Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 256 Bytes of On-chip ERAM
  - 16-KB of On-chip Flash Memory
  - Data Retention: 10 Years at 85°C
  - Read/Write Cycle: 10K
- 2K Bytes of On-chip Flash for Bootloader
- 2K Bytes of On-chip EEPROM
  - Read/Write Cycle: 100k
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- Maximum Crystal Frequency 40 MHz
  - In X2 Mode, 20 MHz (CPU core, 40 MHz)
- Three or Four Ports: 16 or 20 Digital I/O Lines
- Two-channel 16-bit PCA with:
  - PWM (8-bit)
  - High-speed Output
  - Timer and Edge Capture
- Double Data Pointer
- 21-bit WatchDog Timer (7 Programmable Bits)
- A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- Power Saving Modes:
  - Idle Mode
  - Power-down Mode
- Power Supply: 5V ± 10% (or 3V(1) ± 10%)
- Temperature Range: Industrial (-40° to +85°C)
- Packages: SOIC28, PLCC28, VQFP32

Note: 1. Ask for availability

Low Pin Count 8-bit MCU with A/D Converter and 16-Kbytes of Flash Memory

T89C5115

Description

The T89C5115 is a high performance Flash version of the 80C51 single chip 8-bit microcontrollers. It contains a 16-KB Flash memory block for program and data.

The 16-KB Flash memory can be programmed either in parallel mode or in serial mode with the ISP capability or with software. The programming voltage is internally generated from the standard VCC pin.

The T89C5115 retains all features of the 80C52 with 256 bytes of internal RAM, a 7-source 4-level interrupt controller and three timer/counters. In addition, the T89C5115 has a 10-bit A/D converter, a 2-KB Boot Flash memory, 2-KB EEPROM for data, a Programmable Counter Array, an ERAM of 256 bytes, a Hardware WatchDog Timer and a more versatile serial channel that facilitates multiprocessor communication (EUART). The fully static design of the T89C5115 reduces system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

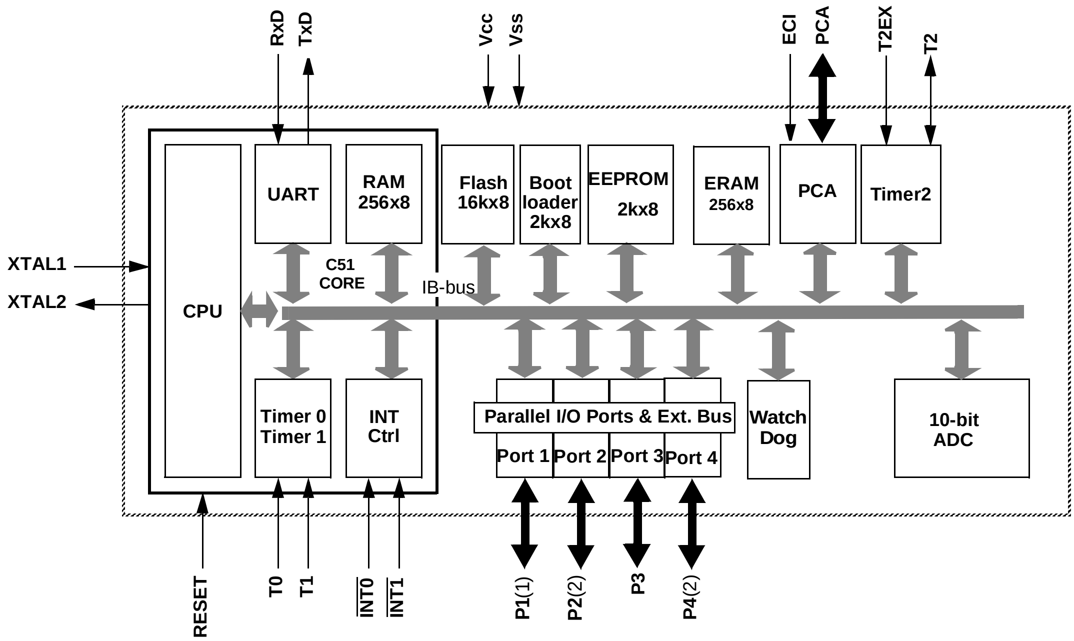
The T89C5115 has two software-selectable modes of reduced activity and an 8 bit clock prescaler for further reduction in power consumption. In the idle mode the CPU is frozen while the peripherals and the interrupt system are still operating. In the power-down mode the RAM is saved and all other functions are inoperative.

The added features of the T89C5115 make it more powerful for applications that need A/D conversion, pulse width modulation, high speed I/O and counting capabilities such as industrial control, consumer goods, alarms, motor control, etc. While remaining fully compatible with the 80C52 it offers a superset of this standard microcontroller.

Rev. 4128A–8051–04/02

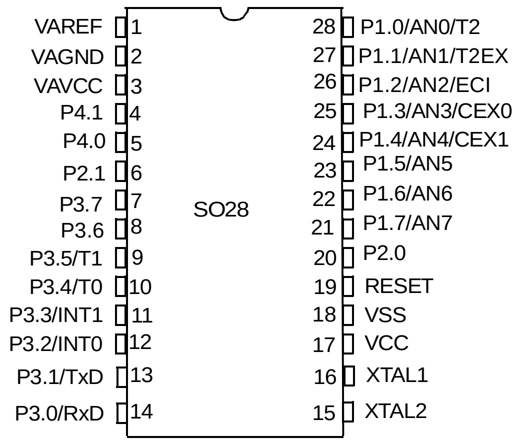
In X2 mode a maximum external clock rate of 20 MHz reaches a 300 ns cycle time.

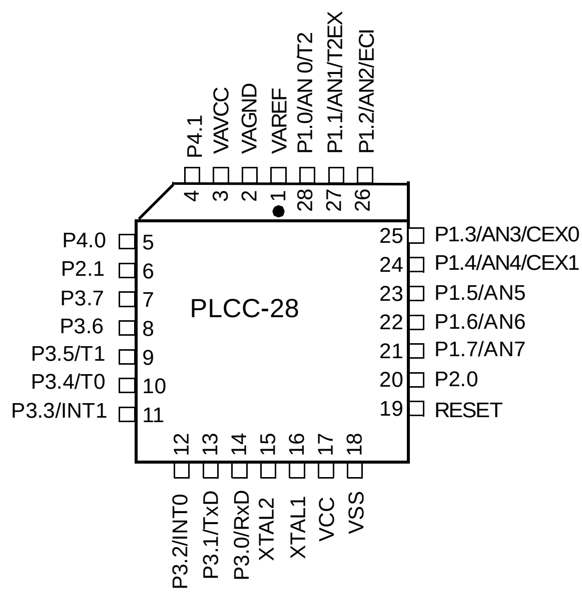
Block Diagram



Notes: 1. 8 analog Inputs/8 Digital I/O
2. 2-Bit I/O Port

Pin Configuration





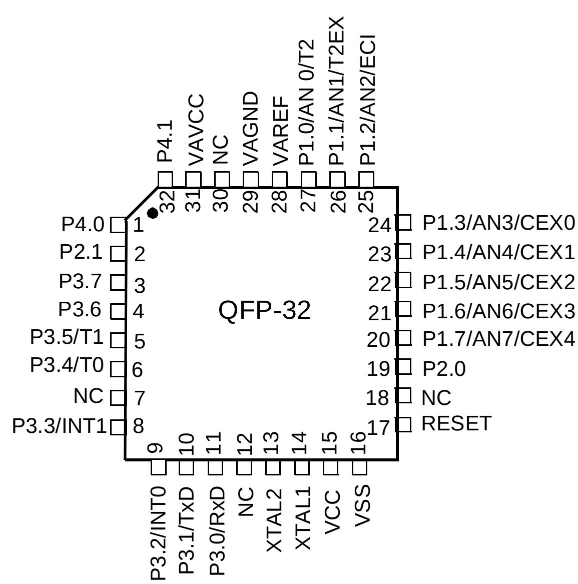


Table 1. Pin Description

Pin Name	Type	Description
VSS	GND	**Circuit ground**
VCC		**Supply Voltage**
VAREF		Reference Voltage for ADC
VAVCC		Supply Voltage for ADC
VAGND		Reference Ground for ADC
P1.0:7	I/O	**Port 1:** Is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins can be used for digital input/output or as analog inputs for the Analog Digital Converter (ADC). Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 1 pins that are being pulled low externally will be the source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 1 pins are assigned to be used as analog inputs via the ADCCF register (in this case the internal pull-ups are disconnected). As a secondary digital function, port 1 contains the Timer 2 external trigger and clock input; the PCA external clock input and the PCA module I/O. P1.0/AN0/T2 Analog input channel 0, External clock input for Timer/counter2. P1.1/AN1/T2EX Analog input channel 1, Trigger input for Timer/counter2. P1.2/AN2/ECI Analog input channel 2, PCA external clock input. P1.3/AN3/CEX0 Analog input channel 3, PCA module 0 Entry of input/PWM output. P1.4/AN4/CEX1 Analog input channel 4, PCA module 1 Entry of input/PWM output. P1.5/AN5 Analog input channel 5, P1.6/AN6 Analog input channel 6, P1.7/AN7 Analog input channel 7, It can drive CMOS inputs without external pull-ups.
P2.0:7	I/O	**Port 2:** Is an 2-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 2 pins that are being pulled low externally will be a source of current (IIL, on the datasheet) because of the internal pull-ups. In the T89C51CC02 Port 2 can sink or source 5 mA. It can drive CMOS inputs without external pull-ups.

Table 1. Pin Description (Continued)

Pin Name	Type	Description
P3.0:7	I/O	**Port 3:** Is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 3 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. The output latch corresponding to a secondary function must be programmed to one for that function to operate (except for TxD). The secondary functions are assigned to the pins of port 3 as follows: P3.0/RxD: Receiver data input (asynchronous) or data input/output (synchronous) of the serial interface P3.1/TxD: Transmitter data output (asynchronous) or clock output (synchronous) of the serial interface P3.2/$\overline{\text{INT0}}$: External interrupt 0 input/timer 0 gate control input P3.3/$\overline{\text{INT1}}$: External interrupt 1 input/timer 1 gate control input P3.4/T0: Timer 0 counter input P3.5/T1: Timer 1 counter input It can drive CMOS inputs without external pull-ups.
P4.0:1	I/O	**Port 4:** Is an 2-bit bi-directional I/O port with internal pull-ups. Port 4 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 4 pins that are being pulled low externally will be a source of current (IIL, on the datasheet) because of the internal pull-up transistor. It can drive CMOS inputs without external pull-ups.
RESET	I/O	**Reset:** A high level on this pin during two machine cycles while the oscillator is running resets the device. An internal pull-down resistor to VSS permits power-on reset using only an external capacitor to VCC.
XTAL1	I	**XTAL1:** Input of the inverting oscillator amplifier and input of the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. To operate above a frequency of 16 MHz, a duty cycle of 50% should be maintained.
XTAL2	O	**XTAL2:** Output from the inverting oscillator amplifier.

I/O Configurations

Each Port SFR operates via type-D latches, as illustrated in Figure 1 for Ports 3 and 4. A CPU "write to latch" signal initiates transfer of internal bus data into the type-D latch. A CPU "read latch" signal transfers the latched Q output onto the internal bus. Similarly, a "read pin" signal transfers the logical level of the Port pin. Some Port data instructions activate the "read latch" signal while others activate the "read pin" signal. Latch instructions are referred to as Read-Modify-Write instructions. Each I/O line may be independently programmed as input or output.

