Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	80C51
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	LED, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.13x19.13)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at89c5131a-s3sul

Signals

All the AT89C5131A-L signals are detailed by functionality on Table 1 through Table 12.

Table 1. Keypad Interface Signal Description

Signal Name	Type	Description	Alternate Function
KIN[7:0]	I	**Keypad Input Lines** Holding one of these pins high or low for 24 oscillator periods triggers a keypad interrupt if enabled. Held line is reported in the KBCON register.	P1[7:0]

Table 2. Programmable Counter Array Signal Description

Signal Name	Type	Description	Alternate Function
ECI	I	**External Clock Input**	P1.2
CEX[4:0]	I/O	Capture External Input Compare External Output	P1.3 P1.4 P1.5 P1.6 P1.7

Table 3. Serial I/O Signal Description

Signal Name	Type	Description	Alternate Function
RxD	I	**Serial Input** The serial input for Extended UART.	P3.0
TxD	O	**Serial Output** The serial output for Extended UART.	P3.1

Table 4. Timer 0, Timer 1 and Timer 2 Signal Description

Signal Name	Type	Description	Alternate Function
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 set IE0 in the TCON register. If bit IT0 in this register is set, bits IE0 are set by a falling edge on $\overline{INT0}$. If bit IT0 is cleared, bits IE0 is set by a low level on $\overline{INT0}$.	P3.2
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 set IE1 in the TCON register. If bit IT1 in this register is set, bits IE1 are set by a falling edge on $\overline{INT1}$. If bit IT1 is cleared, bits IE1 is set by a low level on $\overline{INT1}$.	P3.3

Clock Controller

Introduction

The AT89C5131A-L clock controller is based on an on-chip oscillator feeding an on-chip Phase Lock Loop (PLL). All the internal clocks to the peripherals and CPU core are generated by this controller.

The AT89C5131A-L X1 and X2 pins are the input and the output of a single-stage on-chip inverter (see Figure 7) that can be configured with off-chip components as a Pierce oscillator (see Figure 8). Value of capacitors and crystal characteristics are detailed in the section “DC Characteristics”.

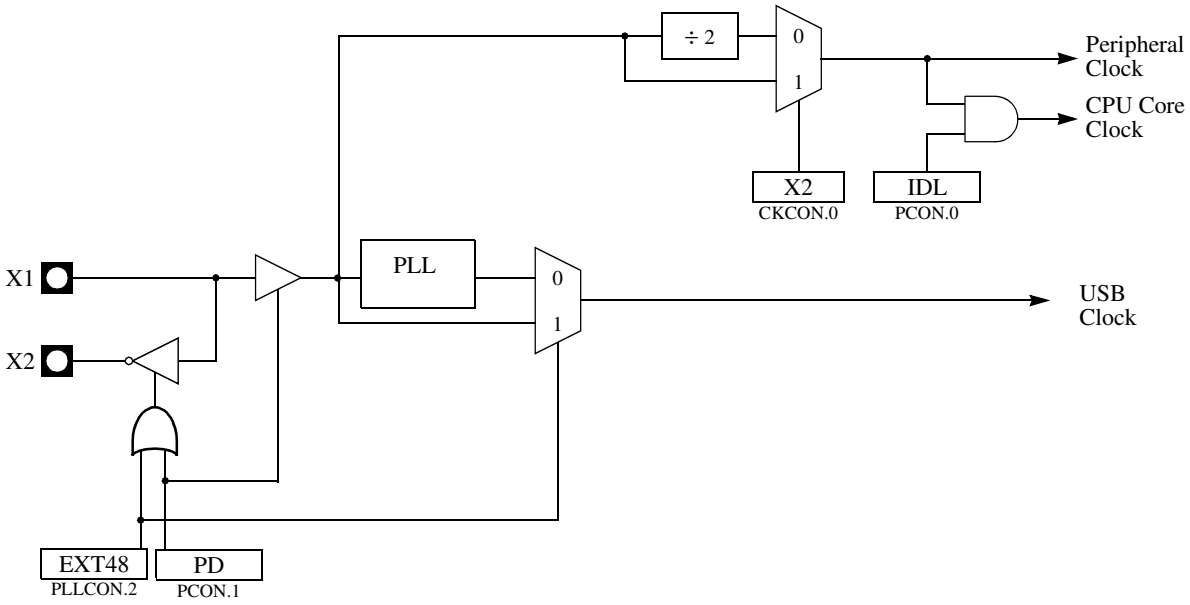
The X1 pin can also be used as input for an external 48 MHz clock.

