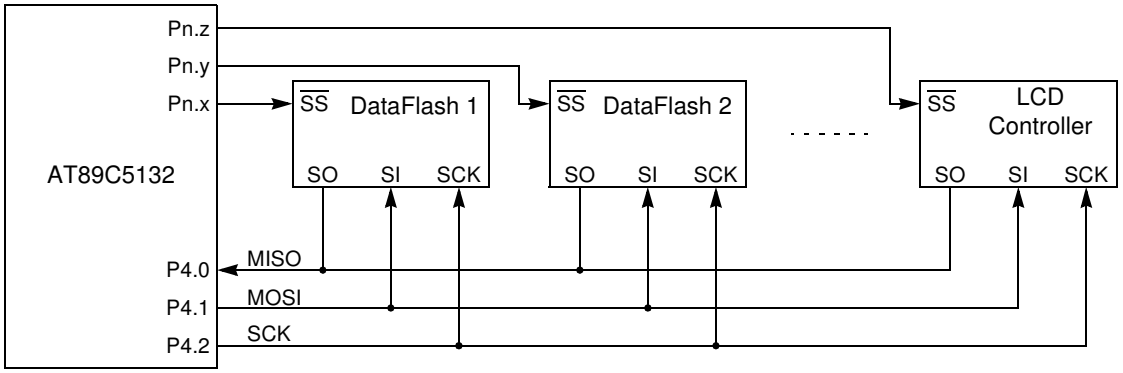
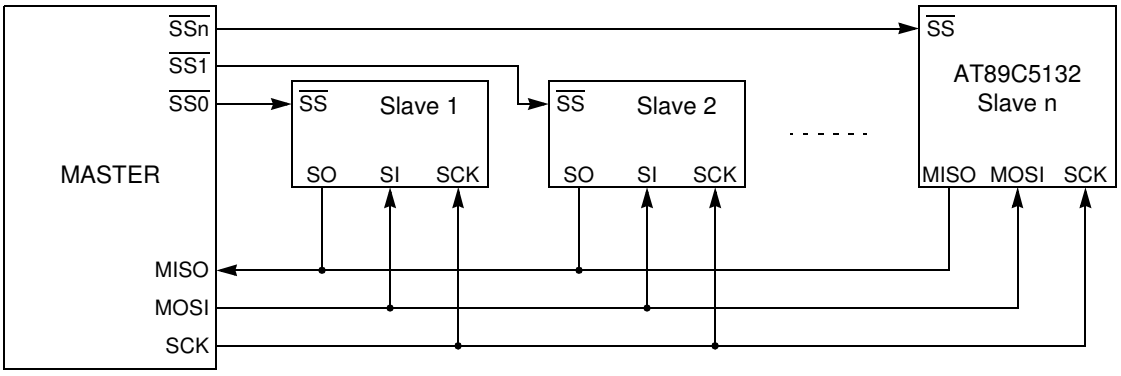


Figure 19-2. Typical Slave SPI Bus Configuration



19.1 Description

The SPI controller interfaces with the C51 core through three special function registers: SPCON, the SPI control register (see Table 98); SPSTA, the SPI status register (see Table 99); and SPDAT, the SPI data register (see Table 100).

19.1.1 Master Mode

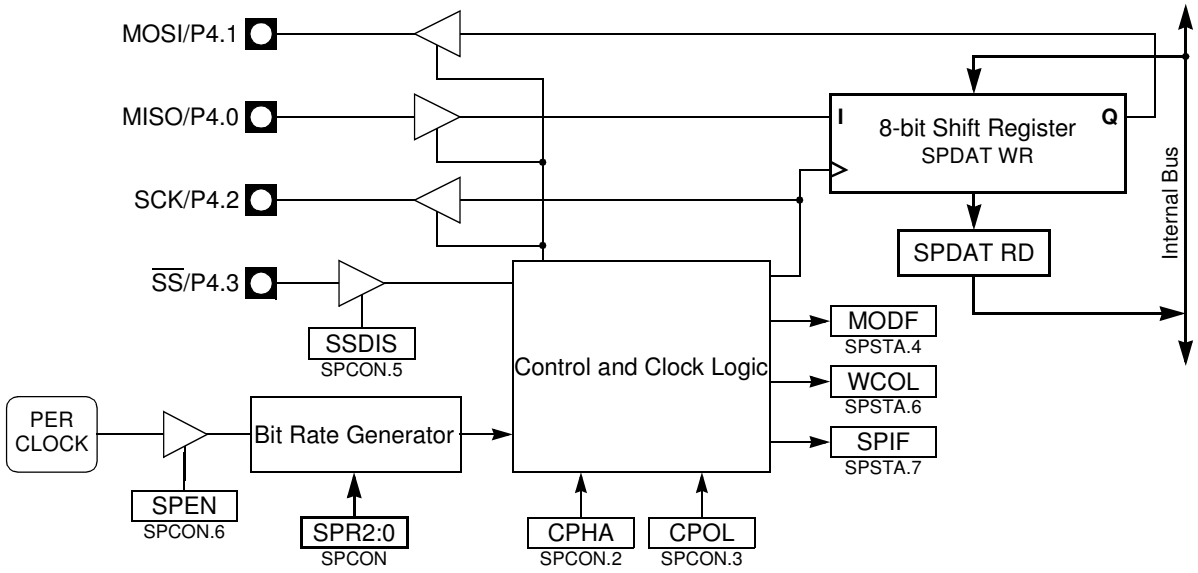
The SPI operates in master mode when the MSTR bit in SPCON is set.

Figure 19-3 shows the SPI block diagram in master mode. Only a master SPI module can initiate transmissions. Software begins the transmission by writing to SPDAT. Writing to SPDAT writes to the shift register while reading SPDAT reads an intermediate register updated at the end of each transfer.

The byte begins shifting out on the MOSI pin under the control of the bit rate generator. This generator also controls the shift register of the slave peripheral through the SCK output pin. As the byte shifts out, another byte shifts in from the slave peripheral on the MISO pin. The byte is transmitted most significant bit (MSB) first. The end of transfer is signalled by SPIF being set.

In case of the AT89C5132 is the only master on the bus, it can be useful not to use $\overline{SS}$ pin and get it back to I/O functionality. This is achieved by setting SSDIS bit in SPCON.

Figure 19-3. SPI Master Mode Block Diagram



Note: MSTR bit in SPCON is set to select master mode.

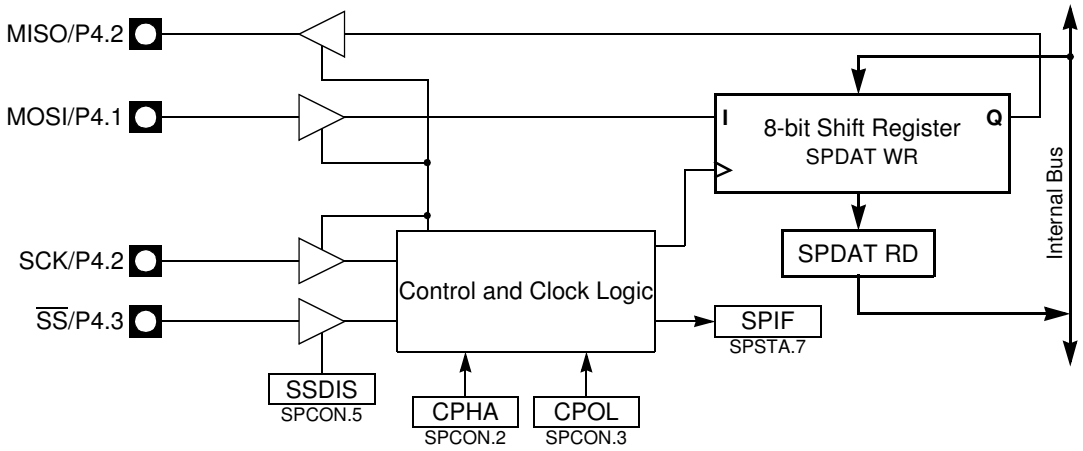
19.1.2 Slave Mode

The SPI operates in slave mode when the MSTR bit in SPCON is cleared and data has been loaded in SPDAT.

Figure 19-4 shows the SPI block diagram in slave mode. In slave mode, before data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be asserted to low level. $\overline{SS}$ must remain low until the transmission of the byte is complete. In the slave SPI module, data enters the shift register through the MOSI pin under the control of the serial clock provided by the master SPI module on the SCK input pin. When the master starts a transmission, the data in the shift register begins shifting out on the MISO pin. The end of transfer is signaled by SPIF being set.

In case of the AT89C5132 is the only slave on the bus, it can be useful not to use SS pin and get it back to I/O functionality. This is achieved by setting SSDIS bit in SPCON. This bit has no effect when CPHA is cleared (see Section "SS Management", page 123).

Figure 19-4. SPI Slave Mode Block Diagram



Note: MSTR bit in SPCON is cleared to select slave mode.

19.1.3 Bit Rate

The bit rate can be selected from seven predefined bit rates using the SPR2, SPR1 and SPR0 control Bits in SPCON according to Table 97. These bit rates are derived from the peripheral clock (F_{PER}) issued from the Clock Controller block as detailed in Section “Clock Controller”, page 12.

Table 97. Serial Bit Rates

			Bit Rate (kHz) Vs F_{PER}						
SPR2	SPR1	SPR0	6 MHz(1)	8 MHz(1)	10 MHz(1)	12 MHz(2)	16 MHz(2)	20 MHz(2)	F_{PER} Divider
0	0	0	3000	4000	5000	6000	8000	10000	2
0	0	1	1500	2000	2500	3000	4000	5000	4
0	1	0	750	1000	1250	1500	2000	2500	8
0	1	1	375	500	625	750	1000	1250	16
1	0	0	187.5	250	312.5	375	500	625	32
1	0	1	93.75	125	156.25	187.5	250	312.5	64
1	1	0	46.875	62.5	78.125	93.75	125	156.25	128
1	1	1	6000	8000	10000	12000	16000	20000	1

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

19.1.4 Data Transfer

The Clock Polarity bit (CPOL in SPCON) defines the default SCK line level in idle state(1) while the Clock Phase bit (CPHA in SPCON) defines the edges on which the input data are sampled and the edges on which the output data are shifted (see Figure 19-5 and Figure 19-6). The SI signal is output from the selected slave and the SO signal is the output from the master. The

AT89C5132 captures data from the SI line while the selected slave captures data from the SO line.

For simplicity, the following figures depict the SPI waveforms in idealized form and do not provide precise timing information. For timing parameters refer to the Section “AC Characteristics”.

Note: 1. When the peripheral is disabled (SPEN = 0), default SCK line is high level.

Figure 19-5. Data Transmission Format (CPHA = 0)

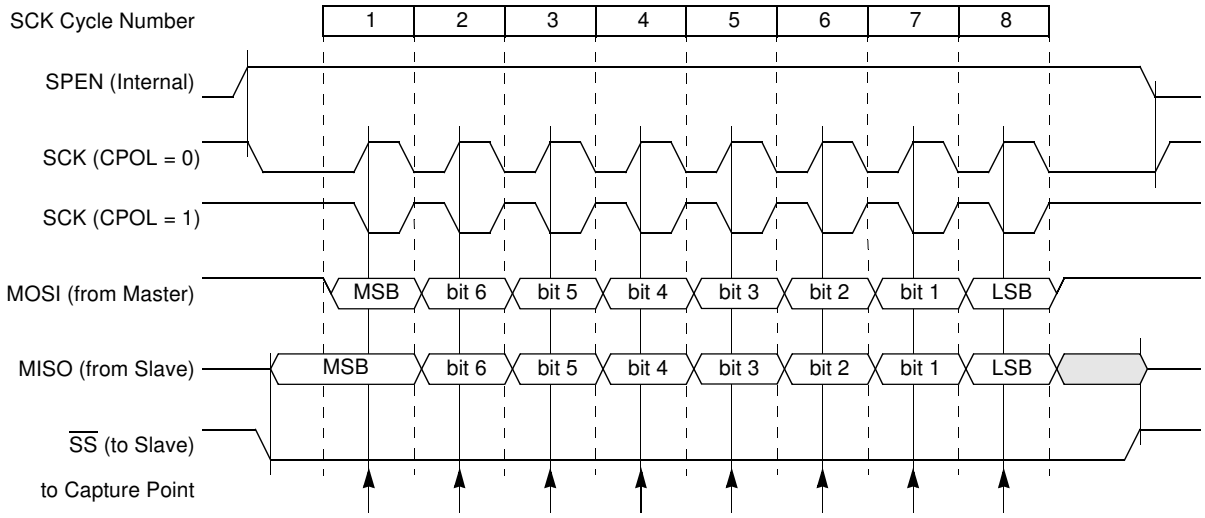
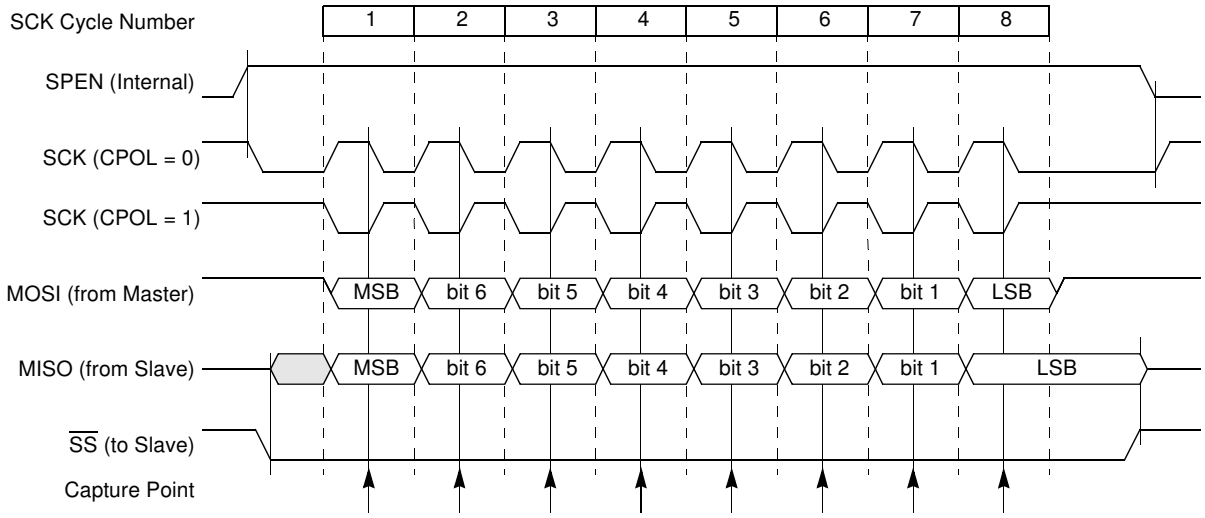


Figure 19-6. Data Transmission Format (CPHA = 1)

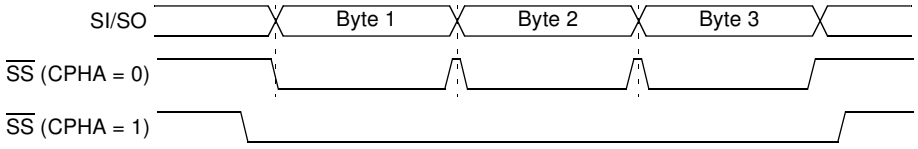


19.1.5 $\overline{SS}$ Management

Figure 19-5 shows an SPI transmission with CPHA = 0, where the first SCK edge is the MSB capture point. Therefore the slave starts to output its MSB as soon as it is selected: $\overline{SS}$ asserted to low level. $\overline{SS}$ must then be deasserted between each byte transmission (see Figure 19-7). SPDAT must be loaded with data before $\overline{SS}$ is asserted again.

Figure 19-6 shows an SPI transmission with CPHA = 1, where the first SCK edge is used by the slave as a start of transmission signal. Therefore $\overline{SS}$ may remain asserted between each byte transmission (see Figure 19-7).

Figure 19-7. $\overline{SS}$ Timing Diagram



19.1.6 Error Conditions

The following flags signal the SPI error conditions:

- MODF in SPSTA signals a mode fault.
  MODF flag is relevant only in master mode when $\overline{SS}$ usage is enabled (SSDIS bit cleared). It signals when set that another master on the bus has asserted $\overline{SS}$ pin and so, may create a conflict on the bus with two masters sending data at the same time.
  A mode fault automatically disables the SPI (SPEN cleared) and configures the SPI in slave mode (MSTR cleared).
  MODF flag can trigger an interrupt as explained in Section "Interrupt", page 124.
  MODF flag is cleared by reading SPSTA and re-configuring SPI by writing to SPCON.
- WCOL in SPSTA signals a write collision.
  WCOL flag is set when SPDAT is loaded while a transfer is on-going. In this case, data is not written to SPDAT and transfer continues uninterrupted. WCOL flag does not trigger any interrupt and is relevant jointly with SPIF flag.
  WCOL flag is cleared after reading SPSTA and writing new data to SPDAT while no transfer is ongoing.

19.2 Interrupt

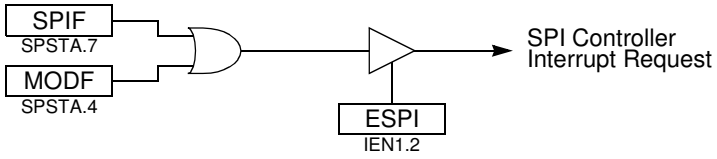
The SPI handles two interrupt sources; the “end of transfer” and the “mode fault” flags.