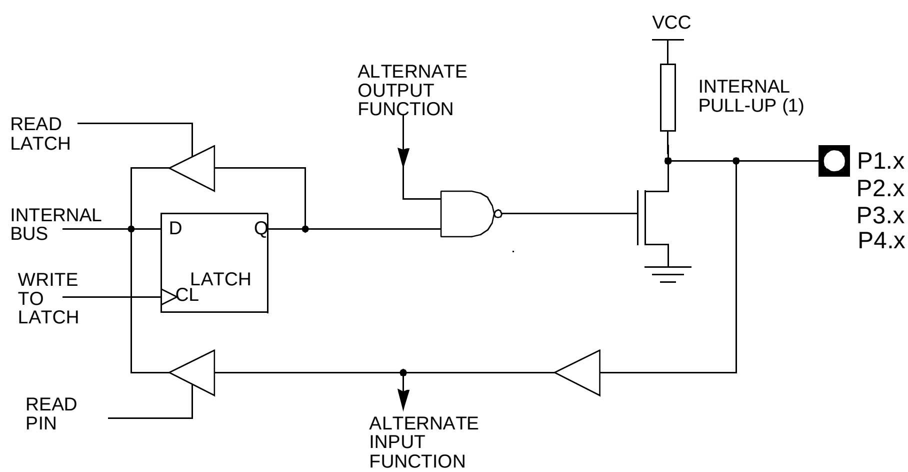
Port Structure

Figure 1 shows the structure of Ports, which have internal pull-ups. An external source can pull the pin low. Each Port pin can be configured either for general-purpose I/O or for its alternate input output function.

To use a pin for general-purpose output, set or clear the corresponding bit in the Px register (x = 1 to 4). To use a pin for general-purpose input, set the bit in the Px register. This turns off the output FET drive.

To configure a pin for its alternate function, set the bit in the Px register. When the latch is set, the "alternate output function" signal controls the output level (see Figure 1). The operation of Ports is discussed further in "quasi-Bidirectional Port Operation" paragraph.

Figure 1. Ports Structure



Note: The internal pull-up can be disabled on P1 when analog function is selected.

Read-Modify-Write Instructions

Some instructions read the latch data rather than the pin data. The latch based instructions read the data, modify the data and then rewrite the latch. These are called "Read-Modify-Write" instructions. Below is a complete list of these special instructions (see Table). When the destination operand is a Port or a Port bit, these instructions read the latch rather than the pin:

Table 2. Read-Modify-Write Instructions

Instruction	Description	Example
ANL	logical AND	ANL P1, A
ORL	logical OR	ORL P2, A
XRL	logical EX-OR	XRL P3, A
JBC	jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	complement bit	CPL P3.0
INC	increment	INC P2
DEC	decrement	DEC P2
DJNZ	decrement and jump if not zero	DJNZ P3, LABEL
MOV Px.y, C	move carry bit to bit y of Port x	MOV P1.5, C
CLR Px.y	clear bit y of Port x	CLR P2.4
SET Px.y	set bit y of Port x	SET P3.3

It is not obvious the last three instructions in this list are Read-Modify-Write instructions. These instructions read the port (all 8 bits), modify the specifically addressed bit and write the new byte back to the latch. These Read-Modify-Write instructions are directed to the latch rather than the pin in order to avoid possible misinterpretation of voltage (and therefore, logic) levels at the pin. For example, a Port bit used to drive the base of an external bipolar transistor can not rise above the transistor's base-emitter junction voltage (a value lower than VIL). With a logic one written to the bit, attempts by the CPU to read the Port at the pin are misinterpreted as logic zero. A read of the latch rather than the pins returns the correct logic-one value.

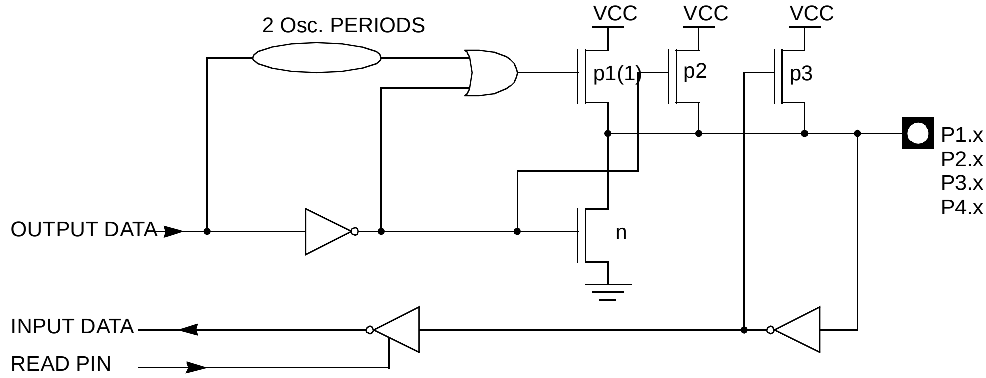
Quasi-bidirectional Port Operation

Port 1, Port 3 and Port 4 have fixed internal pull-ups and are referred to as "quasi-bidirectional" Ports. When configured as an input, the pin impedance appears as logic one and sources current in response to an external logic zero condition. Resets write logic one to all Port latches. If logical zero is subsequently written to a Port latch, it can be returned to input conditions by a logical one written to the latch.

Note: Port latch values change near the end of Read-Modify-Write insruction cycles. Output buffers (and therefore the pin state) update early in the instruction after Read-Modify-Write instruction cycle.

Logical zero-to-one transitions in Port 1, Port 3 and Port 4 use an additional pull-up (p1) to aid this logic transition see Figure 2. This increases switch speed. This extra pull-up sources 100 times normal internal circuit current during 2 oscillator clock periods. The internal pull-ups are field-effect transistors rather than linear resistors. Pull-ups consist of three p-channel FET (pFET) devices. A pFET is on when the gate senses logical zero and off when the gate senses logical one. pFET #1 is turned on for two oscillator periods immediately after a zero-to-one transition in the Port latch. A logical one at the Port pin turns on pFET #3 (a weak pull-up) through the inverter. This inverter and pFET pair form a latch to drive logical one. pFET #2 is a very weak pull-up switched on whenever the associated nFET is switched off. This is traditional CMOS switch convention. Current strengths are 1/10 that of pFET #3

Figure 2. Internal Pull-Up Configurations



SFR Mapping

The Special Function Registers (SFRs) of the T89C5115 fall into the following categories:

Table 3. 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 LSB of DPTR	–	–	–	–	–	–	–	–
DPH	83h	Data Pointer High byte MSB of DPTR	–	–	–	–	–	–	–	–

Table 4. I/O Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
P1	90h	Port 1	–	–	–	–	–	–	–	–
P2	A0h	Port 2 (x2)	–	–	–	–	–	–	–	–
P3	B0h	Port 3	–	–	–	–	–	–	–	–
P4	C0h	Port 4 (x2)	–	–	–	–	–	–	–	–

Table 5. Timers SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
TH0	8Ch	Timer/Counter 0 High byte	–	–	–	–	–	–	–	–
TL0	8Ah	Timer/Counter 0 Low byte	–	–	–	–	–	–	–	–
TH1	8Dh	Timer/Counter 1 High byte	–	–	–	–	–	–	–	–
TL1	8Bh	Timer/Counter 1 Low byte	–	–	–	–	–	–	–	–
TH2	CDh	Timer/Counter 2 High byte	–	–	–	–	–	–	–	–
TL2	CCh	Timer/Counter 2 Low byte	–	–	–	–	–	–	–	–
TCON	88h	Timer/Counter 0 and 1 control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89h	Timer/Counter 0 and 1 Modes	GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Table 5. Timers SFRs (Continued)

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
T2CON	C8h	Timer/Counter 2 control	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#
T2MOD	C9h	Timer/Counter 2 Mode	–	–	–	–	–	–	T2OE	DCEN
RCAP2H	CBh	Timer/Counter 2 Reload/Capture High byte	–	–	–	–	–	–	–	–
RCAP2L	CAh	Timer/Counter 2 Reload/Capture Low byte	–	–	–	–	–	–	–	–
WDTRST	A6h	WatchDog Timer Reset	–	–	–	–	–	–	–	–
WDTPRG	A7h	WatchDog Timer Program	–	–	–	–	–	S2	S1	S0

Table 6. 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	–	–	–	–	–	–	–	–

Table 7. PCA SFRs

Mnemo -nic	Add	Name	7	6	5	4	3	2	1	0
CCON	D8h	PCA Timer/Counter Control	CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0
CMOD	D9h	PCA Timer/Counter Mode	CIDL	WDTE	–	–	–	CPS1	CPS0	ECF
CL	E9h	PCA Timer/Counter Low byte	–	–	–	–	–	–	–	–
CH	F9h	PCA Timer/Counter High byte	–	–	–	–	–	–	–	–
CCAPM0 CCAPM1	DAh DBh	PCA Timer/Counter Mode 0 PCA Timer/Counter Mode 1	–	ECOM0 ECOM1	CAPP0 CAPP1	CAPN0 CAPN1	MAT0 MAT1	TOG0 TOG1	PWM0 PWM1	ECCF0 ECCF1
CCAP0H CCAP1H	FAh FBh	PCA Compare Capture Module 0 H PCA Compare Capture Module 1 H	CCAP0H7 CCAP1H7	CCAP0H6 CCAP1H6	CCAP0H5 CCAP1H5	CCAP0H4 CCAP1H4	CCAP0H3 CCAP1H3	CCAP0H2 CCAP1H2	CCAP0H1 CCAP1H1	CCAP0H0 CCAP1H0
CCAP0L CCAP1L	EAh EBh	PCA Compare Capture Module 0 L PCA Compare Capture Module 1 L	CCAP0L7 CCAP1L7	CCAP0L6 CCAP1L6	CCAP0L5 CCAP1L5	CCAP0L4 CCAP1L4	CCAP0L3 CCAP1L3	CCAP0L2 CCAP1L2	CCAP0L1 CCAP1L1	CCAP0L0 CCAP1L0

Table 8. Interrupt SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
IEN0	A8h	Interrupt Enable Control 0	EA	EC	ET2	ES	ET1	EX1	ET0	EX0
IEN1	E8h	Interrupt Enable Control 1	–	–	–	–	–	–	EADC	–
IPL0	B8h	Interrupt Priority Control Low 0	–	PPC	PT2	PS	PT1	PX1	PT0	PX0
IPH0	B7h	Interrupt Priority Control High 0	–	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
IPL1	F8h	Interrupt Priority Control Low 1	–	–	–	–	–	–	PADCL	–
IPH1	F7h	Interrupt Priority Control High1	–	–	–	–	–	–	PADCH	–

Table 9. ADC SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
ADCON	F3h	ADC Control	–	PSIDLE	ADEN	ADEOC	ADSST	SCH2	SCH1	SCH0
ADCF	F6h	ADC Configuration	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
ADCLK	F2h	ADC Clock	–	–	–	PRS4	PRS3	PRS2	PRS1	PRS0
ADDH	F5h	ADC Data High byte	ADAT9	ADAT8	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2
ADDL	F4h	ADC Data Low byte	–	–	–	–	–	–	ADAT1	ADAT0