The clock controller outputs three different clocks as shown in Figure 7:

- a clock for the CPU core
- a clock for the peripherals which is used to generate the Timers, PCA, WD, and Port sampling clocks
- a clock for the USB controller

These clocks are enabled or disabled depending on the power reduction mode as detailed in Section “Power Management”, page 152.

Figure 7. Oscillator Block Diagram



Oscillator

Two clock sources are available for CPU:

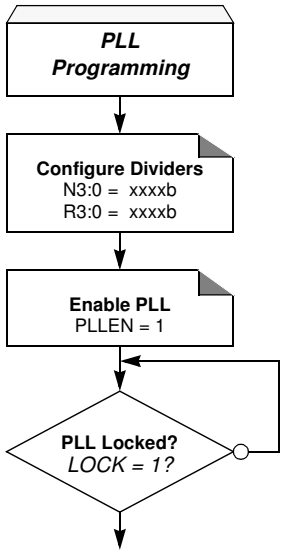
- Crystal oscillator on X1 and X2 pins: Up to 32 MHz
- External 48 MHz clock on X1 pin

In order to optimize the power consumption, the oscillator inverter is inactive when the PLL output is not selected for the USB device.

PLL Programming

The PLL is programmed using the flow shown in Figure 11. As soon as clock generation is enabled user must wait until the lock indicator is set to ensure the clock output is stable.

Figure 11. PLL Programming Flow



Divider Values

To generate a 48 MHz clock using the PLL, the divider values have to be configured following the oscillator frequency. The typical divider values are shown in Table 13.

Table 13. Typical Divider Values

Oscillator Frequency	R+1	N+1	PLLDIV
3 MHz	16	1	F0h
6 MHz	8	1	70h
8 MHz	6	1	50h
12 MHz	4	1	30h
16 MHz	3	1	20h
18 MHz	8	3	72h
20 MHz	12	5	B4h
24 MHz	2	1	10h
32 MHz	3	2	21h
40 MHz	12	10	B9h

The Special Function Registers (SFRs) of the AT89C5131 fall into the following categories:

Table 19. 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								
SP	81h	Stack Pointer LSB of SPX								
DPL	82h	Data Pointer Low byte LSB of DPTR								
DPH	83h	Data Pointer High byte MSB of DPTR								

Table 20. I/O Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P0	80h	Port 0								
P1	90h	Port 1								
P2	A0h	Port 2								
P3	B0h	Port 3								
P4	C0h	Port 4 (2bits)								

ASSEMBLY LANGUAGE

```
; Block move using dual data pointers
; Modifies DPTR0, DPTR1, A and PSW
; note: DPS exits opposite of entry state
; unless an extra INC AUXR1 is added
;
00A2    AUXR1 EQU 0A2H
;
0000 909000MOV DPTR,#SOURCE ; address of SOURCE
0003 05A2 INC AUXR1 ; switch data pointers
0005 90A000 MOV DPTR,#DEST ; address of DEST
0008    LOOP:
0008 05A2 INC AUXR1 ; switch data pointers
000A E0 MOVX A,@DPTR ; get a byte from SOURCE
000B A3 INC DPTR ; increment SOURCE address
000C 05A2 INC AUXR1 ; switch data pointers
000E F0 MOVX @DPTR,A ; write the byte to DEST
000F A3 INC DPTR ; increment DEST address
0010 70F6JNZ LOOP ; check for 0 terminator
0012 05A2 INC AUXR1 ; (optional) restore DPS
```