As shown in Figure 19-8 these flags are combined together to appear as a single interrupt source for the C51 core. The SPIF flag is set at the end of an 8-bit shift in and out and is cleared by reading SPSTA and then reading from or writing to SPDAT.

The MODF flag is set in case of mode fault error and is cleared by reading SPSTA and then writing to SPCON.

The SPI interrupt is enabled by setting ESPI bit in IEN1 register. This assumes interrupts are globally enabled by setting EA bit in IEN0 register.

Figure 19-8. SPI Interrupt System



19.3 Configuration

The SPI configuration is made through SPCON.

19.3.1 Master Configuration

The SPI operates in master mode when the MSTR bit in SPCON is set.

19.3.2 Slave Configuration

The SPI operates in slave mode when the MSTR bit in SPCON is cleared and data has been loaded in SPDAT.

19.3.3 Data Exchange

There are two possible Policies to exchange data in master and slave modes:

- polling
- interrupts

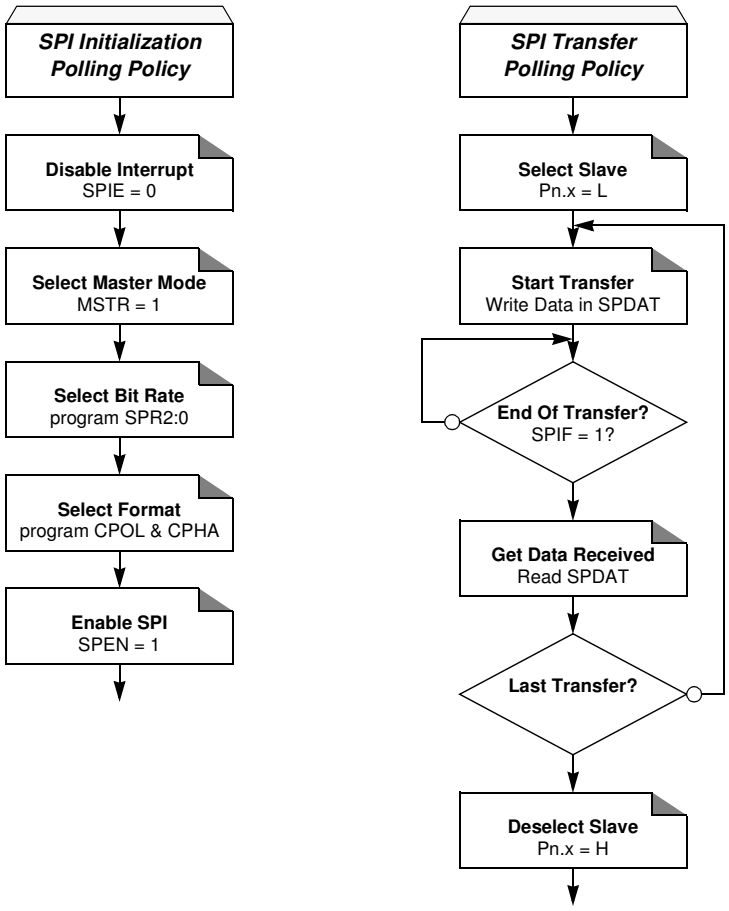
19.3.4 Master Mode with Polling Policy

Figure 19-9 shows the initialization phase and the transfer phase flows using the polling policy. Using this flow prevents any overrun error occurrence.

- The bit rate is selected according to Table 97.
- The transfer format depends on the slave peripheral.
- $\overline{SS}$ may be deasserted between transfers depending also on the slave peripheral.
- SPIF flag is cleared when reading SPDAT (SPSTA has been read before by the “end of transfer” check).

This policy provides the fastest effective transmission and is well adapted when communicating at high speed with other Microcontrollers. However, the procedure may then be interrupted at any time by higher priority tasks.

Figure 19-9. Master SPI Polling Policy Flows



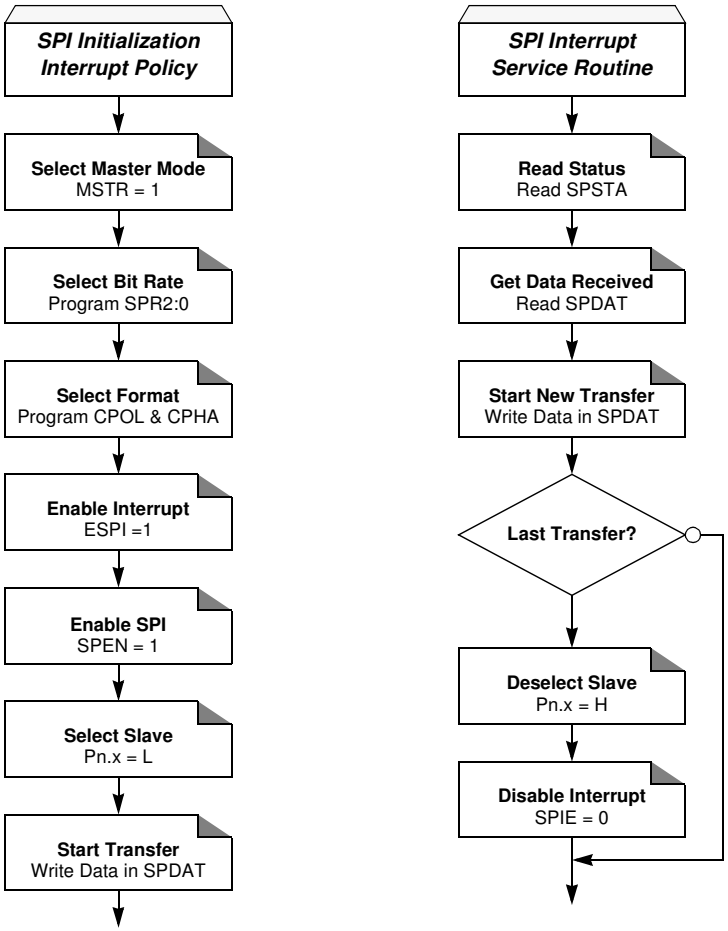
19.3.5 Master Mode with Interrupt Policy

Figure 19-10 shows the initialization phase and the transfer phase flows using the interrupt policy. Using this flow prevents any overrun error occurrence.

- The bit rate is selected according to Table 97.
- The transfer format depends on the slave peripheral.
- $\overline{SS}$ may be deasserted between transfers depending also on the slave peripheral.

Reading SPSTA at the beginning of the ISR is mandatory for clearing the SPIF flag. Clear is effective when reading SPDAT.

Figure 19-10. Master SPI Interrupt Policy Flows



19.3.6 Slave Mode with Polling Policy

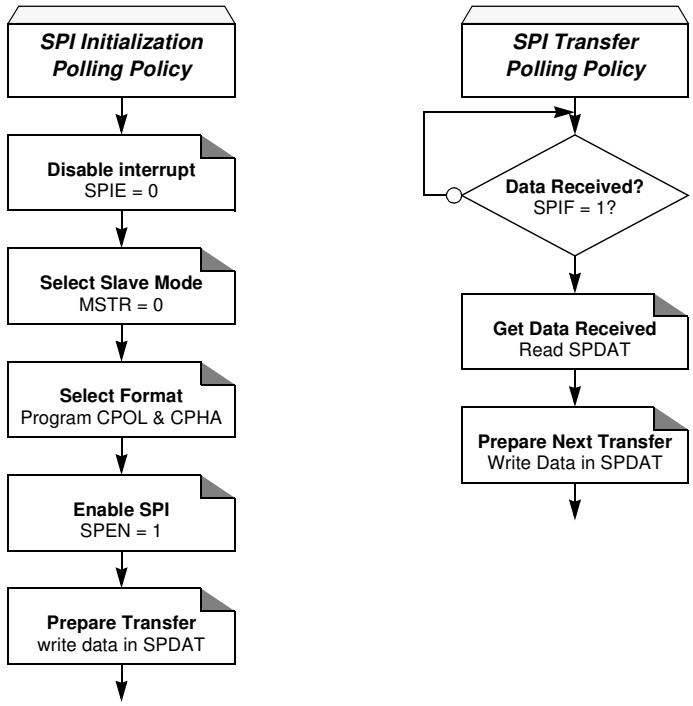
Figure 19-11 shows the initialization phase and the transfer phase flows using the polling policy.

The transfer format depends on the master controller.

SPIF flag is cleared when reading SPDAT (SPSTA has been read before by the “end of reception” check).

This policy provides the fastest effective transmission and is well adapted when communicating at high speed with other Microcontrollers. However, the procedure may be interrupted at any time by higher priority tasks.

Figure 19-11. Slave SPI Polling Policy Flows



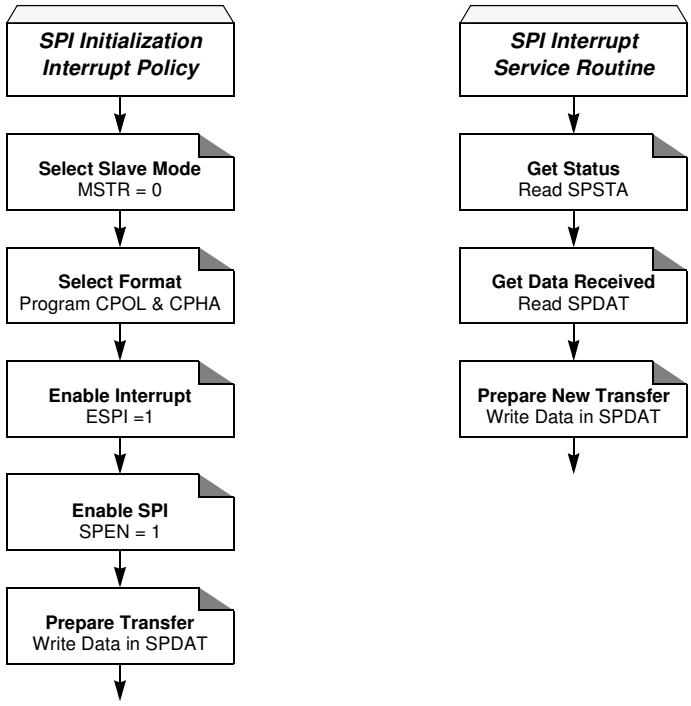
19.3.7 Slave Mode with Interrupt Policy

Figure 19-10 shows the initialization phase and the transfer phase flows using the interrupt policy.

The transfer format depends on the master controller.

Reading SPSTA at the beginning of the ISR is mandatory for clearing the SPIF flag. Clear is effective when reading SPDAT.

Figure 19-12. Slave SPI Interrupt Policy Flows



19.4 Registers

Table 98. SPCON Register

SPCON (S:C3h) – SPI Control Register

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

Bit Number	Bit Mnemonic	Description
7	SPR2	**SPI Rate Bit 2** Refer to Table 97 for bit rate description.
6	SPEN	**SPI Enable Bit** Set to enable the SPI interface. Clear to disable the SPI interface.
5	SSDIS	**Slave Select Input Disable Bit** Set to disable $\overline{SS}$ in both master and slave modes. In slave mode this bit has no effect if CPHA = 0. Clear to enable $\overline{SS}$ in both master and slave modes.
4	MSTR	**Master Mode Select** Set to select the master mode. Clear to select the slave mode.
3	CPOL	SPI Clock Polarity Bit(1) Set to have the clock output set to high level in idle state. Clear to have the clock output set to low level in idle state.
2	CPHA	**SPI Clock Phase Bit** Set to have the data sampled when the clock returns to idle state (see CPOL). Clear to have the data sampled when the clock leaves the idle state (see CPOL).

Bit Number	Bit Mnemonic	Description
1 - 0	SPR1:0	**SPI Rate Bits 0 and 1** Refer to Table 97 for bit rate description.

Reset Value = 0001 0100b

Note: 1. When the SPI is disabled, SCK outputs high level.

Table 99. SPSTA Register

SPSTA (S:C4h) – SPI Status Register

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

Bit Number	Bit Mnemonic	Description
7	SPIF	**SPI Interrupt Flag** Set by hardware when an 8-bit shift is completed. Cleared by hardware when reading or writing SPDAT after reading SPSTA.
6	WCOL	**Write Collision Flag** Set by hardware to indicate that a collision has been detected. Cleared by hardware to indicate that no collision has been detected.
5	-	**Reserved** The values read from this bit is indeterminate. Do not set this bit.
4	MODF	**Mode Fault** Set by hardware to indicate that the $\overline{SS}$ pin is at an appropriate level. Cleared by hardware to indicate that the $\overline{SS}$ pin is at an inappropriate level.
3:0	-	**Reserved** The values read from these Bits are indeterminate. Do not set these Bits.

Reset Value = 00000 0000b

Table 100. SPDAT Register

SPDAT (S:C5h) – Synchronous Serial Data Register

7	6	5	4	3	2	1	0
SPD7	**SPD6**	**SPD5**	**SPD4**	**SPD3**	**SPD2**	**SPD1**	**SPD0**

Bit Number	Bit Mnemonic	Description
7 - 0	SPD7:0	**Synchronous Serial Data**

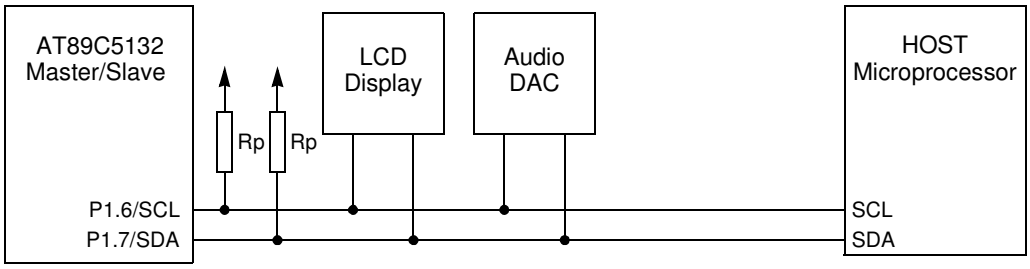
Reset Value = XXXX XXXXb

20. Two-wire Interface (TWI) Controller

The AT89C5132 implements a TWI controller supporting the four standard master and slave modes with multimaster capability. Thus, it allows connection of slave devices like LCD controller, audio DAC, etc., but also external master controlling where the AT89C5132 is used as a peripheral of a host.

The TWI bus is a bi-directional TWI serial communication standard. It is designed primarily for simple but efficient integrated circuit control. The system is comprised of 2 lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. The serial data transfer is limited to 100 Kbit/s in low speed mode, however, some higher bit rates can be achieved depending on the oscillator frequency. Various communication configurations can be designed using this bus. Figure 20-1 shows a typical TWI bus configuration using the AT89C5132 in master and slave modes. All the devices connected to the bus can be master and slave.

Figure 20-1. Typical TWI Bus Configuration



20.1 Description

The CPU interfaces to the TWI logic via the following four 8-bit special function registers: the Synchronous Serial Control register (SSCON SFR, see Table 26), the Synchronous Serial Data register (SSDAT SFR, see Table 28), the Synchronous Serial Status register (SSSTA SFR, see Table 27) and the Synchronous Serial Address register (SSADR SFR, see Table 29).

SSCON is used to enable the controller, to program the bit rate (see Table 26), to enable slave modes, to acknowledge or not a received data, to send a START or a STOP condition on the TWI bus, and to acknowledge a serial interrupt. A hardware reset disables the TWI controller.

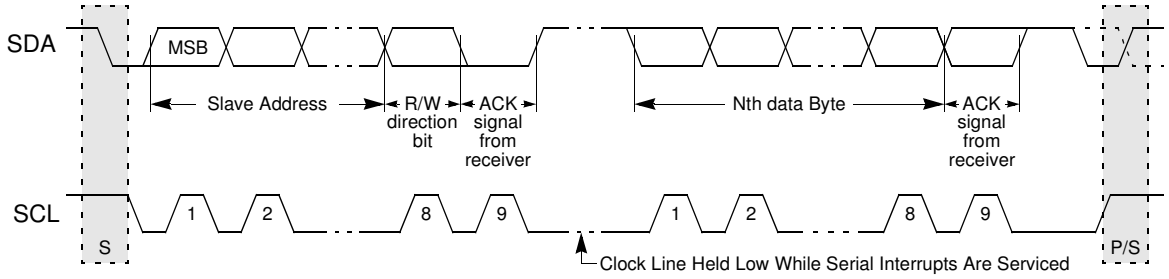
SSSTA contains a status code which reflects the status of the TWI logic and the TWI bus. The three least significant bits are always zero. The five most significant bits contains the status code. There are 26 possible status codes. When SSSTA contains F8h, no relevant state information is available and no serial interrupt is requested. A valid status code is available in SSSTA after SSI is set by hardware and is still present until SSI has been reset by software. Table 20 to Table 20-6 give the status for both master and slave modes and miscellaneous states.

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

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

Figure 20-2 shows how a data transfer is accomplished on the TWI bus.

Figure 20-2. Complete Data Transfer on TWI Bus



The four operating modes are:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

Data transfer in each mode of operation are shown in Figure 20-3 through Figure 20-6. These figures contain the following abbreviations:

A	Acknowledge bit (low level at SDA)
$\overline{A}$	Not acknowledge bit (high level on SDA)
Data	8-bit data Byte
S	START condition
P	STOP condition
MR	Master Receive
MT	Master Transmit
SLA	Slave Address
GCA	General Call Address (00h)
R	Read bit (high level at SDA)
W	Write bit (low level at SDA)

In Figure 20-3 through Figure 20-6, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in SSSTA. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When the serial interrupt routine is entered, the status code in SSSTA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Table 20 through Table 20-6.