Table 10. Other SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
PCON	87h	Power Control	SMOD1	SMOD0	–	POF	GF1	GF0	PD	IDL
AUXR1	A2h	Auxiliary Register 1	–	–	ENBOOT	–	GF3	0	–	DPS
CKCON	8Fh	Clock Control	–	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2
FCON	D1h	Flash Control	FPL3	FPL2	FPL1	FPL0	FPS	FMOD1	FMOD0	FBUSY
EECON	D2h	EEPROM Contol	EEPL3	EEPL2	EEPL1	EEPL0	–	–	EEE	EEBUSY

Table 11. SFR Mapping

	0/8[1]	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	**IPL1** **xxxx x000**	CH 0000 0000	CCAP0H 0000 0000	CCAP1H 0000 0000					FFh
F0h	**B** **0000 0000**		ADCLK xxx0 0000	ADCON x000 0000	ADDL 0000 0000	ADDH 0000 0000	ADCF 0000 0000	IPH1 xxxx x000	F7h
E8h	**IEN1** **xxxx x000**	CL 0000 0000	CCAP0L 0000 0000	CCAP1L 0000 0000					EFh
E0h	**ACC** **0000 0000**								E7h
D8h	**CCON** **0000 0000**	CMOD 00xx x000	CCAPM0 x000 0000	CCAPM1 x000 0000					DFh
D0h	**PSW** **0000 0000**	FCON 0000 0000	EECON xxxx xx00						D7h
C8h	**T2CON** **0000 0000**	T2MOD xxxx xx00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000			CFh
C0h	**P4** **xxxx xx11**								C7h
B8h	**IPL0** **x000 0000**	SADEN 0000 0000							BFh
B0h	**P3** **1111 1111**							IPH0 x000 0000	B7h
A8h	**IEN0** **0000 0000**	SADDR 0000 0000							AFh
A0h	**P2** **1111 1111**		AUXR1 xxxx 00x0				WDTRST 1111 1111	WDTPRG xxxx x000	A7h
98h	**SCON** **0000 0000**	SBUF 0000 0000							9Fh
90h	**P1** **1111 1111**								97h
88h	**TCON** **0000 0000**	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000		CKCON 0000 0000	8Fh
80h	**P0** **1111 1111**	SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 00x1 0000	87h
	0/8[1]	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Reserved

Note: 1. These registers are bit-addressable.
Sixteen addresses in the SFR space are both byte-addressable and bit-addressable. The bit-addressable SFR's are those whose address ends in 0 and 8. The bit addresses, in this area, are 0x80 through to 0xFF.

Clock

The T89C5115 core needs only 6 clock periods per machine cycle. This feature, called 'X2', provides the following advantages:

- Divides frequency crystals by 2 (cheaper crystals) while keeping the same CPU power.
- Saves power consumption while keeping the same CPU power (oscillator power saving).
- Saves power consumption by dividing dynamic operating frequency by 2 in operating and idle modes.
- Increases CPU power by 2 while keeping the same crystal frequency.

In order to keep the original C51 compatibility, a divider-by-2 is inserted between the XTAL1 signal and the main clock input of the core (phase generator). This divider may be disabled by the software.

An extra feature is available to start after Reset in the X2 mode. This feature can be enabled by a bit X2B in the Hardware Security Byte. This bit is described in the section "In-System Programming".

Description

The X2 bit in the CKCON register (see Table 12) allows switching from 12 clock cycles per instruction to 6 clock cycles and vice versa. At reset, the standard speed is activated (STD mode).

Setting this bit activates the X2 feature (X2 mode) for the CPU Clock only (see Figure 3.).

The Timers 0, 1 and 2, Uart, PCA or WatchDog switch in X2 mode only if the corresponding bit is cleared in the CKCON register.

The clock for the whole circuit and peripheral is first divided by two before being used by the CPU core and peripherals. This allows any cyclic ratio to be accepted on the XTAL1 input. In X2 mode, as this divider is bypassed, the signals on XTAL1 must have a cyclic ratio between 40 to 60%. Figure 3. shows the clock generation block diagram. The X2 bit is validated on the XTAL1÷2 rising edge to avoid glitches when switching from the X2 to the STD mode. Figure 4 shows the mode switching waveforms.

Figure 3. Clock CPU Generation Diagram

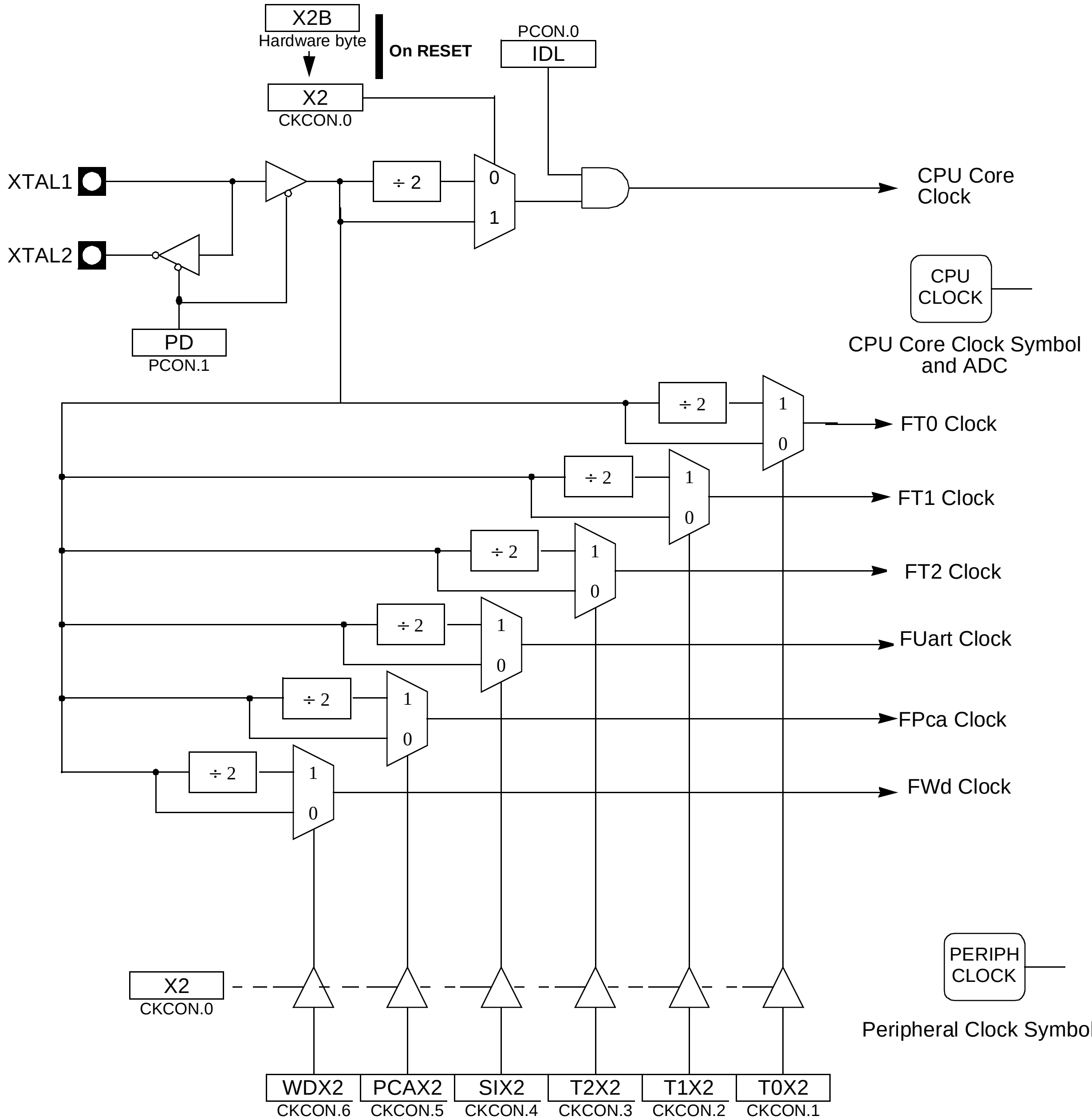
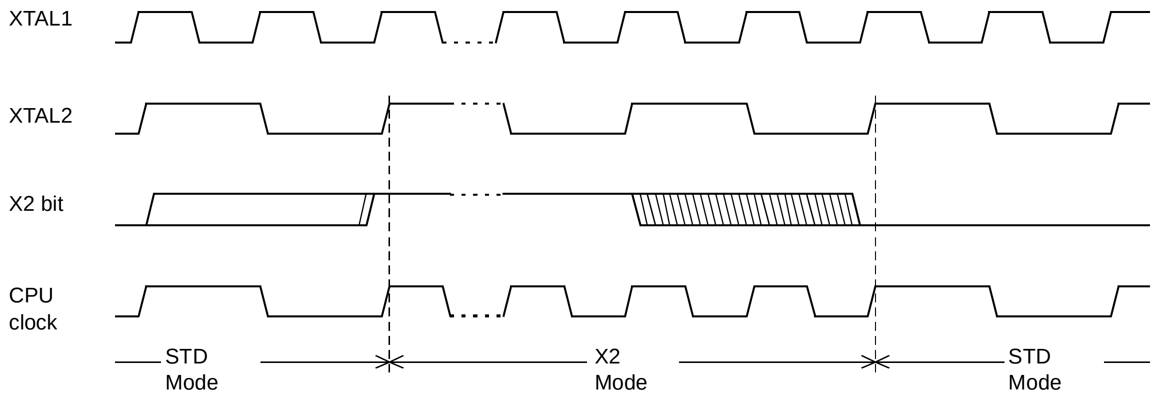


Figure 4. Mode Switching Waveforms



Note: In order to prevent any incorrect operation while operating in the X2 mode, users must be aware that all peripherals using the clock frequency as a time reference (UART, 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. A UART with a 4800 baud rate will have a 9600 baud rate.

Register

Table 12. CKCON Register

CKCON (S:8Fh)
Clock Control Register

7	6	5	4	3	2	1	0
–	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	WDX2	**WatchDog clock**[(1)] Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
5	PCAX2	**Programmable Counter Array clock**[(1)] Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
4	SIX2	**Enhanced UART clock (MODE 0 and 2)**[(1)] Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
3	T2X2	**Timer2 clock**[(1)] Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
2	T1X2	**Timer1 clock**[(1)] Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
1	T0X2	**Timer0 clock**[(1)] Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.
0	X2	**CPU clock** Clear to select 12 clock periods per machine cycle (STD mode) for CPU and all the peripherals. Set to select 6 clock periods per machine cycle (X2 mode) and to enable the individual peripherals "X2"bits.

Notes: 1. This control bit is validated when the CPU clock bit X2 is set; when X2 is low, this bit has no effect.

Reset Value = x000 0000b

Power Management

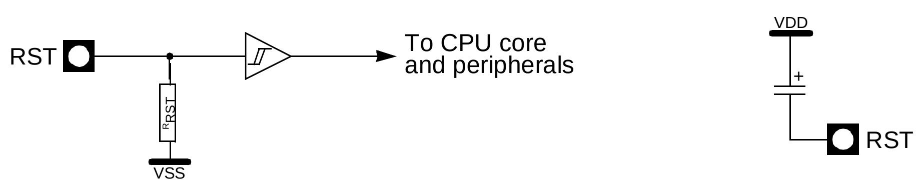
Introduction

Two power reduction modes are implemented in the T89C5115: 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 “Clock”.

Reset

A reset is required after applying power at turn-on. 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 and stabilized and VCC established within the specified operating ranges. A device reset initializes the T89C5115 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 5. Resistor value and input characteristics are discussed in the Section “DC Characteristics” of the T89C5115 datasheet. The status of the Port pins during reset is detailed in Table 13.