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

Boot Process

Software Boot Process Example

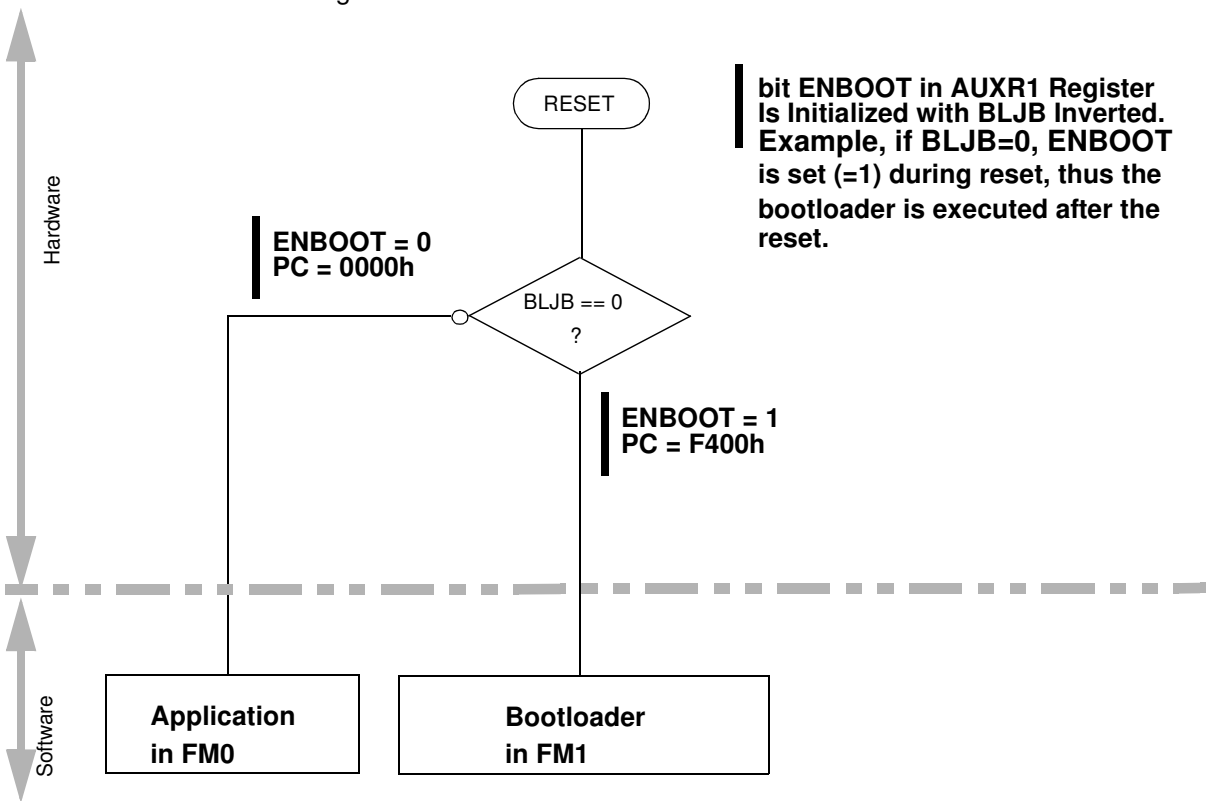
Many algorithms can be used for the software boot process. Below are descriptions of the different flags and Bytes.

Boot Loader Jump bit (BLJB):
- This bit indicates if on RESET the user wants to jump to this application at address @0000h on FM0 or execute the boot loader at address @F400h on FM1.
- BLJB = 0 (i.e. bootloader FM1 executed after a reset) is the default Atmel factory programming.
-To read or modify this bit, the APIs are used.

Boot Vector Address (SBV):
- This byte contains the MSB of the user boot loader address in FM0.
- The default value of SBV is FFh (no user boot loader in FM0).
- To read or modify this byte, the APIs are used.

Extra Byte (EB) & Boot Status Byte (BSB):
- These Bytes are reserved for customer use.
- To read or modify these Bytes, the APIs are used.

Figure 23. Hardware Boot Process Algorithm



As PSEN is an output port in normal operating mode (running user application or bootloader code) after reset, it is recommended to release PSEN after rising edge of reset signal.