20.1.1 Bit Rate

The bit rate can be selected from seven predefined bit rates or from a programmable bit rate generator using the SSCR2, SSCR1, and SSCR0 control bits in SSCON (see Table 26). The predefined bit rates are derived from the peripheral clock (F_{PER}) issued from the Clock Controller block as detailed in Section "Oscillator", page 12, while bit rate generator is based on timer 1 overflow output.

Table 19. Serial Clock Rates

SSCRx			Bit Frequency (kHz)			
2	1	0	F_{PER} = 6 MHz	F_{PER} = 8 MHz	F_{PER} = 10 MHz	F_{PER} Divided By
0	0	0	47	62.5	78.125	128
0	0	1	53.5	71.5	89.3	112
0	1	0	62.5	83	104.2(1)	96
0	1	1	75	100	125(1)	80
1	0	0	12.5	16.5	20.83	480
1	0	1	100	133.3(1)	166.7(1)	60
1	1	0	200(1)	266.7(1)	333.3(1)	30
1	1	1	0.5 < · < 125(1)	0.67 < · < 166.7(1)	0.81 < · < 208.3(1)	96 · (256 – reload value Timer 1)

Note: 1. These bit rates are outside of the low speed standard specification limited to 100 kHz but can be used with high speed TWI components limited to 400 kHz.

20.1.2 Master Transmitter Mode

In the master transmitter mode, a number of data Bytes are transmitted to a slave receiver (see Figure 20-3). Before the master transmitter mode can be entered, SSCON must be initialized as follows:

SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0
Bit Rate	1	0	0	0	X	Bit Rate	Bit Rate

SSCR2:0 define the serial bit rate (see Table 19). SSPE must be set to enable the controller. SSSTA, SSSTO and SSI must be cleared.

The master transmitter mode may now be entered by setting the SSSTA bit. The TWI logic will now monitor the TWI bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SSI bit in SSCON) is set, and the status code in SSSTA is 08h. This status must be used to vector to an interrupt routine that loads SSDAT with the slave address and the data direction bit (SLA+W). The serial interrupt flag (SSI) must then be cleared before the serial transfer can continue.

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

SSPE and SSCR2:0 are not affected by the serial transfer and are not referred to in Table 20. After a repeated START condition (state 10h) the controller may switch to the master receiver mode by loading SSDAT with SLA+R.

20.1.3 Master Receiver Mode

In the master receiver mode, a number of data Bytes are received from a slave transmitter (see Figure 20-4). The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the interrupt routine must load SSDAT with the 7 - bit slave address and the data direction bit (SLA+R). The serial interrupt flag (SSI) must then be cleared before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag is set again and a number of status code in SSSTA are possible. There are 40h, 48h or 38h for the master mode and also 68h, 78h or B0h if the slave mode was enabled (SSAA = logic 1). The appropriate action to be taken for each of these status code is detailed in Table 20-6. This scheme is repeated until a STOP condition is transmitted.

SSPE and SSCR2:0 are not affected by the serial transfer and are not referred to in Table 20-6. After a repeated START condition (state 10h) the controller may switch to the master transmitter mode by loading SSDAT with SLA+W.

20.1.4 Slave Receiver Mode

In the slave receiver mode, a number of data Bytes are received from a master transmitter (see Figure 20-5). To initiate the slave receiver mode, SSADR and SSCON must be loaded as follows:

SSA6	SSA5	SSA4	SSA3	SSA2	SSA1	SSA0	SSGC
←			Own Slave Address			→	X

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

SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0
X	1	0	0	0	1	X	X

SSCR2:0 have no effect in the slave mode. SSPE must be set to enable the controller. The SSAA bit must be set to enable the own slave address or the general call address acknowledgment. SSSTA, SSSTO and SSI must be cleared.

When SSADR and SSCON have been initialized, the controller waits until it is addressed by its own slave address followed by the data direction bit which must be logic 0 (W) for operating in the slave receiver mode. After its own slave address and the W bit has been received, the serial interrupt flag is set and a valid status code can be read from SSSTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status code is detailed in Table 20-6 and Table 24. The slave receiver mode may also be entered if arbitration is lost while the controller is in the master mode (see states 68h and 78h).

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

20.1.5 Slave Transmitter Mode

In the slave transmitter mode, a number of data Bytes are transmitted to a master receiver (see Figure 20-6). Data transfer is initialized as in the slave receiver mode. When SSADR and SSCON have been initialized, the controller waits until it is addressed by its own slave address followed by the data direction bit which must be logic 1 (R) for operating in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag is set and a valid status code can be read from SSSTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status code is detailed in Table 24. The slave transmitter mode may also be entered if arbitration is lost while the controller is in the master mode (see state B0h).

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

20.1.6 Miscellaneous States

There are 2 SSSTA codes that do not correspond to a defined TWI hardware state (see Table 25). These are discussed below.

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

Status 00h indicates that a bus error has occurred during a serial transfer. A bus error is caused when a START or a STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address Byte, a data Byte, or an acknowledge bit. When a bus error occurs, SSI is set. To recover from a bus error, the SSSTO flag must be set and SSI must be cleared. This causes the controller to enter the not addressed slave mode and to clear the SSSTO flag (no other bits in S1CON are affected). The SDA and SCL lines are released and no STOP condition is transmitted.

Note: The TWI controller interfaces to the external TWI bus via 2 port 1 pins: P1.6/SCL (serial clock line) and P1.7/SDA (serial data line). To avoid low level asserting and conflict on these lines when the TWI controller is enabled, the output latches of P1.6 and P1.7 must be set to logic 1.