Figure 5. Reset Circuitry and Power-On Reset



a. RST input circuitry

b. Power-on Reset

Table 13. Pin Conditions in Special Operating Modes

Mode	Port 1	Port 2	Port 3	Port 4
Reset	High	High	High	High
Idle	Data	Data	Data	Data
Power-down	Data	Data	Data	Data

Reset Recommendation to Prevent Flash Corruption

A bad reset sequence will lead to bad microcontroller initialization and system registers like SFR’s, Program Counter, etc. will not be correctly initialized. A bad initialization may lead to unpredictable behaviour of the C51 microcontroller.

An example of this 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 inorder to unmap the boot from the code area (yet due to a bad reset) the bit ENBOOT in SFR’s 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).

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. 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 13.

Entering Idle Mode

To enter Idle mode, set the IDL bit in PCON register (see Table 14). The T89C5115 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 T89C5115 enters Power-down mode. Then it does not go in Idle mode when exiting Power-down mode.

Exiting Idle Mode

There are two 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 T89C5115 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.

Power-down Mode

The Power-down mode places the T89C5115 in a very low power state. Power-down mode stops the oscillator, freezes all clock at known states. 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 13.

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

Entering Power-down Mode

To enter Power-down mode, set PD bit in PCON register. The T89C5115 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.

Exiting Power-down Mode

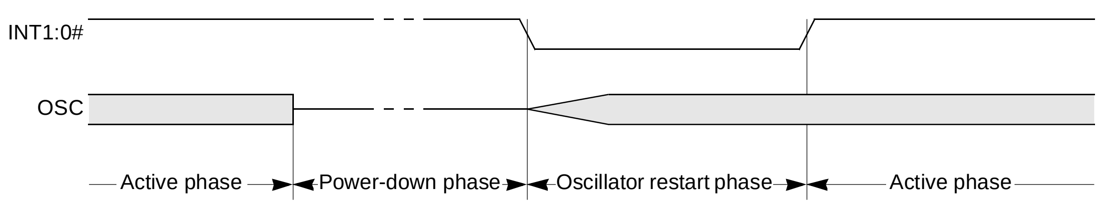
Note: If VDD was reduced during the Power-down mode, do not exit Power-down mode until VDD is restored to the normal operating level.

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

1. Generate an enabled external interrupt.
   – The T89C5115 provides capability to exit from Power-down using INT0#, INT1#.
   Hardware clears PD bit in PCON register which starts the oscillator and restores the clocks to the CPU and peripherals. Using INTx# input, execution resumes when the input is released (see Figure 6). 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.

Notes: 1. The external interrupt used to exit Power-down mode must be configured as level sensitive (INT0# and 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 6. Power-down Exit Waveform Using INT1:0#



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 T89C5115 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.

Registers

PCON (S:87h)

Table 14. PCON Register

Power Configuration Register

7	6	5	4	3	2	1	0
–	–	–	–	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7-4	–	**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 = XXXX 0000b

Data Memory

The T89C5115 provides data memory access in two different spaces:

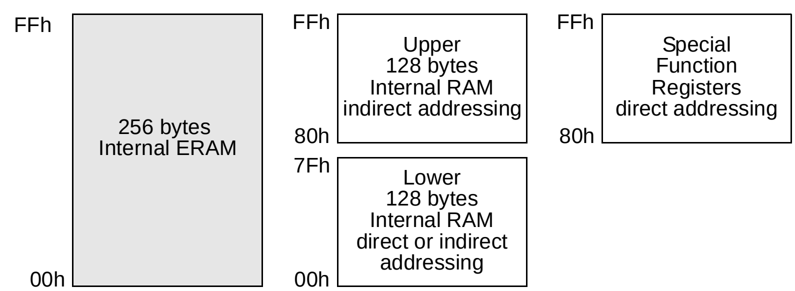
The internal space mapped in three separate segments:

- the lower 128 bytes RAM segment.
- the upper 128 bytes RAM segment.
- the expanded 256 bytes RAM segment (ERAM).

A fourth internal segment is available but dedicated to Special Function Registers, SFRs, (addresses 80h to FFh) accessible by direct addressing mode.

Figure 7 shows the internal data memory spaces organization.

Figure 7. Internal Memory – RAM



Internal Space

Lower 128 Bytes RAM

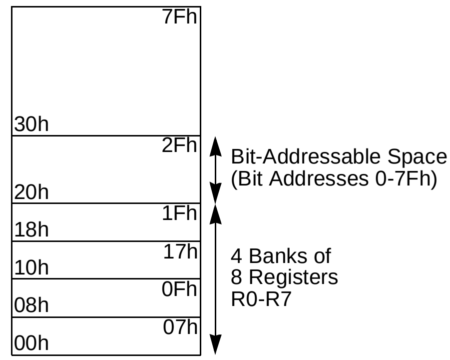
The lower 128 bytes of RAM (see Figure 7) are accessible from address 00h to 7Fh using direct or indirect addressing modes. The lowest 32 bytes are grouped into 4 banks of 8 registers (R0 to R7). Two bits RS0 and RS1 in PSW register (see Figure 16) select which bank is in use according to Table . This 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 15. Register Bank Selection

RS1	RS0	Description
0	0	Register bank 0 from 00h to 07h
0	1	Register bank 0 from 08h to 0Fh
1	0	Register bank 0 from 10h to 17h
1	1	Register bank 0 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. Lower 128 bytes Internal RAM Organization



Upper 128 Bytes RAM

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

Expanded RAM

The on-chip 256 bytes of expanded RAM (ERAM) are accessible from address 0000h to 00FFh using indirect addressing mode through MOVX instructions. In this address range.

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.

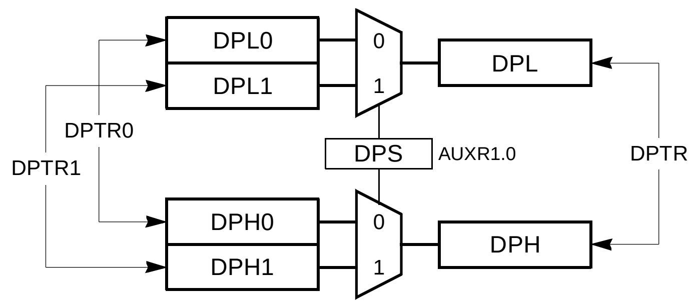
Dual Data Pointer

Description

The T89C5115 implements 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 Figure 17) is used to select whether DPTR is the data pointer 0 or the data pointer 1 (see Figure 9).

Figure 9. Dual Data Pointer Implementation



Application

Software can take advantage of the additional data pointers to both increase speed and reduce code size, for example, block operations (copy, compare…) are well served by using one data pointer as a “source” pointer and the other one as a “destination” pointer. Hereafter is an example of block move implementation using the two pointers and coded in assembler. The latest C compiler takes also advantage of this feature by providing enhanced algorithm libraries.

The INC instruction is a short (2 bytes) and fast (6 machine cycle) way to manipulate the DPS bit in the AUXR1 register. 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.

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

AUXR1EQU0A2h

move:movDPTR,#SOURCE ; address of SOURCE
  incAUXR1 ; switch data pointers
  movDPTR,#DEST ; address of DEST
mv_loop:incAUXR1; switch data pointers
  movxA,@DPTR; get a byte from SOURCE
  incDPTR; increment SOURCE address
  incAUXR1; switch data pointers
  movx@DPTR,A; write the byte to DEST
  incDPTR; increment DEST address
  jnzmv_loop; check for NULL terminator
end_move:
```

Registers

Table 16. 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 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 17. AUXR1 Register

AUXR1 (S:A2h)
Auxiliary Control Register 1

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

Bit Number	Bit Mnemonic	Description
7-6	–	**Reserved** The value read from these bits is indeterminate. Do not set these bits.
5	ENBOOT	**Enable Boot Flash** Set this bit for map the boot flash between F800h -FFFFh Clear this bit for disable boot flash.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	GF3	**General-purpose Flag 3.**
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 dual data pointer: DPTR1. Clear to select first dual data pointer: DPTR0.

Reset Value = xxxx 00x0b

EEPROM Data Memory

The 2-kbyte on-chip EEPROM memory block is located at addresses 0000h to 07FFh of the XRAM/ERAM memory space and is selected by setting control bits in the EECON register. A read in the EEPROM memory is done with a MOVX instruction.

A physical write in the EEPROM memory is done in two steps: write data in the column latches and transfer of all data latches into an EEPROM memory row (programming).

The number of data written on the page may vary from 1 up to 128 bytes (the page size). When programming, only the data written in the column latch is programmed and a ninth bit is used to obtain this feature. This provides the capability to program the whole memory by bytes, by page or by a number of bytes in a page. Indeed, each ninth bit is set when the writing the corresponding byte in a row and all these ninth bits are reset after the writing of the complete EEPROM row.

Write Data in the Column Latches

Data is written by byte to the column latches as for an external RAM memory. Out of the 11 address bits of the data pointer, the 4 MSBs are used for page selection (row) and 7 are used for byte selection. Between two EEPROM programming sessions, all the addresses in the column latches must stay on the same page, meaning that the 4 MSB must no be changed.

The following procedure is used to write to the column latches:

- Save and disable interrupt.
- Set bit EEE of EECON register
- Load DPTR with the address to write
- Store A register with the data to be written
- Execute a MOVX @DPTR, A
- If needed loop the three last instructions until the end of a 128 bytes page
- Restore interrupt.

Note: The last page address used when loading the column latch is the one used to select the page programming address.

Programming

The EEPROM programming consists of the following actions:

- writing one or more bytes of one page in the column latches. Normally, all bytes must belong to the same page; if not, the first page address will be latched and the others discarded.
- launching programming by writing the control sequence (50h followed by A0h) to the EECON register.
- EEBUSY flag in EECON is then set by hardware to indicate that programming is in progress and that the EEPROM segment is not available for reading.
- The end of programming is indicated by a hardware clear of the EEBUSY flag.

Note: The sequence 5xh and Axh must be executed without instructions between them, otherwise the programming is aborted.

Read Data

The following procedure is used to read the data stored in the EEPROM memory:

- Save and disable interrupt
- Set bit EEE of EECON register
- Load DPTR with the address to read
- Execute a MOVX A, @DPTR
- Restore interrupt

Examples

```
;*F***************************************************************************
;* NAME: api_rd_eeprom_byte
;* DPTR contain address to read.
;* Acc contain the reading value
;* NOTE: before execute this function, be sure the EEPROM is not BUSY
;*****************************************************************************
api_rd_eeprom_byte:
 MOV   EECON, #02h; map EEPROM in XRAM space
 MOVX A, @DPTR
 MOV   EECON, #00h; unmap EEPROM
ret

;*F***************************************************************************
;* NAME: api_ld_eeprom_cl
;* DPTR contain address to load
;* Acc contain value to load
;* NOTE: in this example we load only 1 byte, but it is possible upto
;* 128 bytes.
;* before execute this function, be sure the EEPROM is not BUSY
;*****************************************************************************
api_ld_eeprom_cl:
 MOV   EECON, #02h ; map EEPROM in XRAM space
 MOVX @DPTR, A
 MOVEECON, #00h; unmap EEPROM
ret