Low Pin Count Hardware Conditions (SOIC28)

Low pin count products do not have PSEN signal, thus for these products, the bootloader is always executed after reset thanks to the BLJB bit. The Hardware Conditions are detected at the begining of the bootloader execution from reset.

The default factory Hardware Condition is assigned to port P1.

- P1 must be equal to FEh

In order to offer the best flexibility, the user can define its own Hardware Condition on one of the following Ports:

- Port1
- Port3
- Port4 (only bit0 and bit1)

The Hardware Conditions configuration is stored in three bytes called P1_CF, P3_CF, P4_CF.

These bytes can be modified by the user through a set of API or through an ISP command.

Note:
1. The BLJB must be at 0 (programmed) to be able to restart the bootloader.
2. BLJB can always be changed by the means of API, whether it's a low or high pin count package.But for a low pin count version, if BLJB=1, no ISP via the Bootloader is further possible (because the HW conditions are never evaluated, as described in the USB Bootloader Datasheet). To go back to ISP, BLJB needs to be changed by a parallel programmer(or by the APIs).

See a detailed description in the applicable Document.

– Datasheet Bootloader USB AT89C5131.

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

- Instructions that use direct addressing access SFR space. For example: MOV 0A0H, # data, accesses the SFR at location 0A0h (which is P2).
- Instructions that use indirect addressing access the Upper 128 bytes of data RAM. For example: MOV atR0, # data where R0 contains 0A0h, accesses the data byte at address 0A0h, rather than P2 (whose address is 0A0h).
- The ERAM bytes can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory which is physically located on-chip, logically occupies the first bytes of external data memory. The bits XRS0 and XRS1 are used to hide a part of the available ERAM as explained in Table 44. This can be useful if external peripherals are mapped at addresses already used by the internal ERAM.
- With EXTRAM = 0, the ERAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to ERAM will not affect ports P0, P2, P3.6 (WR) and P3.7 (RD). For example, with EXTRAM = 0, MOVX atR0, # data where R0 contains 0A0H, accesses the ERAM at address 0A0H rather than external memory. An access to external data memory locations higher than the accessible size of the ERAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51, with P0 and P2 as data/address busses, and P3.6 and P3.7 as write and read timing signals. Accesses to ERAM above 0FFH can only be done by the use of DPTR.
- With EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard 80C51. MOVX at Ri will provide an eight-bit address multiplexed with data on Port0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a sixteen-bit address. Port2 outputs the high-order eight address bits (the contents of DPH) while Port0 multiplexes the low-order eight address bits (DPL) with data. MOVX at Ri and MOVX @DPTR will generate either read or write signals on P3.6 ($\overline{WR}$) and P3.7 ($\overline{RD}$).