Figure 20-3. Format and States in the Master Transmitter Mode

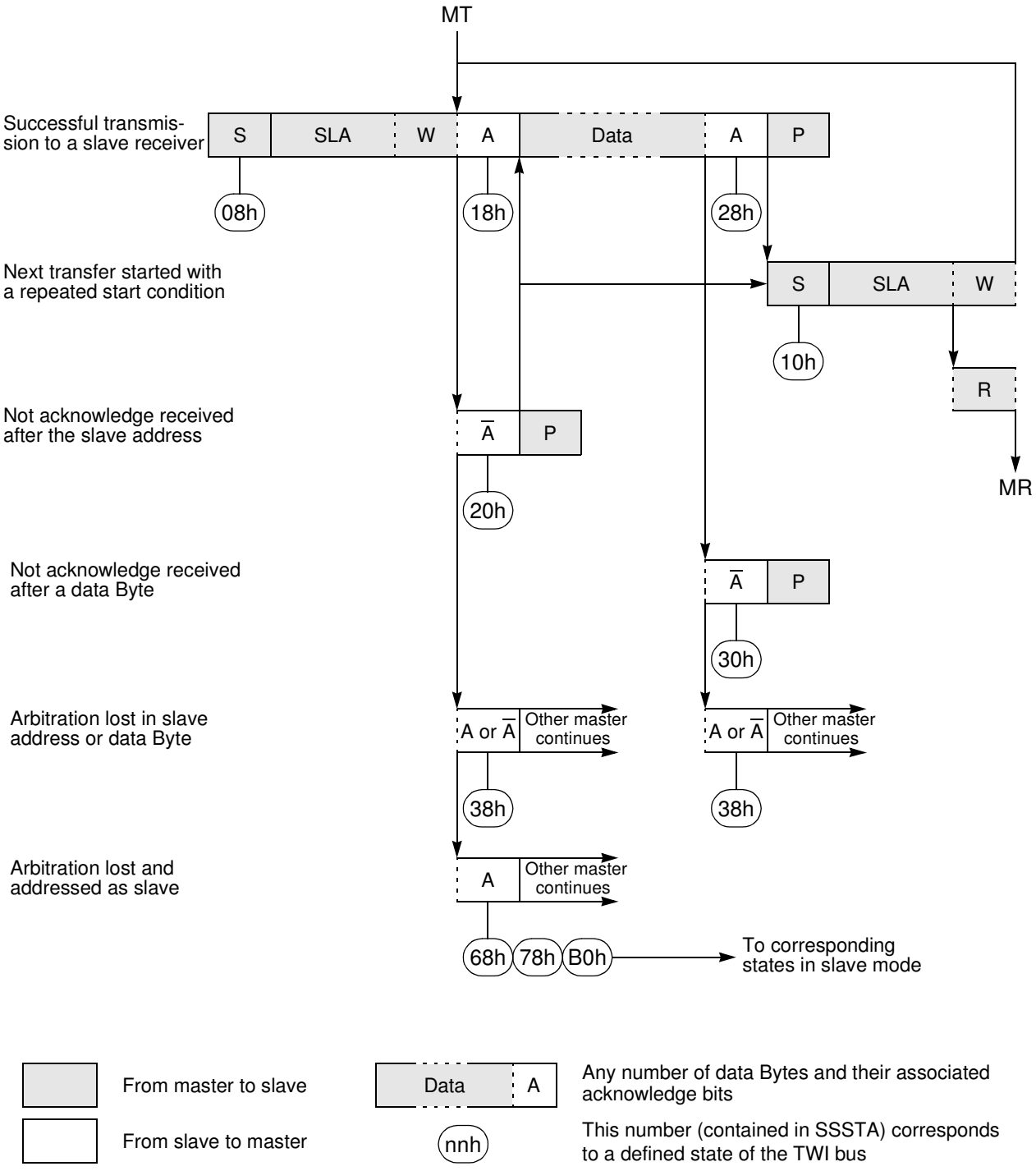


Figure 20-4. Format and States in the Master Receiver Mode

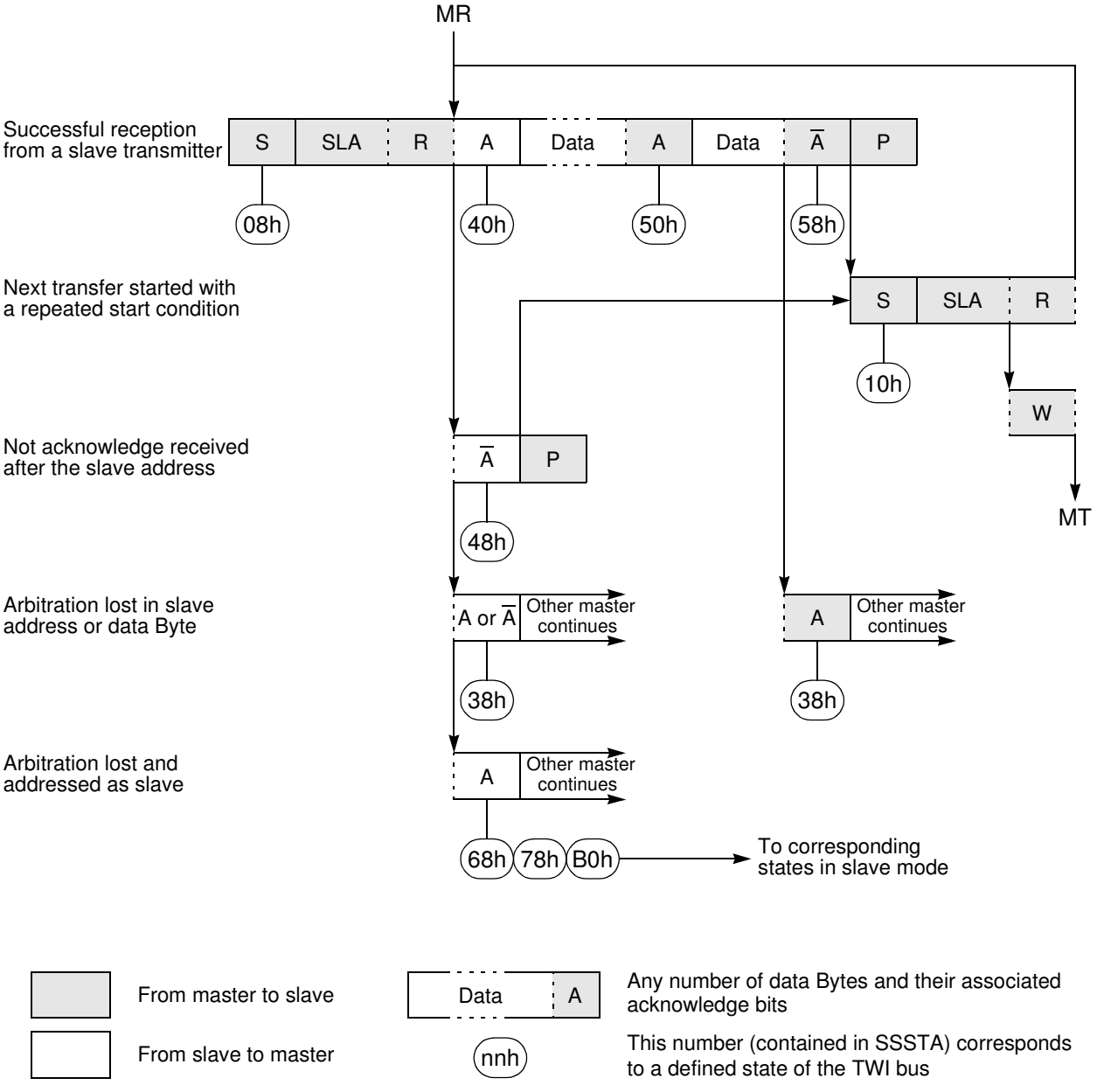


Figure 20-5. Format and States in the Slave Receiver Mode

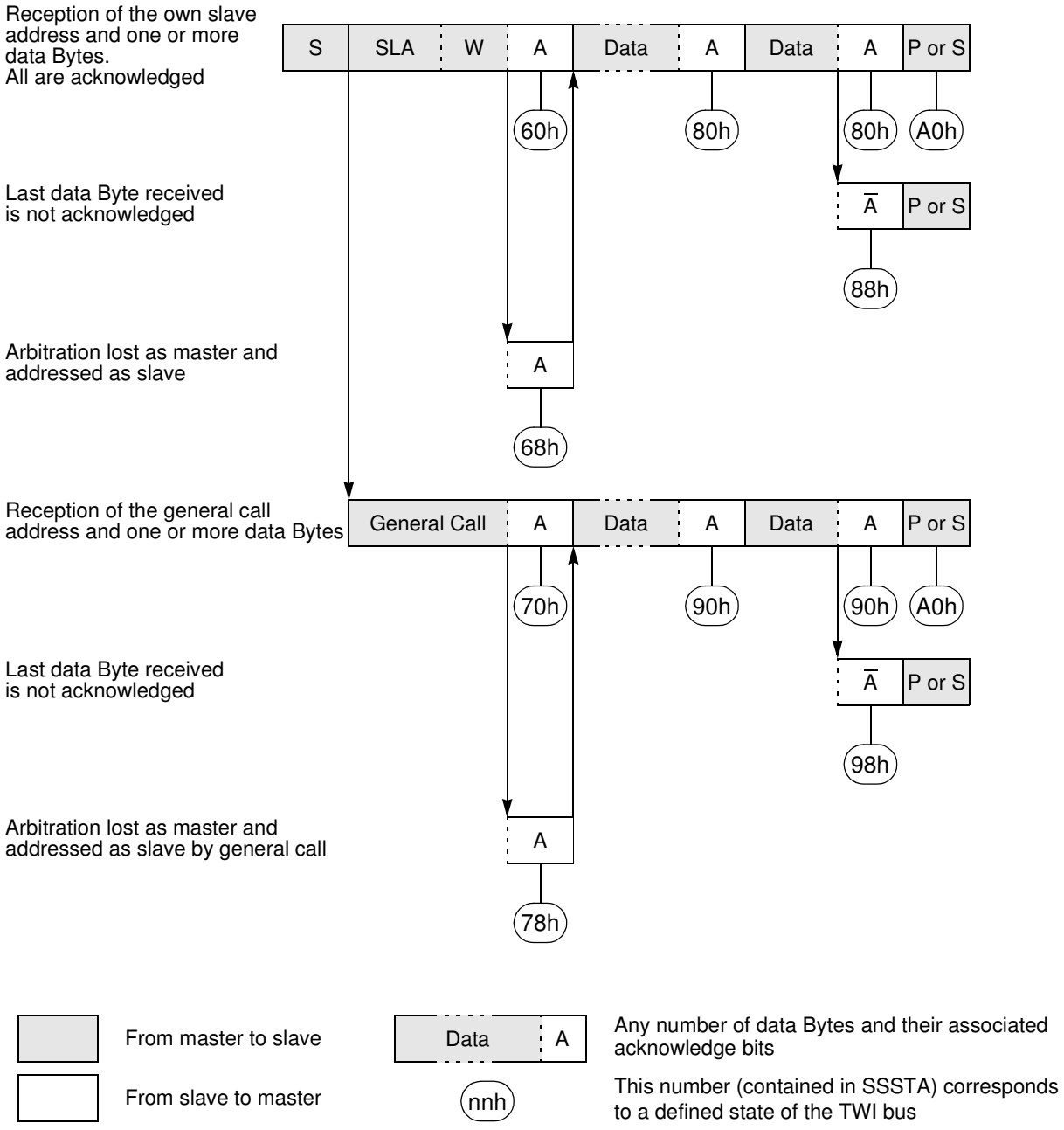


Figure 20-6. Format and States in the Slave Transmitter Mode

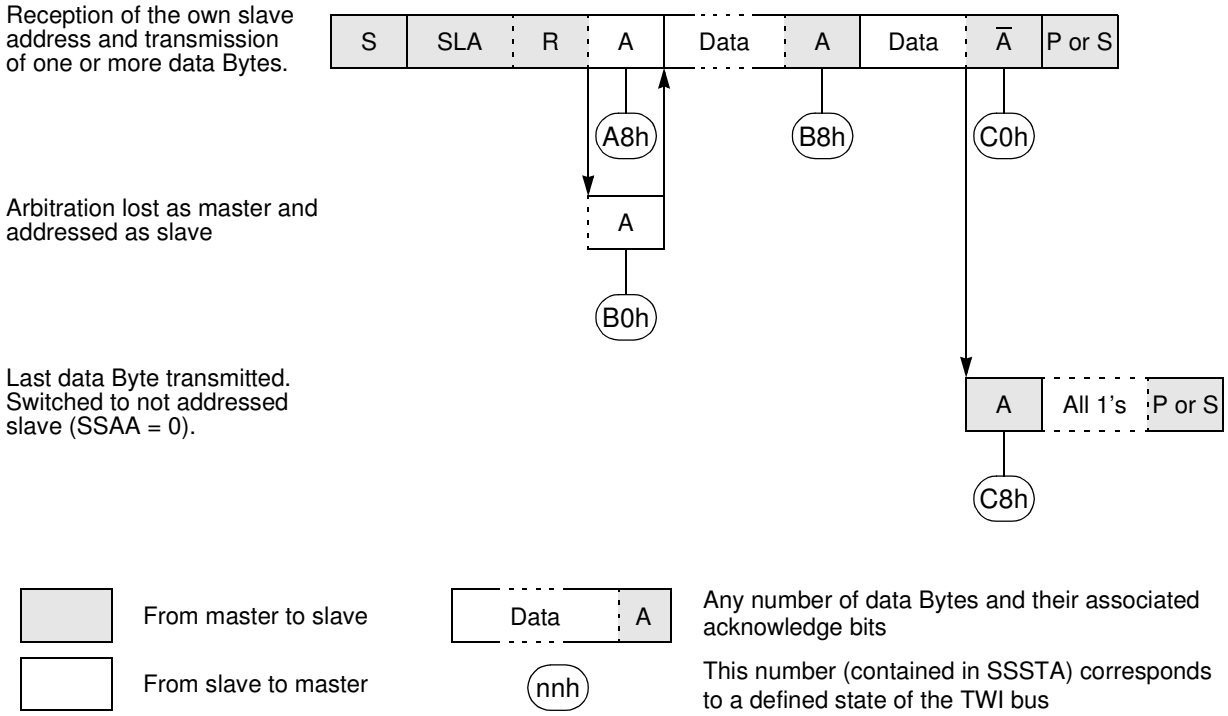


Table 20. Status for Master Transmitter Mode

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

Table 21. Status for Master Receiver Mode

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

Table 22. Status for Slave Receiver Mode with Own Slave Address

Status Code SSSTA	Status of the TWI Bus and TWI Hardware	Application Software Response					Next Action Taken by TWI Hardware
			To SSCON				
		To/From SSDAT	SSSTA	SSSTO	SSI	SSAA	
60h	Own SLA+W has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data Byte will be received and NOT ACK will be returned.
		No SSDAT action	X	0	0	1	Data Byte will be received and ACK will be returned.
68h	Arbitration lost in SLA+R/W as master; own SLA+W has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data Byte will be received and NOT ACK will be returned.
		No SSDAT action	X	0	0	1	Data Byte will be received and ACK will be returned.
80h	Previously addressed with own SLA+W; data has been received; ACK has been returned	Read data Byte	X	0	0	0	Data Byte will be received and NOT ACK will be returned.
		Read data Byte	X	0	0	1	Data Byte will be received and ACK will be returned.
88h	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned	Read data Byte	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA.
		Read data Byte	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1.
		Read data Byte	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free.
		Read data Byte	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1. A START condition will be transmitted when the bus becomes free.
A0h	A STOP condition or repeated START condition has been received while still addressed as slave	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA.
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1.
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free.
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1. A START condition will be transmitted when the bus becomes free.

Table 23. Status for Slave Receiver Mode with General Call Address

Status Code SSSTA	Status of the TWI Bus and TWI Hardware	Application Software Response					Next Action Taken by TWI Hardware
			To SSCON				
		To/From SSDAT	SSSTA	SSSTO	SSI	SSAA	
70h	General call address has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data Byte will be received and NOT ACK will be returned.
		No SSDAT action	X	0	0	1	Data Byte will be received and ACK will be returned.
78h	Arbitration lost in SLA+R/W as master; general call address has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data Byte will be received and NOT ACK will be returned.
		No SSDAT action	X	0	0	1	Data Byte will be received and ACK will be returned.
90h	Previously addressed with general call; data has been received; ACK has been returned	Read data Byte	X	0	0	0	Data Byte will be received and NOT ACK will be returned.
		Read data Byte	X	0	0	1	Data Byte will be received and ACK will be returned.
98h	Previously addressed with general call; data has been received; NOT ACK has been returned	Read data Byte	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA.
		Read data Byte	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1.
		Read data Byte	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free.
		Read data Byte	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1. A START condition will be transmitted when the bus becomes free.
A0h	A STOP condition or repeated START condition has been received while still addressed as slave	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA.
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1.
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free.
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1. A START condition will be transmitted when the bus becomes free.

Table 24. Status for Slave Transmitter Mode

Status Code SSSTA	Status of the TWI Bus and TWI Hardware	Application Software Response					Next Action Taken by TWI Hardware
			To SSCON				
		To/From SSDAT	SSSTA	SSSTO	SSI	SSAA	
A8h	Own SLA+R has been received; ACK has been returned	Write data Byte	X	0	0	0	Last data Byte will be transmitted.
		Write data Byte	X	0	0	1	Data Byte will be transmitted.
B0h	Arbitration lost in SLA+R/W as master; own SLA+R has been received; ACK has been returned	Write data Byte	X	0	0	0	Last data Byte will be transmitted.
		Write data Byte	X	0	0	1	Data Byte will be transmitted.
B8h	Data Byte in SSDAT has been transmitted; ACK has been received	Write data Byte	X	0	0	0	Last data Byte will be transmitted.
		Write data Byte	X	0	0	1	Data Byte will be transmitted.
C0h	Data Byte in SSDAT has been transmitted; NOT ACK has been received	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA.
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1.
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free.
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1. A START condition will be transmitted when the bus becomes free.
C8h	Last data Byte in SSDAT has been transmitted (SSAA= 0); ACK has been received	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA.
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1.
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free.
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC = logic 1. A START condition will be transmitted when the bus becomes free.

Table 25. Status for Miscellaneous States

Status Code SSSTA	Status of the TWI Bus and TWI Hardware	Application Software Response					Next Action Taken by TWI Hardware
			To SSCON				
		To/From SSDAT	SSSTA	SSSTO	SSI	SSAA	
F8h	No relevant state information available; SSI = 0	No SSDAT action	No SSCON action				Wait or proceed current transfer.
00h	Bus error due to an illegal START or STOP condition	No SSDAT action	0	1	0	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and SSSTO is reset.

20.2 Registers

Table 26. SSCON Register

SSCON (S:93h) – Synchronous Serial Control Register

7	6	5	4	3	2	1	0
SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0

Bit Number	Bit Mnemonic	Description
7	SSCR2	**Synchronous Serial Control Rate Bit 2** Refer to Table 19 for rate description.
6	SSPE	**Synchronous Serial Peripheral Enable Bit** Set to enable the controller. Clear to disable the controller.
5	SSSTA	**Synchronous Serial Start Flag** Set to send a START condition on the bus. Clear not to send a START condition on the bus.
4	SSSTO	**Synchronous Serial Stop Flag** Set to send a STOP condition on the bus. Clear not to send a STOP condition on the bus.
3	SSI	**Synchronous Serial Interrupt Flag** Set by hardware when a serial interrupt is requested. Must be cleared by software to acknowledge interrupt.
2	SSAA	**Synchronous Serial Assert Acknowledge Flag** Set to enable slave modes. Slave modes are entered when SLA or GCA (if SSGC set) is recognized. Clear to disable slave modes. Master Receiver Mode in progress Clear to force a not acknowledge (high level on SDA). Set to force an acknowledge (low level on SDA). Master Transmitter Mode in progress This bit has no specific effect when in master transmitter mode. Slave Receiver Mode in progress Clear to force a not acknowledge (high level on SDA). Set to force an acknowledge (low level on SDA). Slave Transmitter Mode in progress Clear to isolate slave from the bus after last data Byte transmission. Set to enable slave mode.
1	SSCR1	**Synchronous Serial Control Rate Bit 1** Refer to Table 19 for rate description.
0	SSCR0	**Synchronous Serial Control Rate Bit 0** Refer to Table 19 for rate description.

Reset Value = 0000 0000b

Table 27. SSSTA Register

SSSTA (S:94h) – Synchronous Serial Status Register

7	6	5	4	3	2	1	0
SSC4	**SSC3**	**SSC2**	**SSC1**	**SSC0**	**0**	**0**	**0**

Bit Number	Bit Mnemonic	Description
7:3	SSC4:0	**Synchronous Serial Status Code Bits 0 to 4** Refer to Table 20 to Table 20-6 for status description.
2:0	0	Always 0.

Reset Value = F8h

Table 28. SSDAT Register

SSDAT (S:95h) – Synchronous Serial Data Register

7	6	5	4	3	2	1	0
SSD7	**SSD6**	**SSD5**	SSD4	SSD3	**SSD2**	**SSD1**	**SSD0**

Bit Number	Bit Mnemonic	Description
7:1	SSD7:1	**Synchronous Serial Address bits 7 to 1 or Synchronous Serial Data Bits 7 to 1**
0	SSD0	**Synchronous Serial Address bit 0 (R/W) or Synchronous Serial Data Bit 0**

Reset Value = 1111 1111b

Table 29. SSADR Register

SSADR (S:96h) – Synchronous Serial Address Register

7	6	5	4	3	2	1	0
SSA7	**SSA6**	**SSA5**	**SSA4**	**SSA3**	**SSA2**	**SSA1**	**SSGC**

Bit Number	Bit Mnemonic	Description
7:1	SSA7:1	**Synchronous Serial Slave Address Bits 7 to 1**
0	SSGC	**Synchronous Serial General Call Bit** Set to enable the general call address recognition. Clear to disable the general call address recognition.

Reset Value = 1111 1110b

21. Analog to Digital Converter

The AT89C5132 implement a 2-channel 10-bit (8 true bits) analog to digital converter (ADC). First channel of this ADC can be used for battery monitoring while the second one can be used for voice sampling at 8 kHz.

21.1 Description

The A/D converter interfaces with the C51 core through four special function registers: ADCON, the ADC control register (see Table 31); ADDH and ADDL, the ADC data registers (see Table 33 and Table 34); and ADCLK, the ADC clock register (see Table 32).

As shown in Figure 21-1, the ADC is composed of a 10-bit cascaded potentiometric digital to analog converter, connected to the negative input of a comparator. The output voltage of this DAC is compared to the analog voltage stored in the Sample and Hold and coming from AIN0 or AIN1 input depending on the channel selected (see Table 30). The 10-bit ADDAT converted value (see formula in Figure 21-1) is delivered in ADDH and ADDL registers, ADDH is giving the 8 most significant bits while ADDL is giving the 2 least significant bits. ADDAT

Figure 21-1. ADC Structure

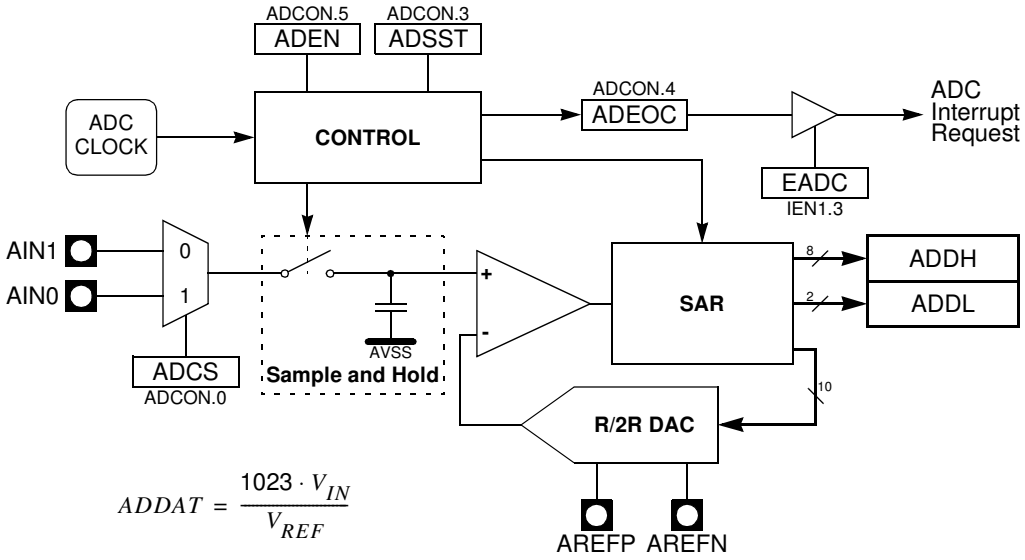
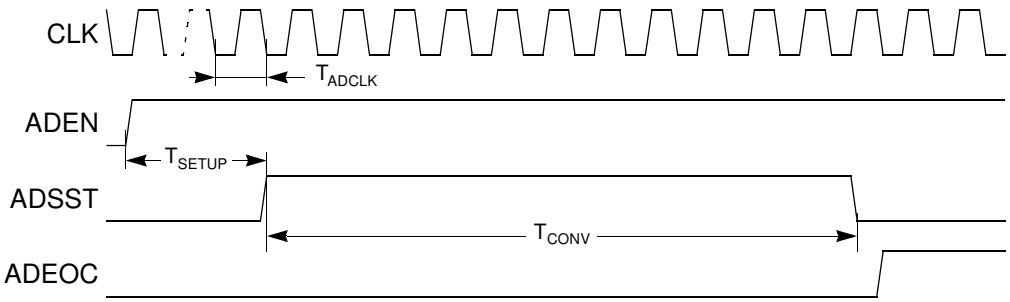


Figure 21-2 shows the timing diagram of a complete conversion. For simplicity, the figure depicts the waveforms in idealized form and do not provide precise timing information. For ADC characteristics and timing parameters refer to the section “AC Characteristics”.

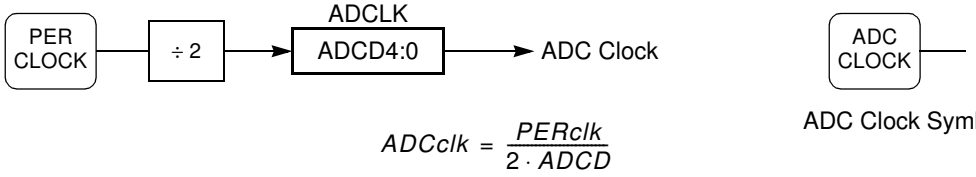
Figure 21-2. Timing Diagram



21.1.1 Clock Generator

The ADC clock is generated by division of the peripheral clock (see details in Section “X2 Feature”, page 12). The division factor is then given by ADCP4:0 bits in ADCLK register. Figure 21-3 shows the ADC clock generator and its calculation formula(1).

Figure 21-3. ADC Clock Generator and Symbol Caution:



$$ADCclk = \frac{PERclk}{2 \cdot ADCD}$$

Note: 1. In all cases, the ADC clock frequency may be higher than the maximum F_{ADCLK} parameter reported in the Section “Analog to Digital Converter”, page 201.

2. The ADCD value of 0 is equivalent to an ADCD value of 32.

21.1.2 Channel Selection

The channel on which conversion is performed is selected by the ADCS bit in ADCON register according to Table 30.

Table 30. ADC Channel Selection

ADCS	Channel
0	AIN1
1	AIN0

21.1.3 Conversion Precision

The 10-bit precision conversion is achieved by stopping the CPU core activity during conversion for limiting the digital noise induced by the core. This mode called the Pseudo-Idle mode(1),(2) is enabled by setting the ADIDL bit in ADCON register(3). Thus, when conversion is launched (see Section "Conversion Launching", page 149), the CPU core is stopped until the end of the con-

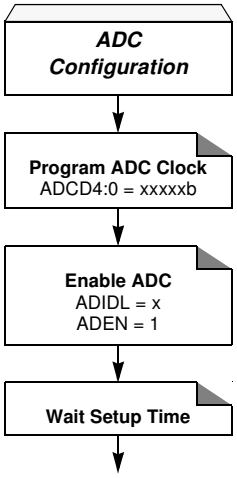
version (see Section "End Of Conversion", page 149). This bit is cleared by hardware at the end of the conversion.

Notes: 1. Only the CPU activity is frozen, peripherals are not affected by the Pseudo-Idle mode.
2. If some interrupts occur during the Pseudo-Idle mode, they will be delayed and processed, according to their priority after the end of the conversion.
3. Concurrently with ADSST bit.

21.1.4 Configuration

The ADC configuration consists in programming the ADC clock as detailed in the Section "Clock Generator", page 148. The ADC is enabled using the ADEN bit in ADCON register. As shown in Figure 93, user must wait the setup time (T_{SETUP}) before launching any conversion.

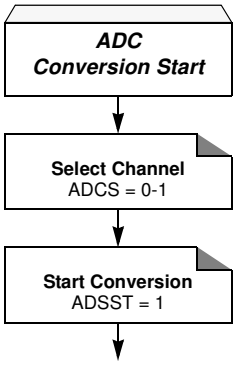
Figure 21-4. ADC Configuration Flow



21.1.5 Conversion Launching

The conversion is launched by setting the ADSST bit in ADCON register, this bit remains set during the conversion. As soon as the conversion is started, it takes 11 clock periods (T_{CONV}) before the data is available in ADDH and ADDL registers.