;*F***************************************************************************
;* NAME: api_wr_eeprom
;* NOTE: before execute this function, be sure the EEPROM is not BUSY
;*****************************************************************************
api_wr_eeprom:
 MOV   EECON, #050h
 MOV   EECON, #0A0h
ret
```

Registers

Table 18. EECON Register

EECON (S:0D2h)
EEPROM Control Register

7	6	5	4	3	2	1	0
EEPL3	EEPL2	EEPL1	EEPL0	-	-	EEE	EEBUSY

Bit Number	Bit Mnemonic	Description
7-4	EEPL3-0	**Programming Launch Command bits** Write 5Xh followed by AXh to EEPL to launch the programming.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	EEE	**Enable EEPROM Space bit** Set to map the EEPROM space during MOVX instructions (Write in the column latches). Clear to map the XRAM space during MOVX.
0	EEBUSY	**Programming Busy flag** Set by hardware when programming is in progress. Cleared by hardware when programming is done. Can not be set or cleared by software.

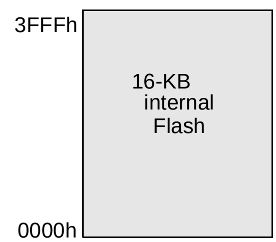
Reset Value = XXXX XX00b
Not bit addressable

Program/Code Memory

The T89C5115 implement 16-KB of on-chip program/code memory.

The Flash memory increases EPROM and ROM functionality by 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 VDD voltage. Thus, the Flash Memory can be programmed using only one voltage and allows In-System Programming commonly known as ISP. Hardware programming mode is also available using specific programming tool.

Figure 10. Program/Code Memory Organization



Flash Memory Architecture

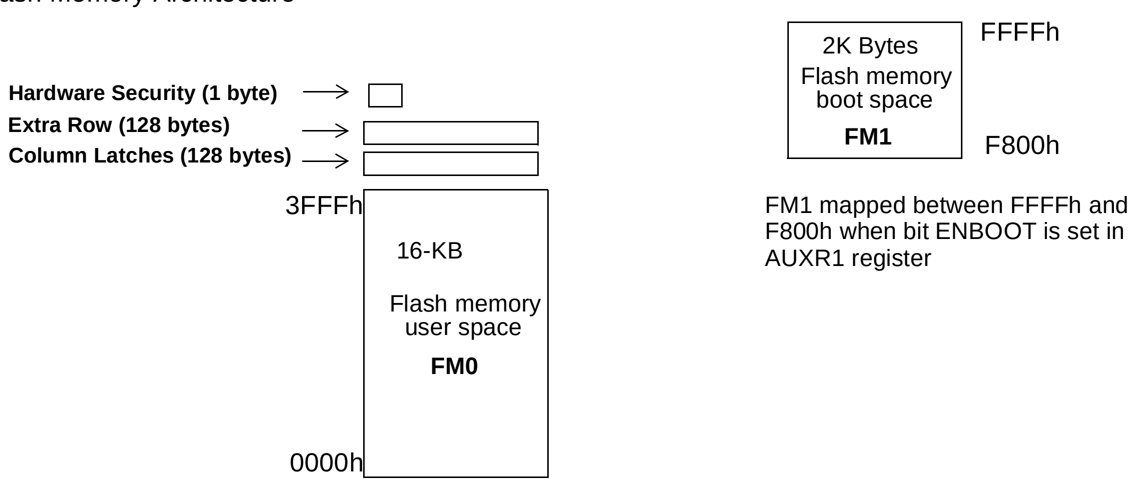
T89C5115 features two on-chip flash memories:

- Flash memory FM0:
  containing 16-KB of program memory (user space) organized into pages 128 bytes
- Flash memory FM1:
  2K Bytes for boot loader and Application Programming Interfaces (API).

The FM0 can be program by both parallel programming and Serial In-System Programming (ISP) whereas FM1 supports only parallel programming by programmers. The ISP mode is detailed in the "In-System Programming" section.

All Read/Write access operations on Flash Memory by user application are managed by a set of API described in the "In-System Programming" section.

Figure 11. Flash Memory Architecture



FM0 Memory Architecture

The Flash memory is made up of 4 blocks (see Figure 11):

1. The memory array (user space) 16-KB.
2. The Extra Row.
3. The Hardware security bits.
4. The column latch registers.

User Space

This space is composed of a 16-KB Flash memory organized in 128 pages of 128 bytes. It contains the user's application code.

Extra Row (XROW)

This row is a part of FM0 and has a size of 128 bytes. The extra row may contain information for boot loader usage.

Hardware security Byte

The Hardware Security Byte space is a part of FM0 and has a size of 1 byte.
The 4 MSB can be read/written by software, the 4 LSB can only be read by software and written by hardware in parallel mode.

Column Latches

The column latches, also part of FM0, have a size of full page (128 bytes).
The column latches are the entrance buffers of the three previous memory locations (user array, XROW and Hardware security byte).

Cross Flash Memory Access Description

The FM0 memory can be program only from FM1. Programming FM0 from FM0 or from external memory is impossible.

The FM1 memory can be program only by parallel programming.

The Table 19 show all software flash access allowed.

Table 19. Cross Flash Memory Access

Code executing from		Action	**FM0 (user Flash)**	FM1 (boot Flash)
	FM0 (user Flash)	Read	ok	-
		Load column latch	ok	-
		Write	-	-
	FM1 (boot flash)	Read	ok	ok
		Load column latch	ok	-
		Write	ok	-

Overview of FM0 Operations

The CPU interfaces to the Flash memory through the FCON register and AUXR1 register.

These registers are used to:

- Map the memory spaces in the adressable space
- Launch the programming of the memory spaces
- Get the status of the flash memory (busy/not busy)

Mapping of the Memory Space

By default, the user space is accessed by MOVC instruction for read only. The column latches space is made accessible by setting the FPS bit in FCON register. Writing is possible from 0000h to 3FFFh, address bits 6 to 0 are used to select an address within a page while bits 14 to 7 are used to select the programming address of the page.
Setting FPS bit takes precedence on the EEE bit in EECON register.

The other memory spaces (user, extra row, hardware security) are made accessible in the code segment by programming bits FMOD0 and FMOD1 in FCON register in accordance with Table 20. A MOVC instruction is then used for reading these spaces.

Table 20. FM0 Blocks Select Bits

FMOD1	FMOD0	FM0 Adressable space
0	0	User (0000h-3FFFh)
0	1	Extra Row(FF80h-FFFFh)
1	0	Hardware Security Byte (0000h)
1	1	reserved

Launching Programming

FPL3:0 bits in FCON register are used to secure the launch of programming. A specific sequence must be written in these bits to unlock the write protection and to launch the programming. This sequence is 5xh followed by Axh. Table 21 summarizes the memory spaces to program according to FMOD1:0 bits.

Table 21. Programming Spaces

	Write to FCON				
	FPL3:0	FPS	FMOD1	FMOD0	Operation
User	5	X	0	0	No action
	A	X	0	0	Write the column latches in user space
Extra Row	5	X	0	1	No action
	A	X	0	1	Write the column latches in extra row space
Hardware Security Byte	5	X	1	0	No action
	A	X	1	0	Write the fuse bits space
Reserved	5	X	1	1	No action
	A	X	1	1	No action

Note: The sequence 5xh and Axh must be executing without instructions between them otherwise the programming is aborted.

Interrupts that may occur during programming time must be disabled to avoid any spurious exit of the programming mode.

Status of the Flash Memory

The bit FBUSY in FCON register is used to indicate the status of programming.

FBUSY is set when programming is in progress.

Selecting FM1

The bit ENBOOT in AUXR1 register is used to map FM1 from F800h to FFFFh.

Loading the Column Latches

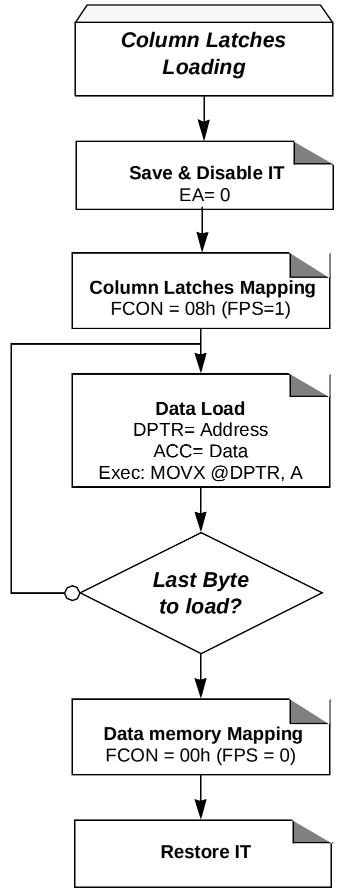
Any number of data from 1 byte to 128 bytes can be loaded in the column latches. This provides the capability to program the whole memory by byte, by page or by any number of bytes in a page.

When programming is launched, an automatic erase of the locations loaded in the column latches is first performed, then programming is effectively done. Thus no page or block erase is needed and only the loaded data are programmed in the corresponding page.

The following procedure is used to load the column latches and is summarized in Figure 12:

- Disable interrupt and map the column latch space by setting FPS bit.
- Load the DPTR with the address to load.
- Load Accumulator register with the data to load.
- Execute the MOVX @DPTR, A instruction.
- If needed loop the three last instructions until the page is completely loaded.
- unmap the column latch and Enable Interrupt

Figure 12. Column Latches Loading Procedure



Note: The last page address used when loading the column latch is the one used to select the page programming address.

Programming the Flash Spaces

User

The following procedure is used to program the User space and is summarized in Figure 13:

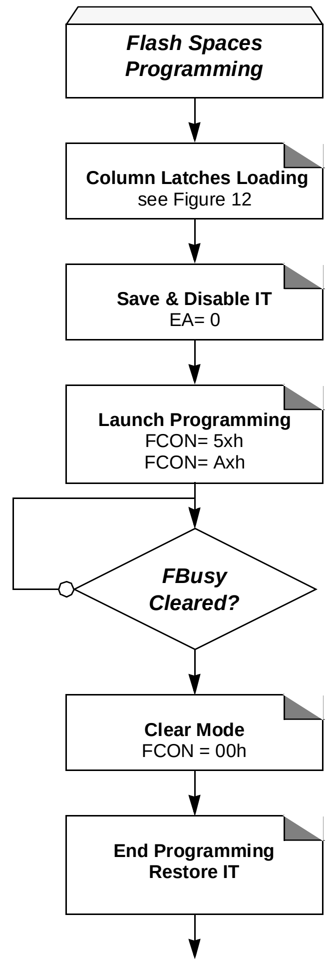
- Load up to one page of data in the column latches from address 0000h to 3FFFh.
- Disable the interrupts.
- Launch the programming by writing the data sequence 50h followed by A0h in FCON register (only from FM1).
  The end of the programming indicated by the FBUSY flag cleared.
- Enable the interrupts.

Extra Row

The following procedure is used to program the Extra Row space and is summarized in Figure 13:

- Load data in the column latches from address FF80h to FFFFh.
- Disable the interrupts.
- Launch the programming by writing the data sequence 52h followed by A2h in FCON register (only from FM1).
  The end of the programming indicated by the FBUSY flag cleared.
- Enable the interrupts.

Figure 13. Flash and Extra row Programming Procedure

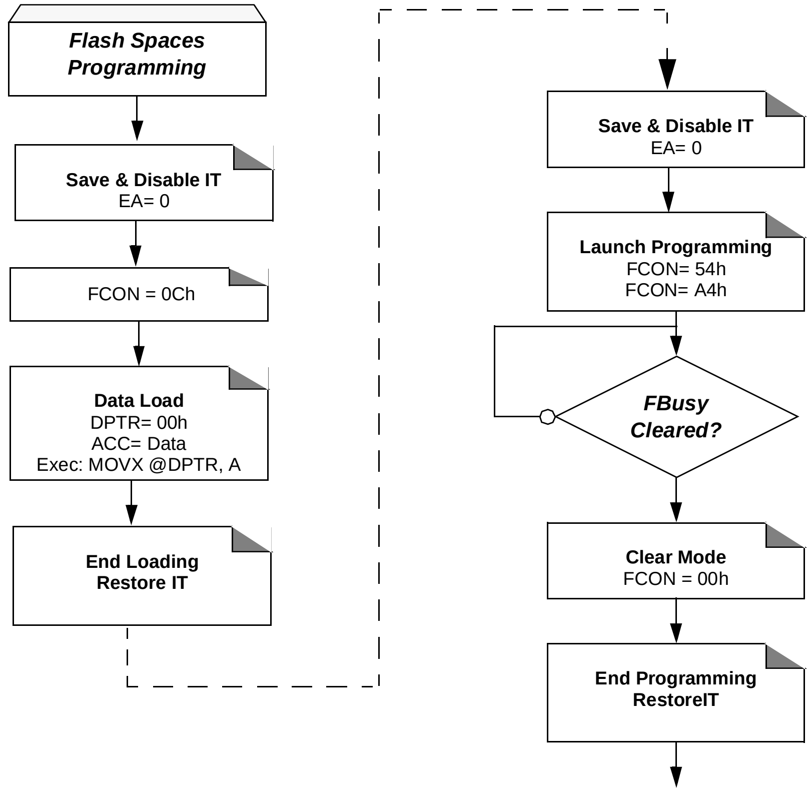


Hardware Security Byte

The following procedure is used to program the Hardware Security Byte space and is summarized in Figure 14:

- Set FPS and map Hardware byte (FCON = 0x0C)
- Save and disable the interrupts.
- Load DPTR at address 0000h.
- Load Accumulator register with the data to load.
- Execute the MOVX @DPTR, A instruction.
- Launch the programming by writing the data sequence 54h followed by A4h in FCON register (only from FM1).
  The end of the programming indicated by the FBusy flag cleared.
- Restore the interrupts

Figure 14. Hardware Programming Procedure



Reading the Flash Spaces

User

The following procedure is used to read the User space:

- Read one byte in Accumulator by executing MOVC A,@A+DPTR with A+DPTR=read@.

Note: FCON is supposed to be reset when not needed.

Extra Row

The following procedure is used to read the Extra Row space and is summarized in Figure 15:

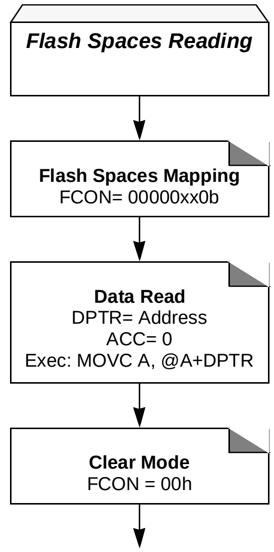
- Map the Extra Row space by writing 02h in FCON register.