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

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

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

Figure 27. Clock-out Mode C/$\overline{T2}$ = 0

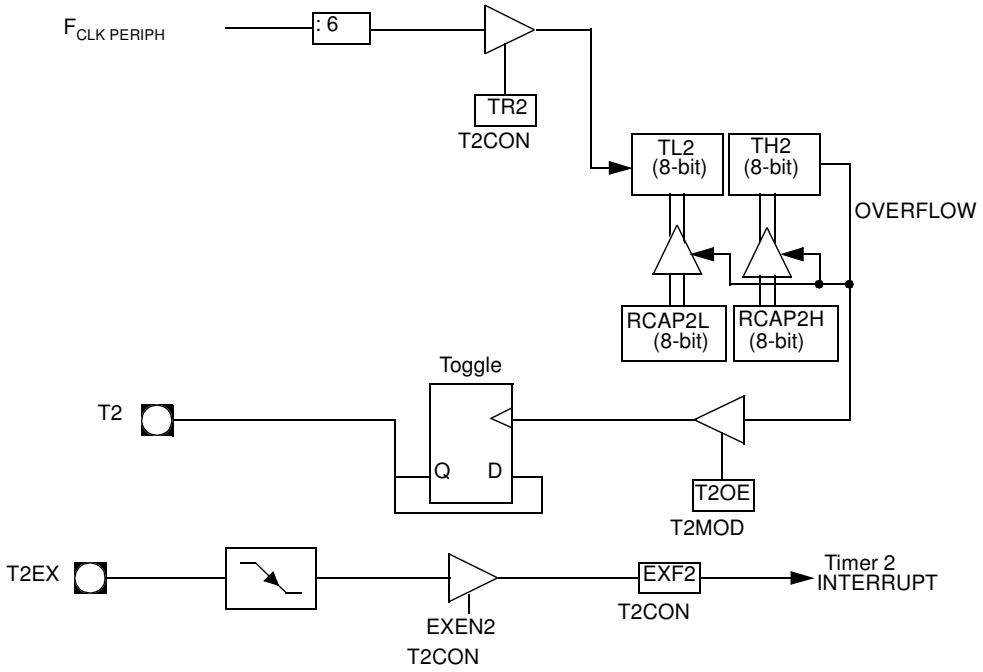


Table 46. T2CON Register

T2CON - Timer 2 Control Register (C8h)

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

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

Reset Value = 0000 0000b

Bit addressable

Programmable Counter Array (PCA)

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

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

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

- rising and/or falling edge capture,
- software timer
- high-speed output, or
- pulse width modulator

Module 4 can also be programmed as a watchdog timer (see Section "PCA Watchdog Timer", page 65).

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

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

PCA Component	External I/O Pin
16-bit Counter	P1.2/ECI
16-bit Module 0	P1.3/CEX0
16-bit Module 1	P1.4/CEX1
16-bit Module 2	P1.5/CEX2
16-bit Module 3	P1.6/CEX3
16-bit Module 4	P1.7/CEX4

The PCA timer is a common time base for all five modules (see Figure 28). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD register (Table 48) and can be programmed to run at:

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

the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition.

- The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function.

Table 51 shows the CCAPMn settings for the various PCA functions.

Table 50. CCAPMn Registers (n = 0-4)

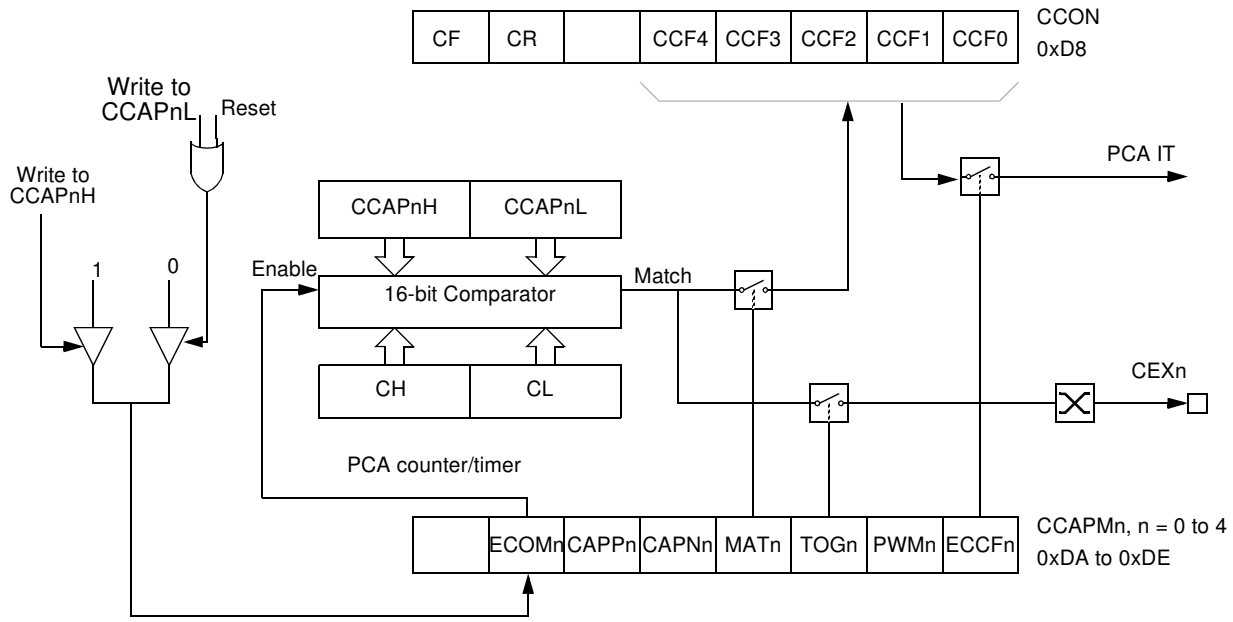
CCAPM0 - PCA Module 0 Compare/Capture Control Register (0DAh)
CCAPM1 - PCA Module 1 Compare/Capture Control Register (0DBh)
CCAPM2 - PCA Module 2 Compare/Capture Control Register (0DCh)
CCAPM3 - PCA Module 3 Compare/Capture Control Register (0DDh)
CCAPM4 - PCA Module 4 Compare/Capture Control Register (0DEh)