Figure 21-5. ADC Conversion Launching Flow



21.1.6 End Of Conversion

The end of conversion is signalled by the ADEOC flag in ADCON register becoming set or by the ADSST bit in ADCON register becoming cleared. ADEOC flag can generate an interrupt if

enabled by setting EADC bit in IEN1 register. This flag is set by hardware and must be reset by software.

21.2 Registers

Table 31. ADCON Register

ADCON (S:F3h) – ADC Control Register

7	6	5	4	3	2	1	0
-	ADIDL	ADEN	ADEOC	ADSST	-	-	ADCS

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
6	ADIDL	**ADC Pseudo-Idle Mode** Set to suspend the CPU core activity (pseudo-idle mode) during conversion. Clear by hardware at the end of conversion.
5	ADEN	**ADC Enable Bit** Set to enable the A to D converter. Clear to disable the A to D converter and put it in low power stand by mode.
4	ADEOC	**End Of Conversion Flag** Set by hardware when ADC result is ready to be read. This flag can generate an interrupt. Must be cleared by software.
3	ADSST	**Start and Status Bit** Set to start an A to D conversion on the selected channel. Cleared by hardware at the end of conversion.
2 - 1	-	**Reserved** The value read from these bits is always 0. Do not set these bits.
0	ADCS	**Channel Selection Bit** Set to select channel 0 for conversion. Clear to select channel 1 for conversion.

Reset Value = 0000 0000b

Table 32. ADCLK Register

ADCLK (S:F2h) – ADC Clock Divider Register

7	6	5	4	3	2	1	0
-	-	-	ADCD4	ADCD3	ADCD2	ADCD1	ADCD0

Bit Number	Bit Mnemonic	Description
7 - 5	-	**Reserved** The value read from these bits is always 0. Do not set these bits.
4 - 0	ADCD4:0	**ADC Clock Divider** 5-bit divider for ADC clock generation.

Reset Value = 0000 0000b

Table 33. ADDH Register

ADDH (S:F5h Read Only) – ADC Data High Byte Register

7	6	5	4	3	2	1	0
ADAT9	**ADAT8**	**ADAT7**	**ADAT6**	**ADAT5**	**ADAT4**	**ADAT3**	**ADAT2**

Bit Number	Bit Mnemonic	Description
7 - 0	ADAT9:2	**ADC Data** 8 Most Significant Bits of the 10-bit ADC data.

Reset Value = 0000 0000b

Table 34. ADDL Register

ADDL (S:F4h Read Only) – ADC Data Low Byte Register

7	6	5	4	3	2	1	0
-	**-**	**-**	**-**	**-**	**-**	**ADAT1**	**ADAT0**

Bit Number	Bit Mnemonic	Description
7 - 2	-	**Reserved** The value read from these bits is always 0. Do not set these bits.
1 - 0	ADAT1:0	**ADC Data** 2 Least Significant Bits of the 10-bit ADC data.

Reset Value = 0000 0000b

22. Keyboard Interface

The AT89C5132 implements a keyboard interface allowing the connection of a 4 x n matrix keyboard. It is based on 4 inputs with programmable interrupt capability on both high or low level. These inputs are available as alternate function of P1.3:0 and allow exit from idle and power down modes.

22.1 Description

The keyboard interfaces with the C51 core through two special function registers: KBCON, the keyboard control register (see Table 101); and KBSTA, the keyboard control and status register (see Table 102).

The keyboard inputs are considered as 4 independent interrupt sources sharing the same interrupt vector. An interrupt enable bit (EKB in IEN1 register) allows global enable or disable of the keyboard interrupt (see Figure 22-1). As detailed in Figure 22-2 each keyboard input has the capability to detect a programmable level according to KINL3:0 bit value in KBCON register. Level detection is then reported in interrupt flags KINF3:0 in KBSTA register.

Any of the KINF3:0 flags can trigger a keyboard interrupt. To do so, corresponding mask bits KINM3:0 in KBCON register must be cleared. The keyboard interrupt service routine is executed each time an unmasked KINFx flag is set. The interrupt must be acknowledged by reading KBSTA which automatically clears KINF3:0 flags.
This structure allows keyboard arrangement from 1 by n to 4 by n matrix and allow usage of KIN inputs for any other purposes.

Figure 22-1. Keyboard Interface Block Diagram

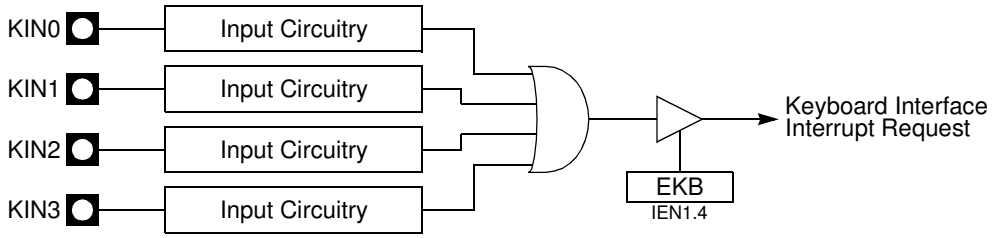
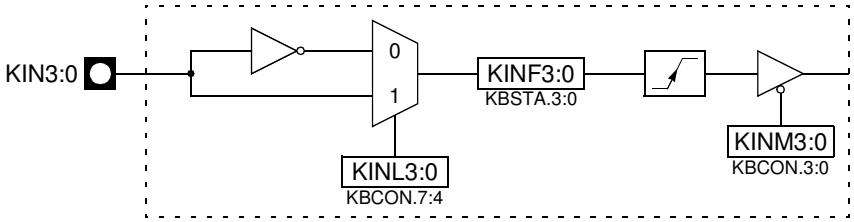


Figure 22-2. Keyboard Input Circuitry



22.1.1 Power Reduction Mode

KIN3:0 inputs allow exit from idle and power down modes as detailed in Section “Power Management”, page 44. To enable this feature, KPDE bit in KBSTA register must be set to logic 1.

Due to the asynchronous keypad detection in power down mode (all clocks are stopped), exit may happen on parasitic key press. In this case, no key is detected and software must enter power-down again.

22.2 Registers

Table 101. KBCON Register

KBCON (S:A3h) – Keyboard Control Register

7	6	5	4	3	2	1	0
KINL3	KINL2	KINL1	KINL0	KINM3	KINM2	KINM1	KINM0

Bit Number	Bit Mnemonic	Description
7 - 4	KINL3:0	**Keyboard Input Level Bit** Set to enable a high level detection on the respective KIN3:0 input. Clear to enable a low level detection on the respective KIN3:0 input.
3 - 0	KINM3:0	**Keyboard Input Mask Bit** Set to prevent the respective KINF3:0 flag from generating a keyboard interrupt. Clear to allow the respective KINF3:0 flag to generate a keyboard interrupt.

Reset Value = 0000 1111b

22.2.0.1

Table 102. KBSTA Register

KBSTA (S:A4h) – Keyboard Control and Status Register

7	6	5	4	3	2	1	0
KPDE	-	-	-	KINF3	KINF2	KINF1	KINF0

Bit Number	Bit Mnemonic	Description
7	KPDE	**Keyboard Power Down Enable Bit** Set to enable exit of power down mode by the keyboard interrupt. Clear to disable exit of power down mode by the keyboard interrupt.
6 - 4	-	**Reserved** The values read from these Bits are always 0. Do not set these Bits.
3 - 0	KINF3:0	**Keyboard Input Interrupt Flag** Set by hardware when the respective KIN3:0 input detects a programmed level. Cleared when reading KBSTA.

Reset Value = 0000 0000b

23. Electrical Characteristics

23.1 Absolute Maximum Ratings

Storage Temperature	-65°C to +150°C
Voltage on any other Pin to V_{SS}	-0.3 to +4.0V
I_{OL} per I/O Pin	5 mA
Power Dissipation	1 W
Ambient Temperature Under Bias	-40°C to +85°C
V_{DD}	2.7V to 3.3V

*NOTICE: Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only. Operation beyond the “operating conditions” is not recommended and extended exposure beyond the “Operating Conditions” may affect device reliability.

23.2 DC Characteristics

23.2.1 Digital Logic

Table 103. Digital DC Characteristics
V_{DD} = 2.7 to 3.3V , T_A = -40 to +85°C

Symbol	Parameter	Min	Typ(1)	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5		$0.2 \cdot V_{DD}$ - 0.1	V	
V_{IH1}	Input High Voltage (except RST, X1)	$0.2 \cdot V_{DD}$ + 1.1		V_{DD}	V	
V_{IH2}	Input High Voltage (RST, X1)	$0.7 \cdot V_{DD}$(2)		V_{DD} + 0.5	V	
V_{OL1}	Output Low Voltage (except P0, ALE, MCMD, MDAT, MCLK, SCLK, DCLK, DSEL, DOUT)			0.45	V	I_{OL}= 1.6 mA
V_{OL2}	Output Low Voltage (P0, ALE, MCMD, MDAT, MCLK, SCLK, DCLK, DSEL, DOUT)			0.45	V	I_{OL}= 3.2 mA
V_{OH1}	Output High Voltage (P1, P2, P3, P4 and P5)	V_{DD} - 0.7			V	I_{OH}= -30 µA
V_{OH2}	Output High Voltage (P0, P2 address mode, ALE, MCMD, MDAT, MCLK, SCLK, DCLK, DSEL, DOUT, D+, D-)	V_{DD} - 0.7			V	I_{OH}= -3.2 mA
I_{IL}	Logical 0 Input Current (P1, P2, P3, P4 and P5)			-50	µA	Vin = 0.45 V

Table 103. Digital DC Characteristics
V_{DD} = 2.7 to 3.3V , T_A = -40 to +85°C

Symbol	Parameter	Min	Typ(1)	Max	Units	Test Conditions
I_{LI}	Input Leakage Current (P0, ALE, MCMD, MDAT, MCLK, SCLK, DCLK, DSEL, DOUT)			10	µA	0.45< V_{IN}< V_{DD}
I_{TL}	Logical 1 to 0 Transition Current (P1, P2, P3, P4 and P5)			-650	µA	Vin = 2.0 V
R_{RST}	Pull-Down Resistor	50	90	200	kΩ	
C_{IO}	Pin Capacitance		10		pF	T_A= 25°C
V_{RET}	V_{DD} Data Retention Limit			1.8	V	
I_{DD}	Operating Current		(3)	X1 / X2 mode 6.5 / 10.5 8 / 13.5 9.5 / 17	mA	V_{DD} < 3.3 V 12 MHz 16 MHz 20 MHz
I_{DL}	Idle Mode Current		(3)	X1 / X2 mode 5.3 / 8.1 6.4 / 10.3 7.5 / 13	mA	V_{DD} < 3.3 V 12 MHz 16 MHz 20 MHz
I_{PD}	Power-Down Mode Current		20	500	µA	V_{RET} < V_{DD} < 3.3 V

Notes: 1. Typical values are obtained using V_{DD}= 3 V and T_A= 25°C. They are not tested and there is no guarantee on these values.
2. Flash retention is guaranteed with the same formula for V_{DD} min down to 0V.
3. See Table 154 for typical consumption in player mode.

23.2.2 I_{DD}, I_{DL} and I_{PD} Test Conditions

Figure 23-1. I_{DD} Test Condition, Active Mode

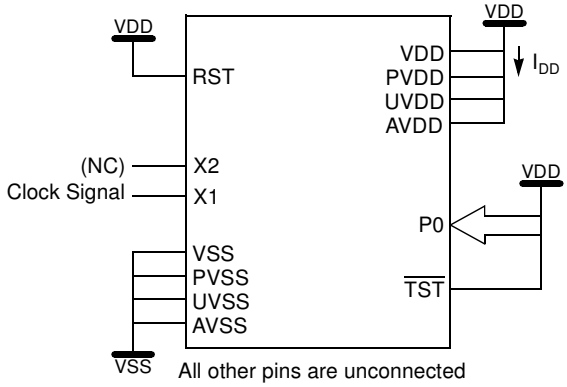


Figure 23-2. I_{DL} Test Condition, Idle Mode

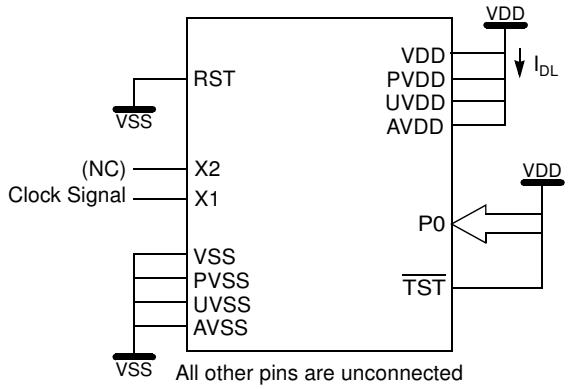
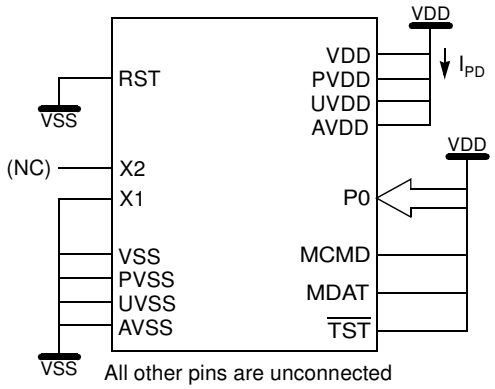


Figure 23-3. I_{PD} Test Condition, Power-Down Mode



23.2.3 A-to-D Converter

Table 104. A-to-D Converter DC Characteristics

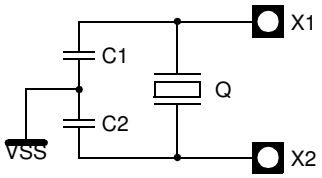
V_{DD} = 2.7 to 3.3V , T_A = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
AV_{DD}	Analog Supply Voltage	2.7		3.3	V	
AI_{DD}	Analog Operating Supply Current			600	µA	AV_{DD} = 3.3V AIN1:0 = 0 to AV_{DD}
AI_{PD}	Analog Standby Current			2	µA	AV_{DD} = 3.3V ADEN = 0 or PD = 1
AV_{IN}	Analog Input Voltage	AV_{SS}		AV_{DD}	V	
AV_{REF}	Reference Voltage A_{REFN} A_{REFP}	 AV_{SS} 2.4		 AV_{DD}	 V V	
R_{REF}	AREF Input Resistance	10		30	kΩ	T_A = 25°C
C_{IA}	Analog Input capacitance			10	pF	T_A = 25°C

23.2.4 Oscillator and Crystal

23.2.4.1 Schematic

Figure 23-4. Crystal Connection



Note: For operation with most standard crystals, no external components are needed on X1 and X2. It may be necessary to add external capacitors on X1 and X2 to ground in special cases (max 10 pF). X1 and X2 may not be used to drive other circuits.

23.2.4.2 Parameters

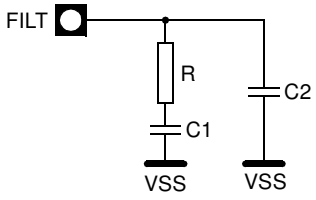
Table 105. Oscillator and Crystal Characteristics
V_{DD} = 2.7 to 3.3V , T_A = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
C_{X1}	Internal Capacitance (X1 - V_{SS})		10		pF
C_{X2}	Internal Capacitance (X2 - V_{SS})		10		pF
C_L	Equivalent Load Capacitance (X1 - X2)		5		pF
DL	Drive Level			50	µW
F	Crystal Frequency			20	MHz
RS	Crystal Series Resistance			40	Ω
CS	Crystal Shunt Capacitance			6	pF

23.2.5 Phase Lock Loop

23.2.5.1 Schematic

Figure 23-5. PLL Filter Connection



23.2.5.2 Parameters

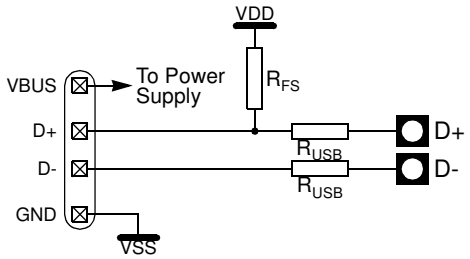
Table 106. PLL Filter Characteristics
V_{DD} = 2.7 to 3.3V , T_A = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
R	Filter Resistor		100		Ω
C1	Filter Capacitance 1		10		nF
C2	Filter Capacitance 2		2.2		nF

23.2.6 USB Connection

23.2.6.1 Schematic

Figure 23-6. USB Connection



23.2.6.2 Parameters

Table 35. USB Characteristics

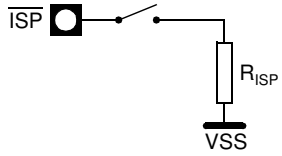
V_{DD} = 3 to 3.3 V, T_A = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
R_{USB}	USB Termination Resistor		27		Ω
R_{FS}	USB Full Speed Resistor		1.5		KΩ

23.2.7 In-system Programming

23.2.7.1 Schematic

Figure 23-7. ISP Pull-down Connection



23.2.7.2 Parameters

Table 107. ISP Pull-Down Characteristics
V_{DD} = 3 to 3.3V , T_A = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
R_{ISP}	ISP Pull-Down Resistor		2.2		kΩ

23.3 AC Characteristics

23.3.1 External 8-bit Bus Cycles

23.3.1.1 Definition of Symbols

Table 108. External 8-bit Bus Cycles Timing Symbol Definitions

Signals	
A	Address
D	Data In
L	ALE
Q	Data Out
R	$\overline{RD}$
W	$\overline{WR}$

Conditions	
H	High
L	Low
V	Valid
X	No Longer Valid
Z	Floating

23.3.1.2 Timings

Test conditions: capacitive load on all pins = 50 pF.