- Read one byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= FF80h to FFFFh.
- Clear FCON to unmap the Extra Row.

Hardware Security Byte

The following procedure is used to read the Hardware Security space and is summarized in Figure 15:

- Map the Hardware Security space by writing 04h in FCON register.
- Read the byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= 0000h.
- Clear FCON to unmap the Hardware Security Byte.

Figure 15. Reading Procedure



Flash Protection from Parallel Programming

The three lock bits in Hardware Security Byte (see "In-System Programming" section) are programmed according to Table 22 provide different level of protection for the on-chip code and data located in FM0 and FM1.

The only way to write this bits are the parallel mode. They are set by default to level 3.

Table 22. Program Lock bit

Program Lock Bits				
Security level	LB0	LB1	LB2	**Protection Description**
1	U	U	U	No program lock features enabled. MOVC instruction executed from external program memory returns non encrypted data.
2	P	U	U	Parallel programming of the Flash is disabled.
3	U	P	U	Same as 2, also verify through parallel programming interface is disabled.

Program Lock bits

U: unprogrammed

P: programmed

WARNING: Security level 2 and 3 should only be programmed after Flash and Core verification.

Preventing Flash Corruption

See paragraph in the "Power Management" section, page 17.

Registers

FCON Register
FCON (S:D1h)
Flash Control Register

7	6	5	4	3	2	1	0
FPL3	FPL2	FPL1	FPL0	FPS	FMOD1	FMOD0	FBUSY

Bit Number	Bit Mnemonic	Description
7-4	FPL3:0	**Programming Launch Command Bits** Write 5Xh followed by AXh to launch the programming according to FMOD1:0. (see Table 21.)
3	FPS	**Flash Map Program Space** Set to map the column latch space in the data memory space. Clear to re-map the data memory space.
2-1	FMOD1:0	**Flash Mode** See Table 20 or Table 21.
0	FBUSY	**Flash Busy** Set by hardware when programming is in progress. Clear by hardware when programming is done. Can not be changed by software.

Reset Value = 0000 0000b

In-System Programming (ISP)

With the implementation of the User Space (FM0) and the Boot Space (FM1) in Flash technology the T89C5115 allows the system engineer the development of applications with a very high level of flexibility. This flexibility is based on the possibility to alter the customer program at any stages of a product's life:

- Before assembly the 1st personalization of the product by programming in the FM0 and if needed also a customized Boot loader in the FM1.
  Atmel provide also a standard Boot loader by default UART
- After assembling on the PCB in its final embedded position by serial mode via the UART.

This In-System Programming (ISP) allows code modification over the total lifetime of the product.

Besides the default Boot loader Atmel provide to the customer also all the needed Application-Programming-Interfaces (API) which are needed for the ISP. The API are located also in the Boot memory.

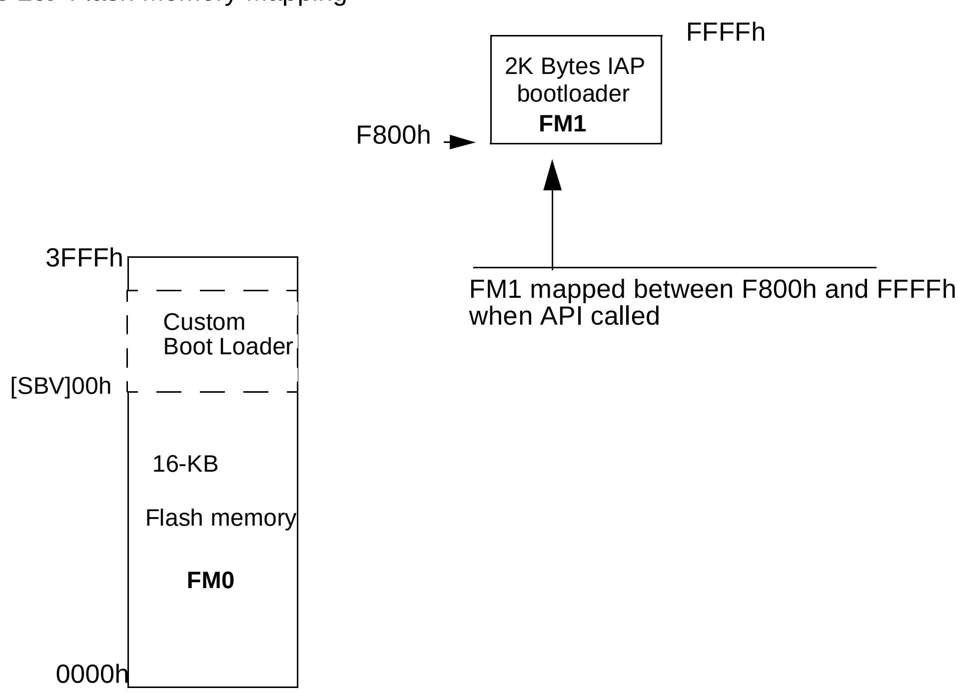
This allow the customer to have a full use of the 16-Kbyte user memory.

Flash Programming and Erasure

There are three methods of programming the Flash memory:

- The Atmel bootloader located in FM1 is activated by the application. Low level API routines (located in FM1)will be used to program FM0. The interface used for serial downloading to FM0 is the UART. API can be called also by user's bootloader located in FM0 at [SBV]00h.
- A further method exist in activating the Atmel boot loader by hardware activation.
- The FM0 can be programmed also by the parallel mode using a programmer.

Figure 16. Flash Memory Mapping



Boot Process

Software Boot Process Example

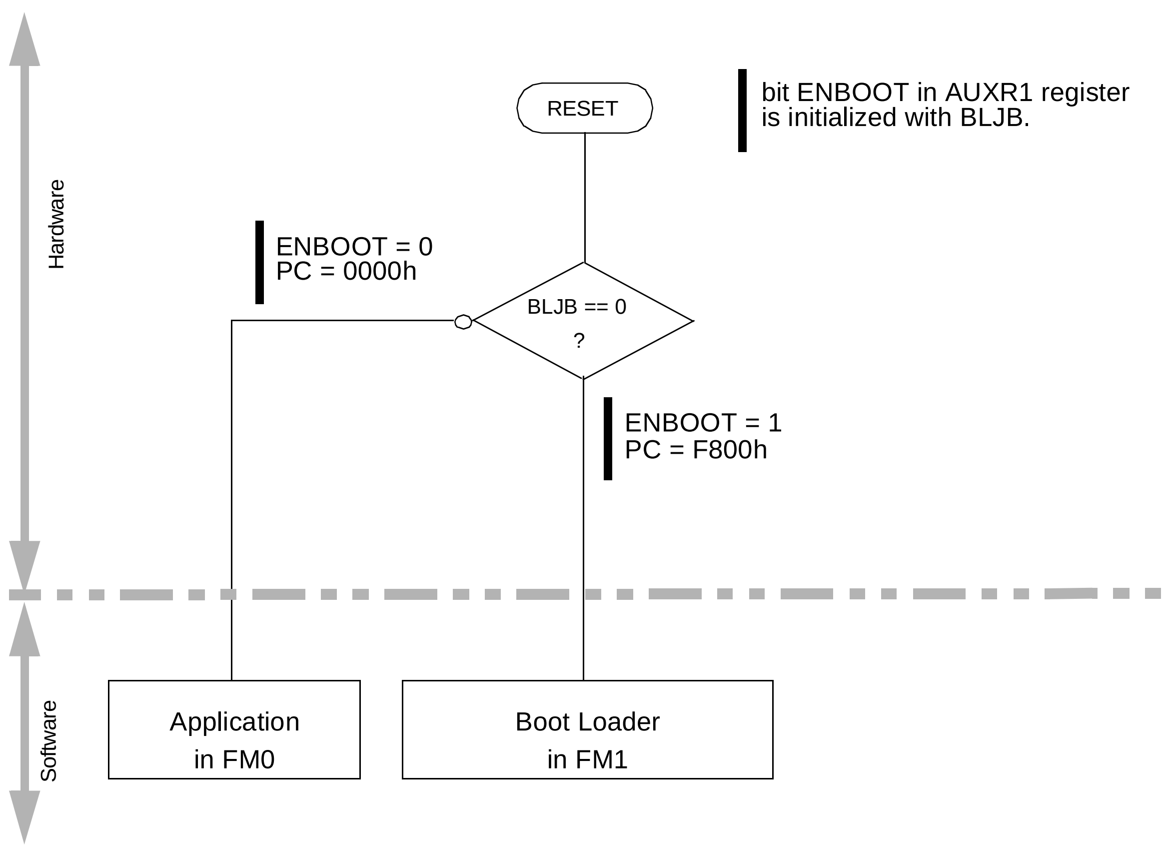
Many algorithms can be used for the software boot process. Before describing them, The description of the different flags and bytes is given below:

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 @F800h on FM1.
- BLJB = 0 on parts delivered with bootloader programmed.
- 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 17. Hardware Boot Process Algorithm



Application Programming Interface

Several Application Program Interface (API) calls are available for use by an application program to permit selective erasing and programming of Flash pages. All calls are made by functions.

All APIs are describe in: "In-System Programing: Flash Library for T89C5115", available on the Atmel web site at www.atmel.com.

Table 23. List of API

API Call	Description
PROGRAM DATA BYTE	Write a byte in flash memory
PROGRAM DATA PAGE	Write a page (128 bytes) in flash memory
PROGRAM EEPROM BYTE	Write a byte in Eeprom memory
ERASE BLOCK	Erase all flash memory
ERASE BOOT VECTOR (SBV)	Erase the boot vector
PROGRAM BOOT VECTOR (SBV)	Write the boot vector
PROGRAM EXTRA BYTE (EB)	Write the extra byte
READ DATA BYTE	–
READ EEPROM BYTE	–
READ FAMILY CODE	–
READ MANUFACTURER CODE	–
READ PRODUCT NAME	–
READ REVISION NUMBER	–
READ STATUS BIT (BSB)	Read the status bit
READ BOOT VECTOR (SBV)	Read the boot vector
READ EXTRA BYTE (EB)	Read the extra byte
PROGRAM X2	Write the hardware flag for X2 mode
READ X2	Read the hardware flag for X2 mode
START BOOTLOADER	To start the bootloader from the application

XROW Bytes

Table 24. XROW Mapping

Mnemonic	Description	Default value	Address
–	Copy of the Manufacturer Code	58h	30h
–	Copy of the Device ID#1: Family code	D7h	31h
–	Copy of the Device ID#2: Memories size and type	BBh	60h
–	Copy of the Device ID#3: Name and Revision	FFh	61h

Hardware Security Byte

Table 25. Hardware Security byte

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

Bit Number	Bit Mnemonic	Description
7	X2B	**X2 Bit** Set this bit to start in standard mode Clear this bit to start in X2 mode.
6	BLJB	**Boot Loader Jump Bit** - 1: To start the user's application on next RESET (@0000h) located in FM0, - 0: To start the boot loader(@F800h) located in FM1.
5-3	-	**Reserved** The value read from these bits are indeterminate.
2-0	LB2:0	**Lock Bits**

Default value after erasing chip: FFh

Notes: 1. Only the 4 MSB bits can be accessed by software.
2. The 4 LSB bits can only be accessed by parallel mode.

Serial I/O Port

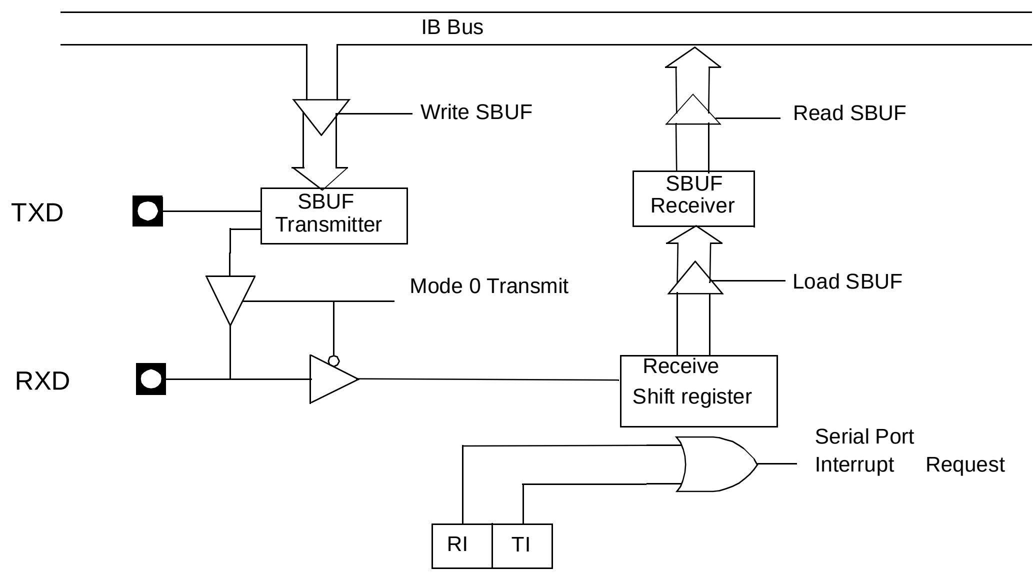
The T89C5115 I/O serial port is compatible with the I/O serial port in the 80C52.

It provides both synchronous and asynchronous communication modes. It operates as a Universal Asynchronous Receiver and Transmitter (UART) in three full-duplex modes (Modes 1, 2 and 3). Asynchronous transmission and reception can occur simultaneously and at different baud rates

Serial I/O port includes the following enhancements:

- Framing error detection
- Automatic address recognition

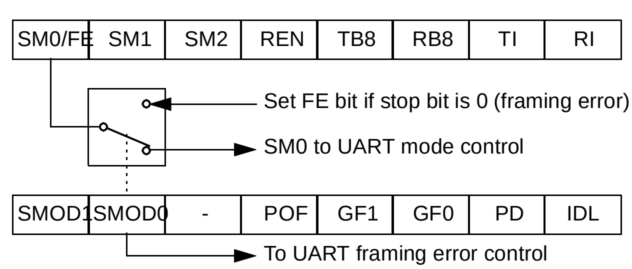
Figure 18. Serial I/O Port Block Diagram



Framing Error Detection

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

Figure 19. Framing Error Block Diagram



When this feature is enabled, the receiver checks each incoming data frame for a valid stop bit. An invalid stop bit may result from noise on the serial lines or from simultaneous transmission by two CPUs. If a valid stop bit is not found, the Framing Error bit (FE) in SCON register bit is set.

The software may examine the FE bit after each reception to check for data errors. Once set, only software or a reset clears the FE bit. Subsequently received frames with valid stop bits cannot clear the FE bit. When the FE feature is enabled, RI rises on the stop bit instead of the last data bit (See Figure 20 and Figure 21).

Figure 20. UART Timing in Mode 1

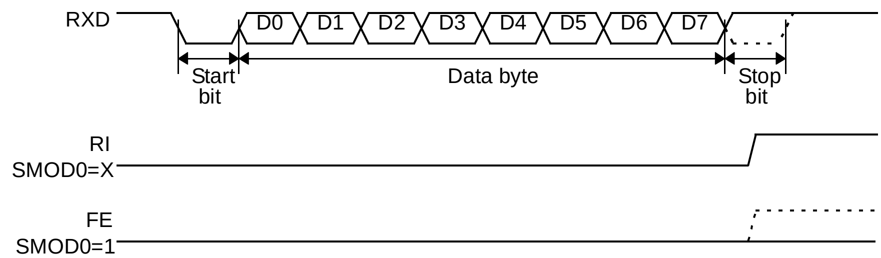
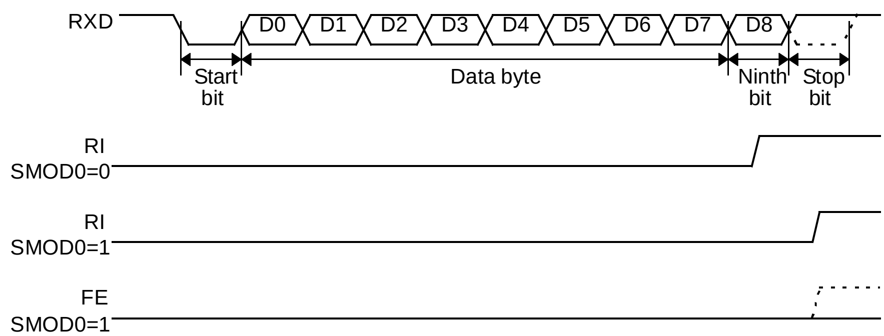


Figure 21. UART Timing in Modes 2 and 3



Automatic Address Recognition

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

Implemented in the 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 will the receiver set the RI bit in the SCON register to generate an interrupt. This ensures that the CPU is not interrupted by command frames addressed to other devices.

If necessary, you can enable the automatic address recognition feature in mode 1. In this configuration, the stop bit takes the place of the ninth data bit. Bit RI is set only when the received command frame address matches the device's address and is terminated by a valid stop bit.

To support automatic address recognition, a device is identified by a given address and a broadcast address.

Note: The multiprocessor communication and automatic address recognition features cannot be enabled in mode 0 (i.e. setting SM2 bit in SCON register in mode 0 has no effect).

Given Address

Each device has an individual address that is specified in the SADDR register; the SADEN register is a mask byte that contains don't-care bits (defined by zeros) to form the device's given address. The don't-care bits provide the flexibility to address one or more slaves at a time. The following example illustrates how a given address is formed. To address a device by its individual address, the SADEN mask byte must be `1111 1111b`.

For example:

```
SADDR0101 0110b
SADEN1111 1100b
Given0101 01XXb
```

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

```
Slave A:SADDR1111 0001b
    SADEN1111 1010b
    Given1111 0X0Xb

Slave B:SADDR1111 0011b
    SADEN1111 1001b
    Given1111 0XX1b

Slave C:SADDR1111 0010b
    SADEN1111 1101b
    Given1111 00X1b
```

The SADEN byte is selected so that each slave may be addressed separately.

For slave A, bit 0 (the LSB) is a don't-care bit; for slaves B and C, bit 0 is a 1. To communicate with slave A only, the master must send an address where bit 0 is clear (e.g. `1111 0000b`).

For slave A, bit 1 is a 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`).

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 OR SADEN1111 111Xb
```

The use of don't-care bits provides flexibility in defining the broadcast address, however in most applications, a broadcast address is FFh. The following is an example of using broadcast addresses:

```
Slave A:SADDR1111 0001b
    SADEN1111 1010b
    Given1111 1X11b,

Slave B:SADDR1111 0011b
    SADEN1111 1001b
    Given1111 1X11B,

Slave C:SADDR=1111 0010b
    SADEN1111 1101b
    Given1111 1111b
```

For slaves A and B, bit 2 is a don’t care bit; for slave C, bit 2 is set. To communicate with all of the slaves, the master must send an address FFh. To communicate with slaves A and B, but not slave C, the master can send and address FBh.

Registers

Table 26. SCON Register

SCON (S:98h)
Serial Control Register

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

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error bit (SMOD0 = 1**) Clear to reset the error state, not cleared by a valid stop bit. Set by hardware when an invalid stop bit is detected.
–	SM0	**Serial port Mode bit 0 (SMOD0 = 0)** Refer to SM1 for serial port mode selection.
6	SM1	**Serial port Mode bit 1** SM0 SM1 Mode Baud Rate 0 0 Shift Register F_{XTAL}/12 (or F_{XTAL} /6 in mode X2) 0 1 8-bit UART Variable 1 0 9-bit UART F_{XTAL}/64 or F_{XTAL}/32 1 1 9-bit UART Variable
5	SM2	**Serial port Mode 2 bit/Multiprocessor Communication Enable bit** Clear to disable multiprocessor communication feature. Set to enable multiprocessor communication feature in mode 2 and 3.
4	REN	**Reception Enable bit** Clear to disable serial reception. Set to enable serial reception.
3	TB8	**Transmitter Bit 8/Ninth bit to transmit in modes 2 and 3** Clear to transmit a logic 0 in the 9th bit. Set to transmit a logic 1 in the 9th bit.
2	RB8	**Receiver Bit 8/Ninth bit received in modes 2 and 3** Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1.
1	TI	**Transmit Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0 or at the beginning of the stop bit in the other modes.
0	RI	**Receive Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 20 and Figure 21 in the other modes.