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

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

Reset Value = X000 0000b
Not bit addressable

Figure 32. PCA High-speed Output Mode



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

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

Pulse Width Modulator Mode

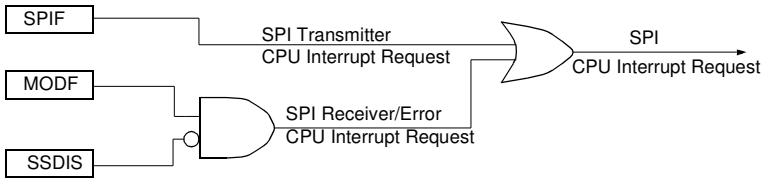
All of the PCA modules can be used as PWM outputs. Figure 33 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. This allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

Interrupt Sources and Vector Addresses

Table 67. Vector Table

Number	Polling Priority	Interrupt Source	Interrupt Request	Vector Address
0	0	Reset		0000h
1	1	INT0	IE0	0003h
2	2	Timer 0	TF0	000Bh
3	3	INT1	IE1	0013h
4	4	Timer 1	IF1	001Bh
5	6	UART	RI+TI	0023h
6	7	Timer 2	TF2+EXF2	002Bh
7	5	PCA	CF + CCFn (n = 0-4)	0033h
8	8	Keyboard	KBDIT	003Bh
9	9	TWI	TWIIT	0043h
10	10	SPI	SPIIT	004Bh
11	11			0053h
12	12			005Bh
13	13			0063h
14	14	USB	UEPINT + USBINT	006Bh
15	15			0073h

Figure 48. SPI Interrupt Requests Generation



Registers

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

Serial Peripheral Control Register (SPCON)

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

Table 74 describes this register and explains the use of each bit.

Table 74. SPCON Register

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

Bit Number	Bit Mnemonic	Description
7	SPR2	**Serial Peripheral Rate 2** Bit with SPR1 and SPR0 define the clock rate.
6	SPEN	**Serial Peripheral Enable** Cleared to disable the SPI interface. Set to enable the SPI interface.
5	SSDIS	**$\overline{SS}$ Disable** Cleared to enable $\overline{SS}$ in both Master and Slave modes. Set to disable $\overline{SS}$ in both Master and Slave modes. In Slave mode, this bit has no effect if CPHA = “0”.
5	MSTR	**Serial Peripheral Master** Cleared to configure the SPI as a Slave. Set to configure the SPI as a Master.
4	CPOL	**Clock Polarity** Cleared to have the SCK set to “0” in idle state. Set to have the SCK set to “1” in idle state.
3	CPHA	**Clock Phase** Cleared to have the data sampled when the SCK leaves the idle state (see CPOL). Set to have the data sampled when the SCK returns to idle state (see CPOL).

have been received, the serial interrupt flag is set and a valid status code can be read from SSCS. This status code is used to vector to an interrupt service routine. The appropriate action to be taken for each of these status code is detailed in Table . The slave transmitter mode may also be entered if arbitration is lost while the TWI module is in the master mode.

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

Miscellaneous States

There are two SSCS codes that do not correspond to a define TWI hardware state (Table 85). These codes are discuss hereafter.