Table 109. External 8-bit Bus Cycle – Data Read AC Timings
V_{DD} = 2.7 to 3.3V, T_A = -40° to +85°C

		Variable Clock Standard Mode		Variable Clock X2 Mode		
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Unit**
T_{CLCL}	Clock Period	50		50		ns
T_{LHLL}	ALE Pulse Width	$2 \cdot T_{CLCL}-15$		$T_{CLCL}-15$		ns
T_{AVLL}	Address Valid to ALE Low	$T_{CLCL}-20$		$0.5 \cdot T_{CLCL}-20$		ns
T_{LLAX}	Address hold after ALE Low	$T_{CLCL}-20$		$0.5 \cdot T_{CLCL}-20$		ns
T_{LLRL}	ALE Low to $\overline{RD}$ Low	$3 \cdot T_{CLCL}-30$		$1.5 \cdot T_{CLCL}-30$		ns
T_{RLRH}	$\overline{RD}$ Pulse Width	$6 \cdot T_{CLCL}-25$		$3 \cdot T_{CLCL}-25$		ns
T_{RHLH}	$\overline{RD}$ high to ALE High	$T_{CLCL}-20$	$T_{CLCL}+20$	$0.5 \cdot T_{CLCL}-20$	$0.5 \cdot T_{CLCL}+20$	ns
T_{AVDV}	Address Valid to Valid Data In		$9 \cdot T_{CLCL}-65$		$4.5 \cdot T_{CLCL}-65$	ns
T_{AVRL}	Address Valid to $\overline{RD}$ Low	$4 \cdot T_{CLCL}-30$		$2 \cdot T_{CLCL}-30$		ns
T_{RLDV}	$\overline{RD}$ Low to Valid Data		$5 \cdot T_{CLCL}-30$		$2.5 \cdot T_{CLCL}-30$	ns
T_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
T_{RHDX}	Data Hold After $\overline{RD}$ High	0		0		ns
T_{RHDZ}	Instruction Float After $\overline{RD}$ High		$2 \cdot T_{CLCL}-25$		$T_{CLCL}-25$	ns

Table 110. External 8-bit Bus Cycle – Data Write AC Timings
V_{DD} = 2.7 to 3.3V, T_A = -40° to +85°C

Symbol	Parameter	Variable Clock Standard Mode		Variable Clock X2 Mode		Unit
		Min	Max	Min	Max	
T_{CLCL}	Clock Period	50		50		ns
T_{LHLL}	ALE Pulse Width	$2 \cdot T_{CLCL}$-15		T_{CLCL}-15		ns
T_{AVLL}	Address Valid to ALE Low	T_{CLCL}-20		$0.5 \cdot T_{CLCL}$-20		ns
T_{LLAX}	Address hold after ALE Low	T_{CLCL}-20		$0.5 \cdot T_{CLCL}$-20		ns
T_{LLWL}	ALE Low to $\overline{WR}$ Low	$3 \cdot T_{CLCL}$-30		$1.5 \cdot T_{CLCL}$-30		ns
T_{WLWH}	$\overline{WR}$ Pulse Width	$6 \cdot T_{CLCL}$-25		$3 \cdot T_{CLCL}$-25		ns
T_{WHLH}	$\overline{WR}$ High to ALE High	T_{CLCL}-20	T_{CLCL}+20	$0.5 \cdot T_{CLCL}$-20	$0.5 \cdot T_{CLCL}$+20	ns
T_{AVWL}	Address Valid to $\overline{WR}$ Low	$4 \cdot T_{CLCL}$-30		$2 \cdot T_{CLCL}$-30		ns
T_{QVWH}	Data Valid to $\overline{WR}$ High	$7 \cdot T_{CLCL}$-20		$3.5 \cdot T_{CLCL}$-20		ns
T_{WHQX}	Data Hold after $\overline{WR}$ High	T_{CLCL}-15		$0.5 \cdot T_{CLCL}$-15		ns

23.3.1.3 Waveforms

Figure 23-8. External 8-bit Bus Cycle – Data Read Waveforms

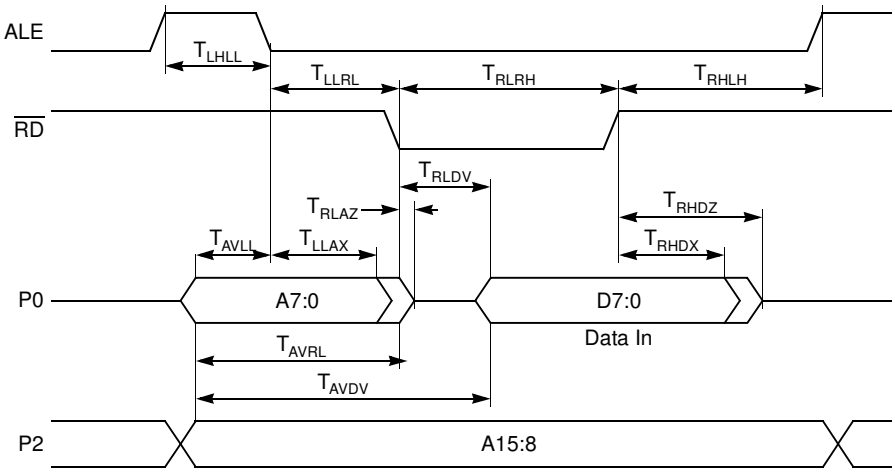
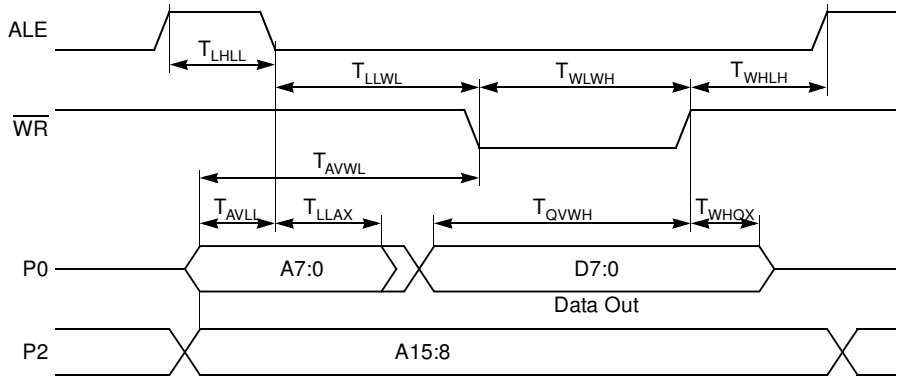


Figure 23-9. External 8-bit Bus Cycle – Data Write Waveforms



23.3.2 External IDE 16-bit Bus Cycles

23.3.2.1 Definition of Symbols

Table 111. External IDE 16-bit Bus Cycles Timing Symbol Definitions

Signals	
A	Address
D	Data In
L	ALE
Q	Data Out
R	$\overline{RD}$
W	$\overline{WR}$

Conditions	
H	High
L	Low
V	Valid
X	No Longer Valid
Z	Floating

23.3.2.2 Timings

Test conditions: capacitive load on all pins = 50 pF.

Table 112. External IDE 16-bit Bus Cycle – Data Read AC Timings
V_{DD} = 2.7 to 3.3V, T_A = -40° to +85°C

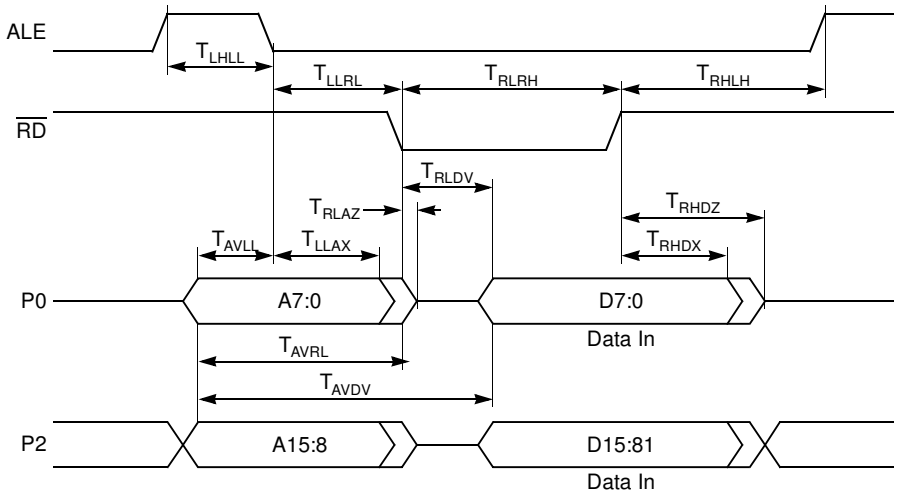
Symbol	Parameter	Variable Clock Standard Mode		Variable Clock X2 Mode		Unit
		Min	Max	Min	Max	
T_{CLCL}	Clock Period	50		50		ns
T_{LHLL}	ALE Pulse Width	$2 \cdot T_{CLCL}$-15		T_{CLCL}-15		ns
T_{AVLL}	Address Valid to ALE Low	T_{CLCL}-20		$0.5 \cdot T_{CLCL}$-20		ns
T_{LLAX}	Address hold after ALE Low	T_{CLCL}-20		$0.5 \cdot T_{CLCL}$-20		ns
T_{LLRL}	ALE Low to $\overline{RD}$ Low	$3 \cdot T_{CLCL}$-30		$1.5 \cdot T_{CLCL}$-30		ns
T_{RLRH}	$\overline{RD}$ Pulse Width	$6 \cdot T_{CLCL}$-25		$3 \cdot T_{CLCL}$-25		ns
T_{RHLH}	$\overline{RD}$ high to ALE High	T_{CLCL}-20	T_{CLCL}+20	$0.5 \cdot T_{CLCL}$-20	$0.5 \cdot T_{CLCL}$+20	ns
T_{AVDV}	Address Valid to Valid Data In		$9 \cdot T_{CLCL}$-65		$4.5 \cdot T_{CLCL}$-65	ns
T_{AVRL}	Address Valid to $\overline{RD}$ Low	$4 \cdot T_{CLCL}$-30		$2 \cdot T_{CLCL}$-30		ns
T_{RLDV}	$\overline{RD}$ Low to Valid Data		$5 \cdot T_{CLCL}$-30		$2.5 \cdot T_{CLCL}$-30	ns
T_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
T_{RHDX}	Data Hold After $\overline{RD}$ High	0		0		ns
T_{RHDZ}	Instruction Float After $\overline{RD}$ High		$2 \cdot T_{CLCL}$-25		T_{CLCL}-25	ns

Table 113. External IDE 16-bit Bus Cycle – Data Write AC Timings
V_{DD} = 2.7 to 3.3V, T_A = -40° to +85°C

Symbol	Parameter	Variable Clock Standard Mode		Variable Clock X2 Mode		Unit
		Min	Max	Min	Max	
T_{CLCL}	Clock Period	50		50		ns
T_{LHLL}	ALE Pulse Width	$2 \cdot T_{CLCL}$-15		T_{CLCL}-15		ns
T_{AVLL}	Address Valid to ALE Low	T_{CLCL}-20		$0.5 \cdot T_{CLCL}$-20		ns
T_{LLAX}	Address hold after ALE Low	T_{CLCL}-20		$0.5 \cdot T_{CLCL}$-20		ns
T_{LLWL}	ALE Low to $\overline{WR}$ Low	$3 \cdot T_{CLCL}$-30		$1.5 \cdot T_{CLCL}$-30		ns
T_{WLWH}	$\overline{WR}$ Pulse Width	$6 \cdot T_{CLCL}$-25		$3 \cdot T_{CLCL}$-25		ns
T_{WHLH}	$\overline{WR}$ High to ALE High	T_{CLCL}-20	T_{CLCL}+20	$0.5 \cdot T_{CLCL}$-20	$0.5 \cdot T_{CLCL}$+20	ns
T_{AVWL}	Address Valid to $\overline{WR}$ Low	$4 \cdot T_{CLCL}$-30		$2 \cdot T_{CLCL}$-30		ns
T_{QVWH}	Data Valid to $\overline{WR}$ High	$7 \cdot T_{CLCL}$-20		$3.5 \cdot T_{CLCL}$-20		ns
T_{WHQX}	Data Hold after $\overline{WR}$ High	T_{CLCL}-15		$0.5 \cdot T_{CLCL}$-15		ns

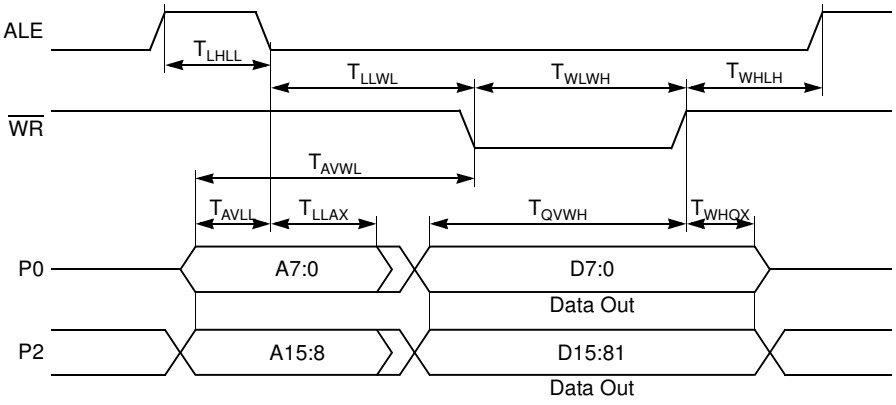
23.3.2.3 Waveforms

Figure 23-10. External IDE 16-bit Bus Cycle – Data Read Waveforms



Note: D15:8 is written in DAT16H SFR.

Figure 23-11. External IDE 16-bit Bus Cycle – Data Write Waveforms



Note: D15:8 is the content of DAT16H SFR.

23.3.3 SPI Interface

23.3.3.1 Definition of Symbols

Table 114. SPI Interface Timing Symbol Definitions

Signals	
C	Clock
I	Data In
O	Data Out

Conditions	
H	High
L	Low
V	Valid
X	No Longer Valid
Z	Floating

23.3.3.2 Timings

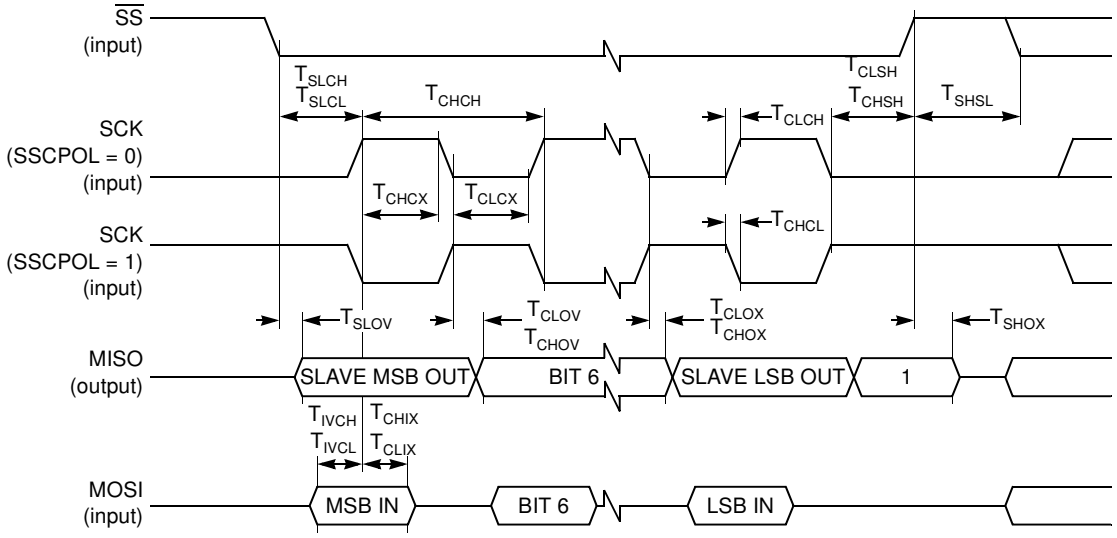
Table 115. SPI Interface Master AC Timing
V_{DD} = 2.7 to 3.3V, T_A = -40° to +85°C

Symbol	Parameter	Min	Max	Unit
Slave Mode				
T_{CHCH}	Clock Period	8		T_{OSC}
T_{CHCX}	Clock High Time	3.2		T_{OSC}
T_{CLCX}	Clock Low Time	3.2		T_{OSC}
T_{SLCH}, T_{SLCL}	$\overline{SS}$ Low to Clock edge	200		ns
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	100		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	100		ns
$T_{CLOV,}$ T_{CHOV}	Output Data Valid after Clock Edge		100	ns
T_{CLOX}, T_{CHOX}	Output Data Hold Time after Clock Edge	0		ns
T_{CLSH}, T_{CHSH}	$\overline{SS}$ High after Clock Edge	0		ns
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	100		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	100		ns
T_{SLOV}	$\overline{SS}$ Low to Output Data Valid		130	ns
T_{SHOX}	Output Data Hold after $\overline{SS}$ High		130	ns
T_{SHSL}	$\overline{SS}$ High to $\overline{SS}$ Low	(1)		
T_{ILIH}	Input Rise Time		2	µs
T_{IHIL}	Input Fall Time		2	µs
T_{OLOH}	Output Rise Time		100	ns
T_{OHOL}	Output Fall Time		100	ns
Master Mode				
T_{CHCH}	Clock Period	4		T_{OSC}
T_{CHCX}	Clock High Time	1.6		T_{OSC}
T_{CLCX}	Clock Low Time	1.6		T_{OSC}
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	50		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	50		ns
$T_{CLOV,}$ T_{CHOV}	Output Data Valid after Clock Edge		65	ns
T_{CLOX}, T_{CHOX}	Output Data Hold Time after Clock Edge	0		ns
T_{ILIH}	Input Data Rise Time		2	µs
T_{IHIL}	Input Data Fall Time		2	µs
T_{OLOH}	Output Data Rise Time		50	ns
T_{OHOL}	Output Data Fall Time		50	ns

Notes: 1. Value of this parameter depends on software.
2. Test conditions: capacitive load on all pins = 100 pF

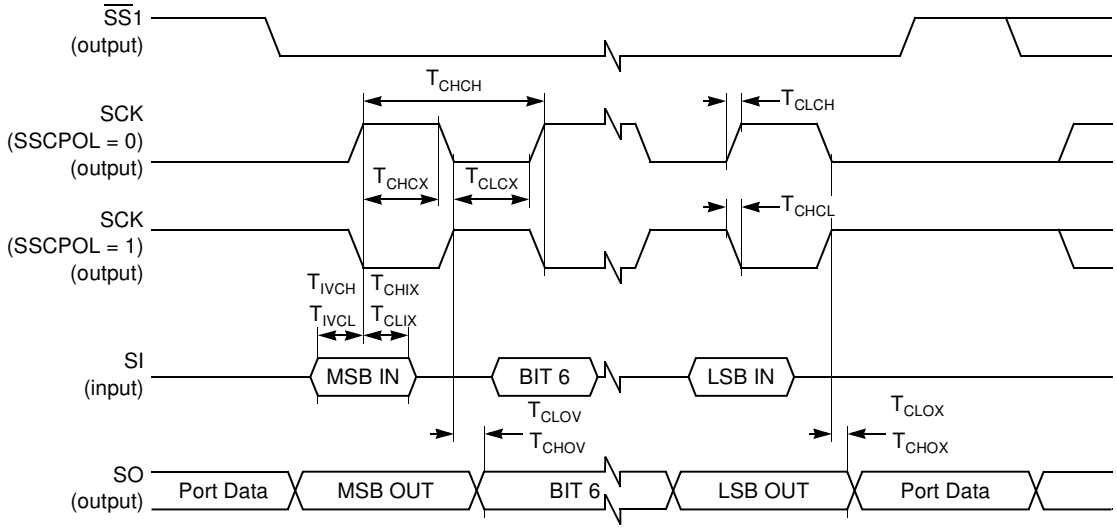
23.3.3.3 Waveforms

Figure 23-12. SPI Slave Waveforms (SSCPHA = 0)



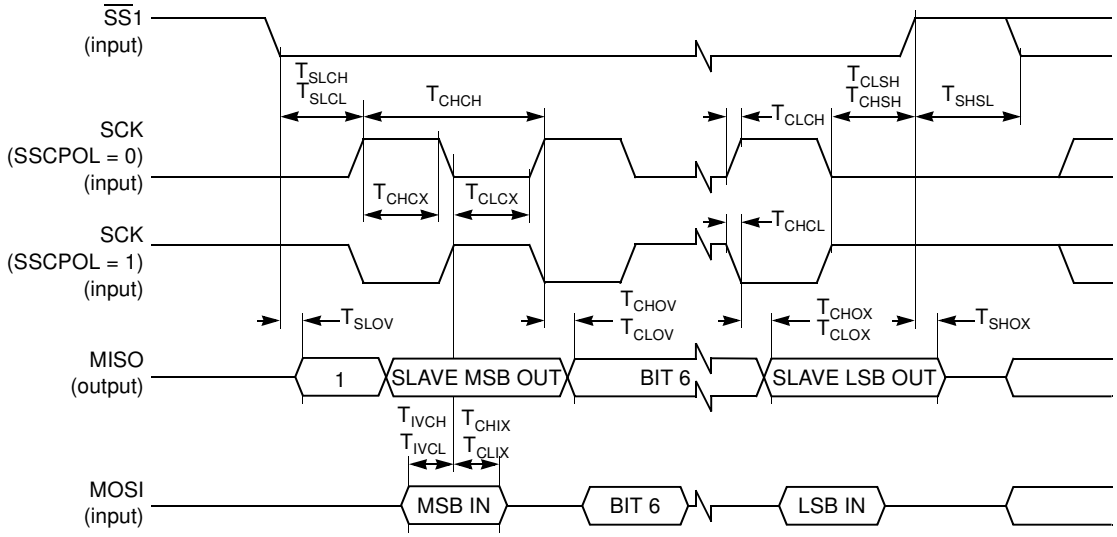
Note: 1. Not Defined but generally the MSB of the character which has just been received.

Figure 23-13. SPI Slave Waveforms (SSCPHA = 1)



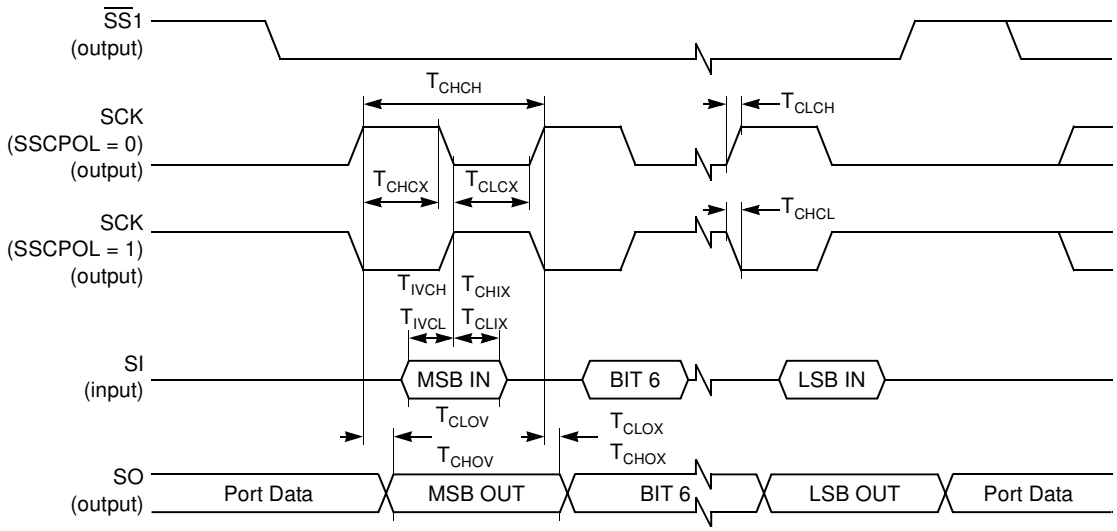
Note: 1. Not Defined but generally the LSB of the character which has just been received.

Figure 23-14. SPI Master Waveforms (SSCPHA = 0)



Note: 1. $\overline{SS}$ handled by software using general purpose port pin.

Figure 23-15. SPI Master Waveforms (SSCPHA = 1)



Note: 1. $\overline{SS}$ handled by software using general purpose port pin.

23.3.4 Two-wire Interface

23.3.4.1 Timings

Table 36. TWI Interface AC Timing

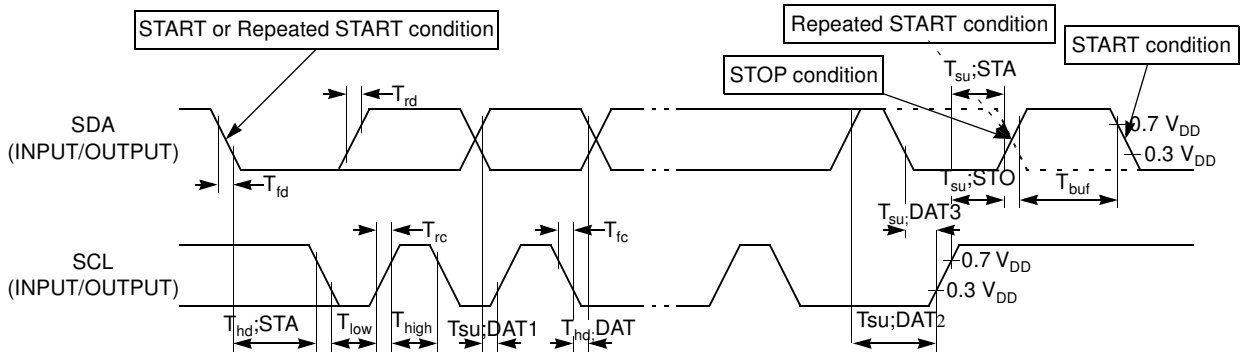
V_{DD} = 2.7 to 3.3 V, T_A = -40 to +85°C

Symbol	Parameter	INPUT Min Max	OUTPUT Min Max
T_{HD}; STA	Start condition hold time	14·T_{CLCL}(4)	4.0 µs(1)
T_{LOW}	SCL low time	16·T_{CLCL}(4)	4.7 µs(1)
T_{HIGH}	SCL high time	14·T_{CLCL}(4)	4.0 µs(1)
T_{RC}	SCL rise time	1 µs	-(2)
T_{FC}	SCL fall time	0.3 µs	0.3 µs(3)
T_{SU}; DAT1	Data set-up time	250 ns	20·T_{CLCL}(4)- T_{RD}
T_{SU}; DAT2	SDA set-up time (before repeated START condition)	250 ns	1 µs(1)
T_{SU}; DAT3	SDA set-up time (before STOP condition)	250 ns	8·T_{CLCL}(4)
T_{HD}; DAT	Data hold time	0 ns	8·T_{CLCL}(4) - T_{FC}
T_{SU}; STA	Repeated START set-up time	14·T_{CLCL}(4)	4.7 µs(1)
T_{SU}; STO	STOP condition set-up time	14·T_{CLCL}(4)	4.0 µs(1)
T_{BUF}	Bus free time	14·T_{CLCL}(4)	4.7 µs(1)
T_{RD}	SDA rise time	1 µs	-(2)
T_{FD}	SDA fall time	0.3 µs	0.3 µs(3)

Notes:
1. At 100 kbit/s. At other bit-rates this value is inversely proportional to the bit-rate of 100 kbit/s.
2. Determined by the external bus-line capacitance and the external bus-line pull-up resistor, this must be < 1 µs.
3. Spikes on the SDA and SCL lines with a duration of less than 3·T_{CLCL} will be filtered out. Maximum capacitance on bus-lines SDA and SCL= 400 pF.
4. T_{CLCL}= T_{OSC}= one oscillator clock period.

23.3.4.2 Waveforms

Figure 23-16. Two Wire Waveforms



23.3.5 MMC Interface

23.3.5.1 Definition of Symbols

Table 116. MMC Interface Timing Symbol Definitions

Signals	
C	Clock
D	Data In
O	Data Out

Conditions	
H	High
L	Low
V	Valid
X	No Longer Valid

23.3.5.2 Timings

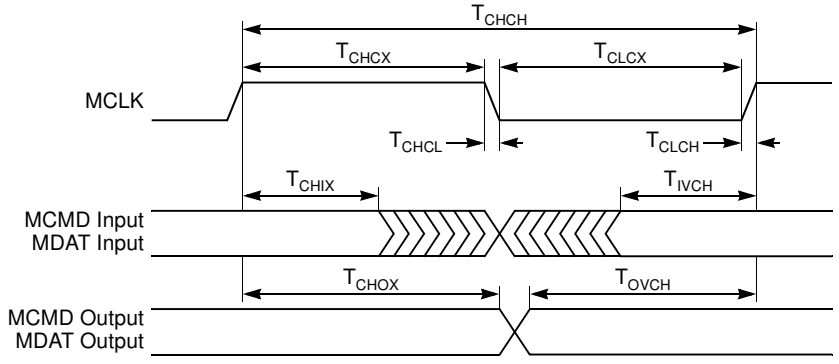
Table 117. MMC Interface AC Timings

V_{DD} = 2.7 to 3.3 V, T_A = -40 to +85°C, CL ≤ 100pF (10 cards)

Symbol	Parameter	Min	Max	Unit
T_{CHCH}	Clock Period	50		ns
T_{CHCX}	Clock High Time	10		ns
T_{CLCX}	Clock Low Time	10		ns
T_{CLCH}	Clock Rise Time		10	ns
T_{CHCL}	Clock Fall Time		10	ns
T_{DVCH}	Input Data Valid to Clock High	3		ns
T_{CHDX}	Input Data Hold after Clock High	3		ns
T_{CHOX}	Output Data Hold after Clock High	5		ns
T_{OVCH}	Output Data Valid to Clock High	5		ns

23.3.5.3 Waveforms

Figure 23-17. MMC Input Output Waveforms



23.3.6 Audio Interface

23.3.6.1 Definition of Symbols

Table 118. Audio Interface Timing Symbol Definitions

Signals	
C	Clock
O	Data Out
S	Data Select

Conditions	
H	High
L	Low
V	Valid
X	No Longer Valid

23.3.6.2 Timings

Table 119. Audio Interface AC timings

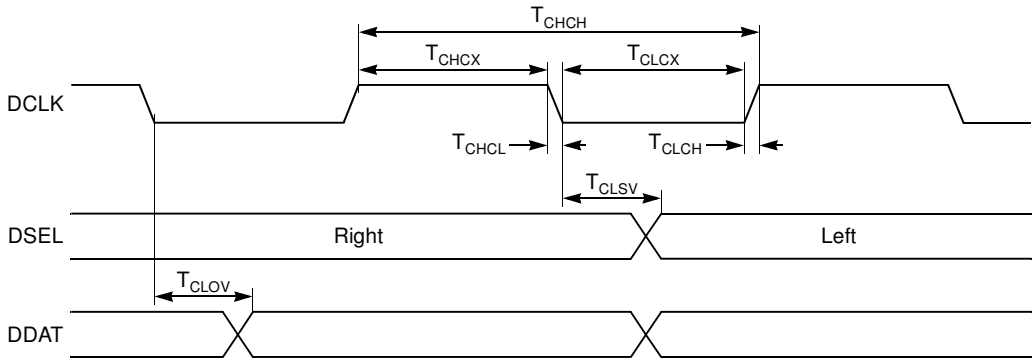
V_{DD} = 2.7 to 3.3V, T_A = -40 to +85°C, CL ≤ 30pF

Symbol	Parameter	Min	Max	Unit
T_{CHCH}	Clock Period		325.5(1)	ns
T_{CHCX}	Clock High Time	30		ns
T_{CLCX}	Clock Low Time	30		ns
T_{CLCH}	Clock Rise Time		10	ns
T_{CHCL}	Clock Fall Time		10	ns
T_{CLSV}	Clock Low to Select Valid		10	ns
T_{CLOV}	Clock Low to Data Valid		10	ns

Note: 32-bit format with Fs = 48 kHz.

23.3.6.3 Waveforms

Figure 23-18. Audio Interface Waveforms



23.3.7 Analog to Digital Converter

23.3.7.1 Definition of Symbols

Table 120. Analog to Digital Converter Timing Symbol Definitions

Signals	
C	Clock
E	Enable (ADEN bit)
S	Start Conversion (ADSST bit)

Conditions	
H	High
L	Low

23.3.7.2 Characteristics

Table 37. Analog to Digital Converter AC Characteristics

V_{DD} = 2.7 to 3.3 V, T_A = -40 to +85°C

Symbol	Parameter	Min	Max	Unit
T_{CLCL}	Clock Period	4		µs
T_{EHSH}	Start-up Time		4	µs
T_{SHSL}	Conversion Time		$11 \cdot T_{CLCL}$	µs
DLe	Differential non-linearity error(1)(2)		1	LSB
ILe	Integral non-linearity errorss(1)(3)		2	LSB
OSe	Offset error(1)(4)		4	LSB
Ge	Gain error(1)(5)		4	LSB

Notes:
1. AV_{DD}= AV_{REFP}= 3.0 V, AV_{SS}= AV_{REFN}= 0 V. ADC is monotonic with no missing code.
2. The differential non-linearity is the difference between the actual step width and the ideal step width (see Figure 23-20).
3. The integral non-linearity is the peak difference between the center of the actual step and the ideal transfer curve after appropriate adjustment of gain and offset errors (see Figure 23-20).
4. The offset error is the absolute difference between the straight line which fits the actual transfer curve (after removing of gain error), and the straight line which fits the ideal transfer curve (see Figure 23-20).
5. The gain error is the relative difference in percent between the straight line which fits the actual transfer curve (after removing of offset error), and the straight line which fits the ideal transfer curve (see Figure 23-20).