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

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

Notes

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

Table 80. Bit Frequency Configuration

			Bit Frequency (kHz)		
CR2	**CR1**	**CR0**	**F_{OSCA}= 12 MHz**	**F_{OSCA} = 16 MHz**	**F_{OSCA} divided by**
0	0	0	47	62.5	256
0	0	1	53.5	71.5	224
0	1	0	62.5	83	192
0	1	1	75	100	160
1	0	0	-	-	Unused
1	0	1	100	133.3	120
1	1	0	200	266.6	60
1	1	1	0.5 <. < 62.5	0.67 <. < 83	Timer 1 in mode 2 can be used as TWI baudrate generator with the following formula: 96.(256-"Timer1 reload value")

- Endpoint enable

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

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

- Endpoint type configuration

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

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

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

- Endpoint direction configuration

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

  – IN:EPDIR = 1b
  – OUT:EPDIR = 0b

  For Control endpoints, the EPDIR bit has no effect.

- Summary of Endpoint Configuration:

  Do not forget to select the correct endpoint number in the UEPNUM register before accessing to endpoint specific registers.

Table 90. Summary of Endpoint Configuration

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

ONCE Mode (ON Chip Emulation)

The ONCE mode facilitates testing and debugging of systems using AT89C5131A-L without removing the circuit from the board. The ONCE mode is invoked by driving certain pins of the AT89C5131A-L; the following sequence must be exercised:

- Pull ALE low while the device is in reset (RST high) and $\overline{\text{PSEN}}$ is high.
- Hold ALE low as RST is deactivated.

While the AT89C5131A-L is in ONCE mode, an emulator or test CPU can be used to drive the circuit Table 111 shows the status of the port pins during ONCE mode.

Normal operation is restored when normal reset is applied.

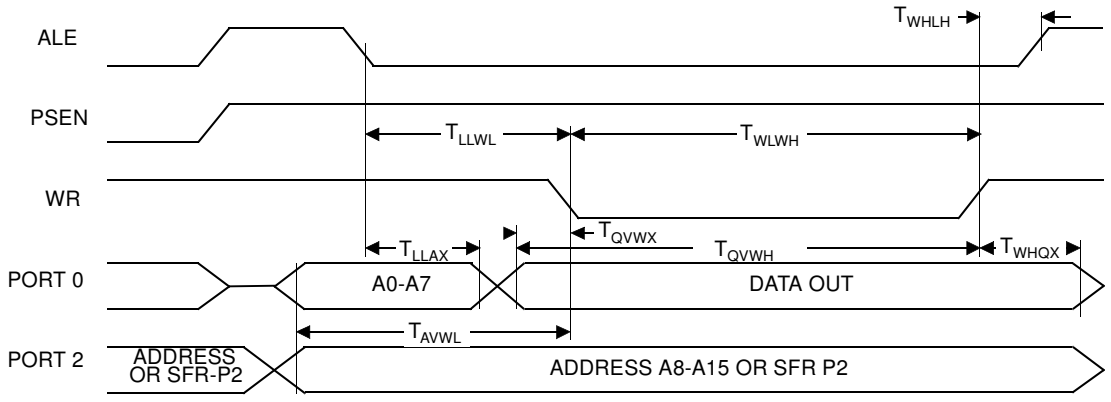
Table 111. External Pin Status during ONCE Mode

ALE	PSEN	Port 0	Port 1	Port 2	Port 3	Port I2	XTAL1/2
Weak pull-up	Weak pull-up	Float	Weak pull-up	Weak pull-up	Weak pull-up	Float	Active

Table 119. AC Parameters for a Variable Clock

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

External Data Memory Write Cycle



Document Revision History

Changes from 4338D - 09/05 to 4338E - 06/06

1. Correction to Figure 4 on page 11.

Changes from 4338E - 06/06 to 4338F - 08/07

1. Hardware Conditions section Page 45 changed to recommend the use of 1K pull-up between PSEN and GND in ISP mode.
2. Updated 52-lead PLCC package.