23.3.7.3 Waveforms

Figure 23-19. Analog-to-Digital Converter Internal Waveforms

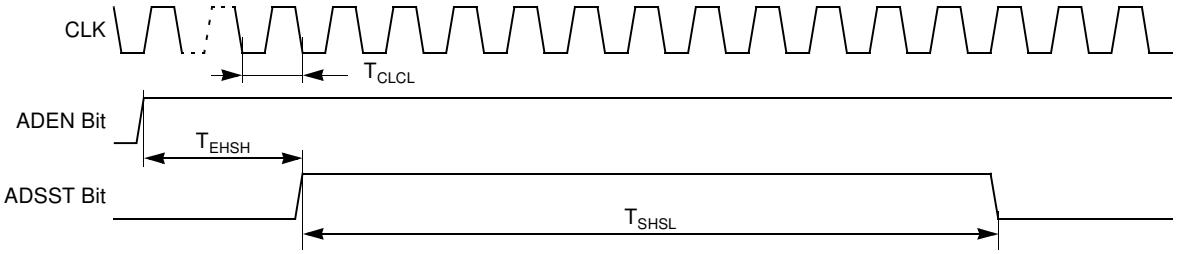
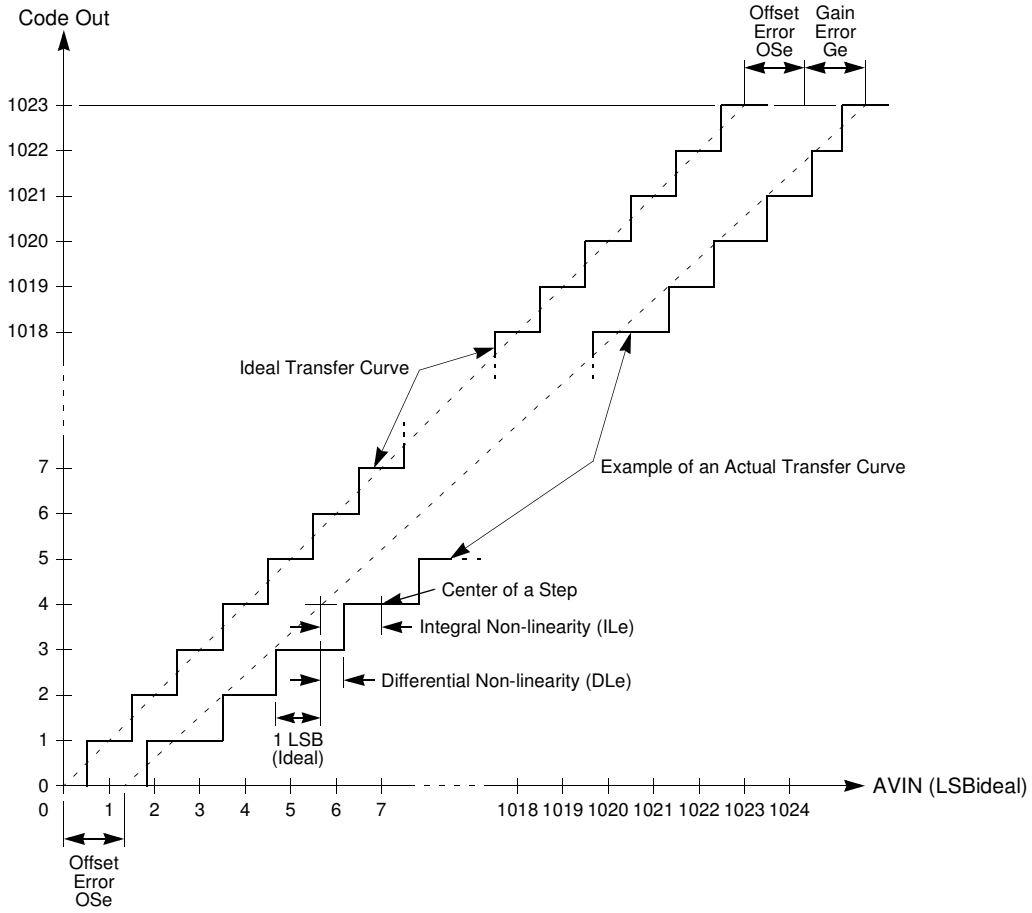


Figure 23-20. Analog-to-Digital Converter Characteristics



23.3.8 Flash Memory

23.3.8.1 Definition of Symbols

Table 121. Flash Memory Timing Symbol Definitions

Signals	
S	$\overline{ISP}$
R	RST
B	FBUSY flag

Conditions	
L	Low
V	Valid
X	No Longer Valid

23.3.8.2 Timings

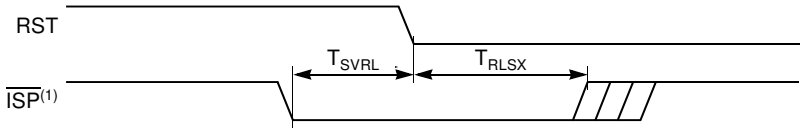
Table 122. Flash Memory AC Timing

V_{DD} = 2.7 to 3.3V, T_A = -40° to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
T_{SVRL}	Input $\overline{ISP}$ Valid to RST Edge	50			ns
T_{RLSX}	Input $\overline{ISP}$ Hold after RST Edge	50			ns
T_{BHBL}	FLASH Internal Busy (Programming) Time		10		ms
N_{FCY}	Number of Flash Write Cycles	100K			Cycle
T_{FDR}	Flash Data Retention Time	10			Year

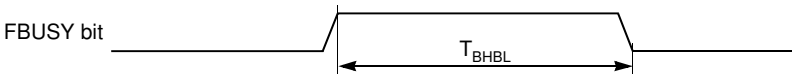
23.3.8.3 Waveforms

Figure 23-21. Flash Memory – ISP Waveforms



Note: 1. $\overline{ISP}$ must be driven through a pull-down resistor (see Section “In-system Programming”, page 158).

Figure 23-22. Flash Memory – Internal Busy Waveforms



23.3.9 External Clock Drive and Logic Level References

23.3.9.1 Definition of Symbols

Table 123. External Clock Timing Symbol Definitions

Signals	
C	Clock

Conditions	
H	High
L	Low
X	No Longer Valid

23.3.9.2 Timings

Table 124. External Clock AC Timings

V_{DD} = 2.7 to 3.3V, T_A= -40 to +85°C

Symbol	Parameter	Min	Max	Unit
T_{CLCL}	Clock Period	50		ns
T_{CHCX}	High Time	10		ns
T_{CLCX}	Low Time	10		ns
T_{CLCH}	Rise Time	3		ns
T_{CHCL}	Fall Time	3		ns
T_{CR}	Cyclic Ratio in X2 Mode	40	60	%

23.3.9.3 Waveforms

Figure 23-23. External Clock Waveform

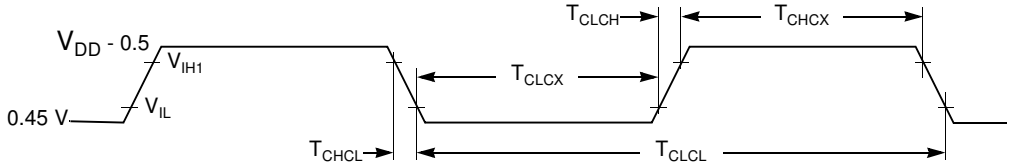
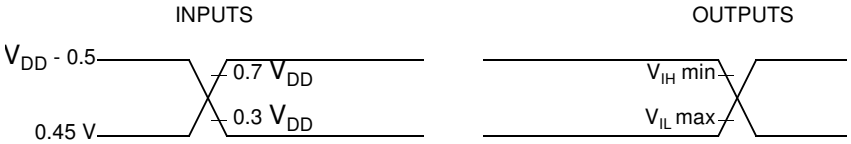
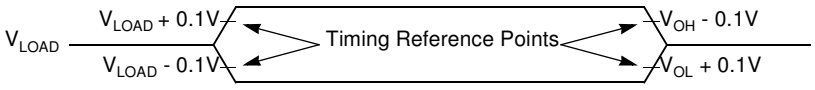


Figure 23-24. AC Testing Input/Output Waveforms



Notes: 1. During AC testing, all inputs are driven at V_{DD} -0.5V for a logic 1 and 0.45V for a logic 0.
2. Timing measurements are made on all outputs at V_{IH} min for a logic 1 and V_{IL} max for a logic 0.

Figure 23-25. Float Waveforms



Note: For timing purposes, a 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 loading V_{OH}/V_{OL} level occurs with I_{OL}/I_{OH} = ±20 mA.

24. Ordering Information

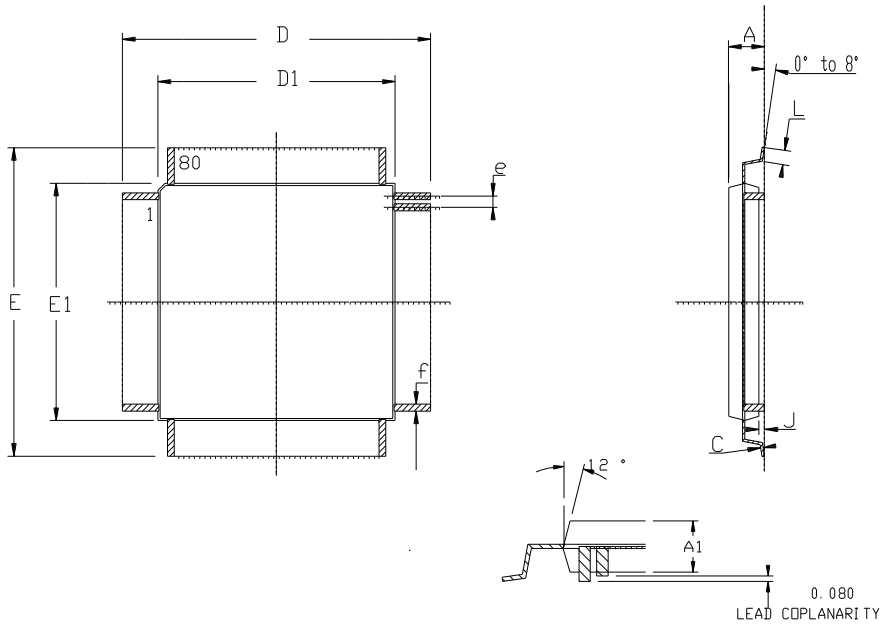
Possible Order Entries[1]

Part Number	Memory Size (Bytes)	Supply Voltage	Temperature Range	Max Frequency (MHz)	Package	Packing	Product Marking
AT89C5132-ROTIL	64K Flash	3V	Industrial	40	TQFP80	Tray	895132-IL
AT89C5132-ROTUL	64K Flash	3V	Industrial & Green	40	TQFP80	Tray	895132-UL

Note: 1. PLCC84 package only available for development board.

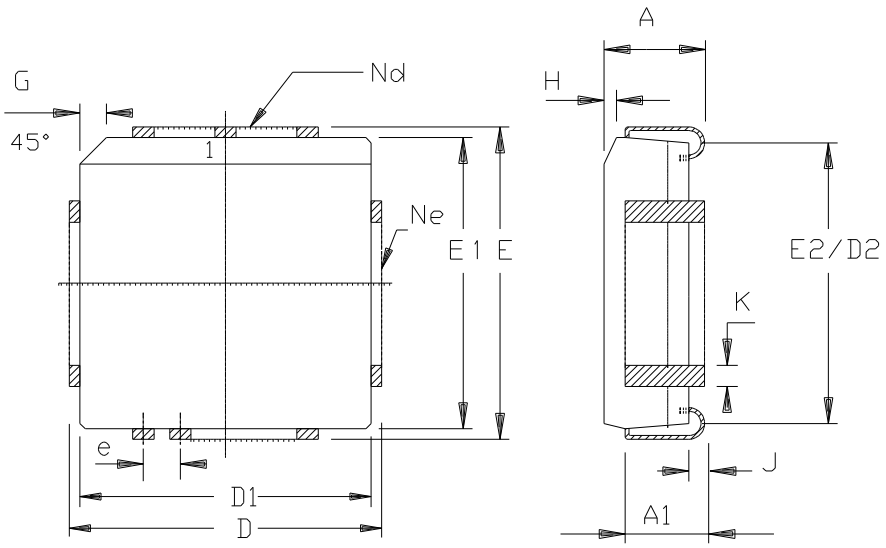
25. Package Information

25.1 TQFP80



	MM		INCH	
	Min	Max	Min	Max
A	1.40	1.60	.055	.063
A1	1.35	1.45	.053	.057
C	0.17 BSC		.007 BSC	
D	15.80	16.20	.622	.638
D1	13.90	14.10	.547	.555
E	15.80	16.20	.622	.638
E1	13.90	14.10	.547	.555
J	0.05	0.15	.002	.006
L	0.45	0.75	.018	.030
e	0.65 BSC		.0256 BSC	
f	0.30 BSC		.012 BSC	

25.2 PLCC84



	MM		INCH	
A	4.20	5.08	.165	.200
A1	2.29	3.30	.090	.130
D	30.10	30.35	1.185	1.195
D1	29.21	29.41	1.150	1.158
D2	27.69	28.70	1.090	1.130
E	30.10	30.35	1.185	1.195
E1	29.21	29.41	1.150	1.158
E2	27.69	28.70	1.090	1.130
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	21		21	
Ne	21		21	
PKG STD	00			

26. Datasheet Revision History for AT89C5132

26.1 Changes from 4173A-08/02 to 4173B-03/04

1. Suppression of ROM product version.
2. Suppression of TQFP64 package.

26.2 Changes from 4173B-03/04 - 4173C - 07/04

1. Add USB connection schematic in USB section.
2. Add USB termination characteristics in DC Characteristics section.
3. Page access mode clarification in Data Memory section.

26.3 Changes from 4173C-07/04 - 4173D - 01/05

1. Interrupt priority number clarification to match number defined by development tools.

26.4 Changes from to 4317D - 01/05 to 4173E - 09/07

1. Added green product ordering information.
2. Removed ‘Preliminary’ status. Product now fully Industrialised.

1 Description 1
2 Typical Applications 1
3 Block Diagram 2
4 Pin Description 3
4.1 Signals 4
4.2 Internal Pin Structure 10
5 Address Spaces 11
6 Clock Controller 12
6.1 Oscillator 12
6.2 X2 Feature 12
6.3 PLL 13
6.4 Registers 14
7 Program/Code Memory 17
7.1 Flash Memory Architecture 17
7.2 Hardware Security System 18
7.3 Boot Memory Execution 19
7.4 Registers 20
7.5 Hardware Bytes 20
8 Data Memory 22
8.1 Internal Space 22
8.2 External Space 23
8.3 Dual Data Pointer 26
8.4 Registers 27
9 Special Function Registers 29
10 Interrupt System 34
10.1 Interrupt System Priorities 34
10.2 External Interrupts 37
10.3 Registers 38
11 Power Management 44
11.1 Reset 44
11.2 Reset Recommendation to Prevent Flash Corruption 45
11.3 Idle Mode 46
11.4 Power-down Mode 46

11.5 Registers 48

12 Timers/Counters 49

12.1 Timer/Counter Operations 49
12.2 Timer Clock Controller 49
12.3 Timer 0 50
12.4 Timer 1 52
12.5 Interrupt 53
12.6 Registers 54

13 Watchdog Timer 57

13.1 Description 57
13.2 Watchdog Clock Controller 57
13.3 Watchdog Operation 58
13.4 Registers 59

14 Audio Output Interface 60

14.1 Description 60
14.2 Clock Generator 60
14.3 Data Converter 61
14.4 Audio Buffer 62
14.5 Interrupt Request 63
14.6 Voice or Sound Playing 63
14.7 Registers 64

15 Universal Serial Bus 67

15.1 Description 68
15.2 USB Interrupt System 70
15.3 Registers 72

16 MultiMedia Card Controller 82

16.1 Card Concept 82
16.2 Bus Concept 82
16.3 Description 87
16.4 Clock Generator 88
16.5 Command Line Controller 88
16.6 Data Line Controller 90
16.7 Interrupt 96
16.8 Registers 97

17 IDE/ATAPI Interface 104

17.1 Description 104

17.2 Registers 106

18 Serial I/O Port 107

18.1 Mode Selection 107

18.2 Baud Rate Generator 107

18.3 Synchronous Mode (Mode 0) 108

18.4 Asynchronous Modes (Modes 1, 2 and 3) 111

18.5 Multiprocessor Communication (Modes 2 and 3) 114

18.6 Automatic Address Recognition 114

18.7 Interrupt 116

18.8 Registers 116

19 Synchronous Peripheral Interface 120

19.1 Description 121

19.2 Interrupt 124

19.3 Configuration 124

19.4 Registers 129

20 Two-wire Interface (TWI) Controller 131

20.1 Description 131

20.2 Registers 145

21 Analog to Digital Converter 147

21.1 Description 147

21.2 Registers 150

22 Keyboard Interface 152

22.1 Description 152

22.2 Registers 152

23 Electrical Characteristics 154

23.1 Absolute Maximum Ratings 154

23.2 DC Characteristics 154

23.3 AC Characteristics 159

24 Ordering Information 174

25 Package Information 175

25.1 TQFP80 175

25.2 PLCC84 176

26 Datasheet Revision History for AT89C5132 177

26.1 Changes from 4173A-08/02 to 4173B-03/04 177

26.2 Changes from 4173B-03/04 - 4173C - 07/04 177

26.3 Changes from 4173C-07/04 - 4173D - 01/05 177

26.4 Changes from to 4317D - 01/05 to 4173E - 09/07 177

Atmel Corporation

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

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
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, USA
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, USA
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

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically providedotherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel'sAtmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2007. **All rights reserved.** Atmel®, logo and combinations thereof, are registered trademarks, and Everywhere You Are® are the trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.


Printed on recycled paper.

4173E–USB–09/07