Reset Value = 0000 0000b
Bit addressable

Table 27. SADEN Register

SADEN (S:B9h)
Slave Address Mask Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7-0	–	**Mask Data for Slave Individual Address**

Reset Value = 0000 0000b
Not bit addressable

Table 28. SADDR Register

SADDR (S:A9h)
Slave Address Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7-0	–	**Slave Individual Address**

Reset Value = 0000 0000b
Not bit addressable

Table 29. SBUF Register

SBUF (S:99h)
Serial Data Buffer

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

Bit Number	Bit Mnemonic	Description
7-0	–	**Data sent/received by Serial I/O Port**

Reset Value = 0000 0000b
Not bit addressable

Table 30. PCON Register

PCON (S:87h)
Power Control Register

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

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial port Mode bit 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	**Serial port Mode bit 0** Clear to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-Off Flag** Clear to recognize next reset type. Set by hardware when VCC rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General-purpose Flag** Cleared by user for general-purpose usage. Set by user for general-purpose usage.
2	GF0	**General-purpose Flag** Cleared by user for general-purpose usage. Set by user for general-purpose usage.
1	PD	**Power-down mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle mode bit** Clear by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00x1 0000b
Not bit addressable

Timers/Counters

The T89C5115 implements two general-purpose, 16-bit Timers/Counters. Such are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a Timer or 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.

Timer/Counter Operations

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 Figure 31) 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.

Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 22 to Figure 25 show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of TMOD register (see Figure 32) and bits 0, 1, 4 and 5 of TCON register (see Figure 31). TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (T/C0#) 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 INT0# to control Timer operation.

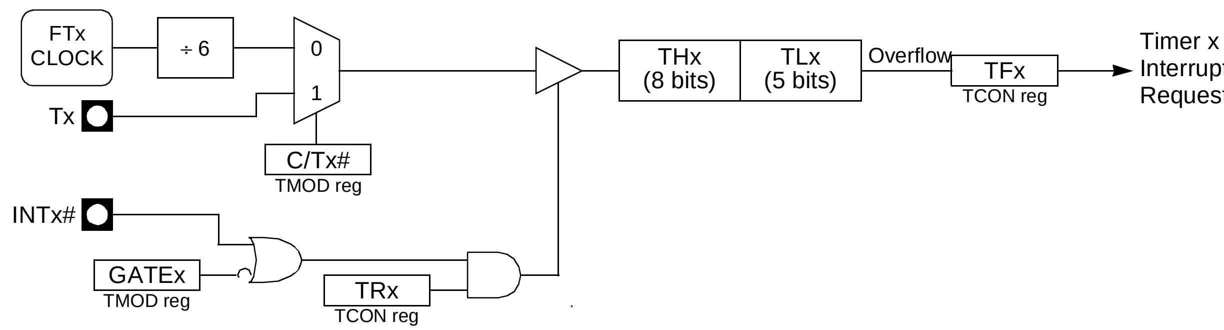
Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag generating an interrupt request.

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

Mode 0 (13-bit Timer)

Mode 0 configures Timer 0 as an 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 22). The upper three bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments TH0 register.

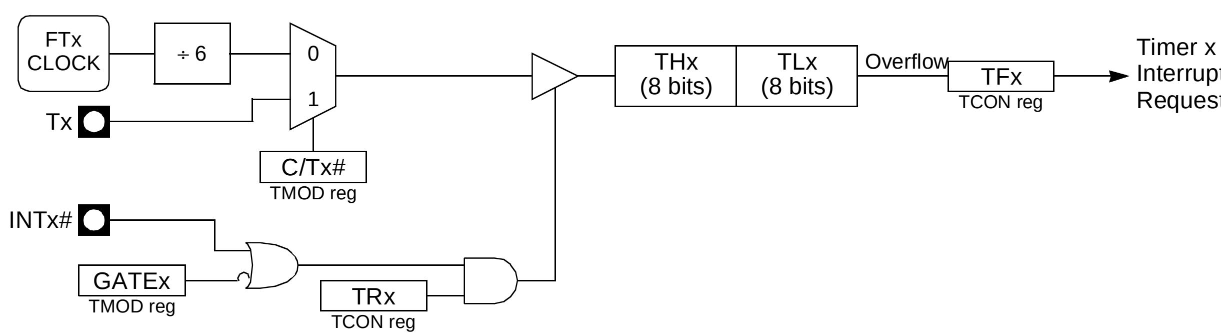
Figure 22. Timer/Counter x (x= 0 or 1) in Mode 0



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 23). The selected input increments TL0 register.

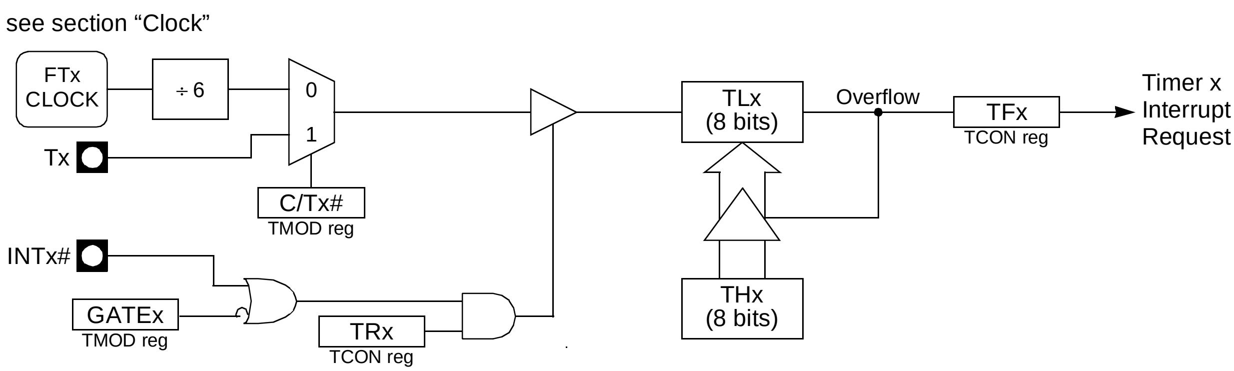
Figure 23. Timer/Counter x (x= 0 or 1) in Mode 1



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 Figure 24). 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.

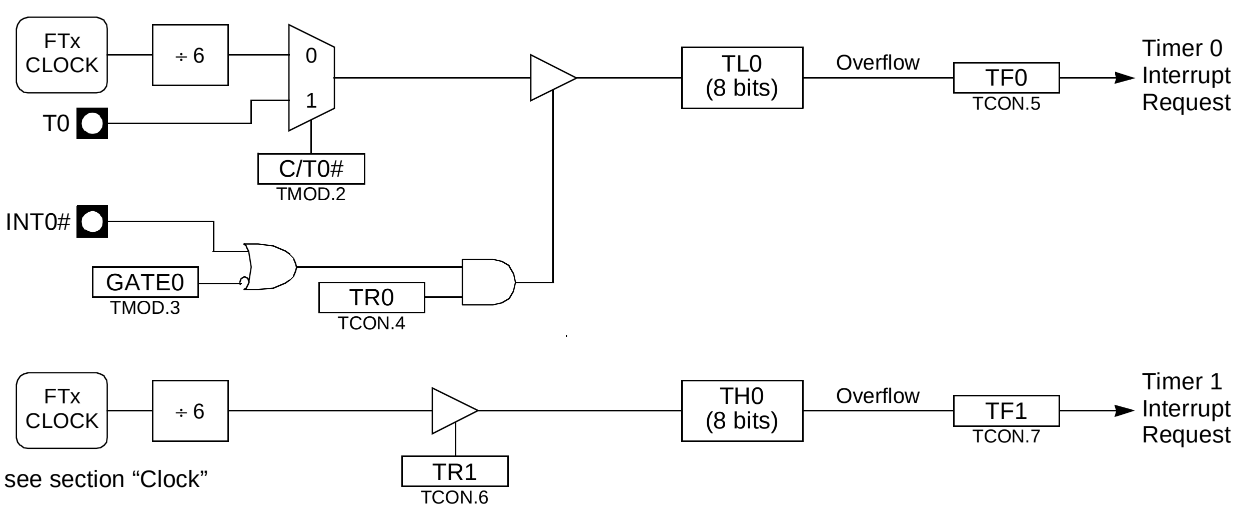
Figure 24. Timer/Counter x (x= 0 or 1) in Mode 2



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 25). 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_{PER} /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 25. Timer/Counter 0 in Mode 3: Two 8-bit Counters



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 22 to Figure 24 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 Figure 32) and bits 2, 3, 6 and 7 of TCON register (see Figure 31). 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 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 Timer/Counter before changing mode.

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 22). The upper 3 bits of TL1 register are ignored. Prescaler overflow increments TH1 register.

Mode 1 (16-bit Timer)

Mode 1 configures Timer 1 as a 16-bit Timer with TH1 and TL1 registers connected in cascade (see Figure 23). The selected input increments TL1 register.

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

Mode 2 configures Timer 1 as an 8-bit Timer (TL1 register) with automatic reload from TH1 register on overflow (see Figure 24). 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.

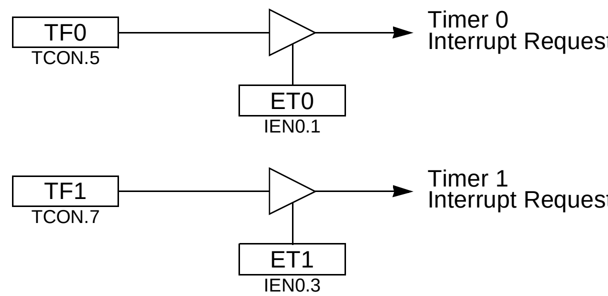
Mode 3 (Halt)

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.

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 26. Timer Interrupt System



Registers

Table 31. 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 INT1# pin.
2	IT1	**Interrupt 1 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	**Interrupt 0 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	**Interrupt 0 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.

Reset Value = 0000 0000b

Table 32. TMOD Register

TMOD (S:89h)
Timer/Counter Mode Control Register

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** <u>M11</u> <u>M01</u> <u>Operating mode</u> 0 0 Mode 0: 8-bit Timer/Counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1: 16-bit Timer/Counter. 1 1 Mode 3: Timer 1 halted. Retains count. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL1).(1)
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** <u>M10</u> <u>M00</u> <u>Operating mode</u> 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	

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

Reset Value = 0000 0000b

Table 33. 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 34. 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 35. 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 36. 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

Timer 2

The T89C5115 Timer 2 is compatible with Timer 2 in the 80C52.

It is a 16-bit timer/counter: the count is maintained by two eight-bit timer registers, TH2 and TL2 that are cascade- connected. It is controlled by T2CON register (See Table) and T2MOD register (See Table 39). Timer 2 operation is similar to Timer 0 and Timer 1. C/$\overline{T2}$ selects $F_{T2\ clock}$/6 (timer operation) or external pin T2 (counter operation) as timer clock. Setting TR2 allows TL2 to be incremented by the selected input.

Timer 2 includes the following enhancements:

- Auto-reload mode (up or down counter)
- Programmable clock-output

Auto-reload Mode

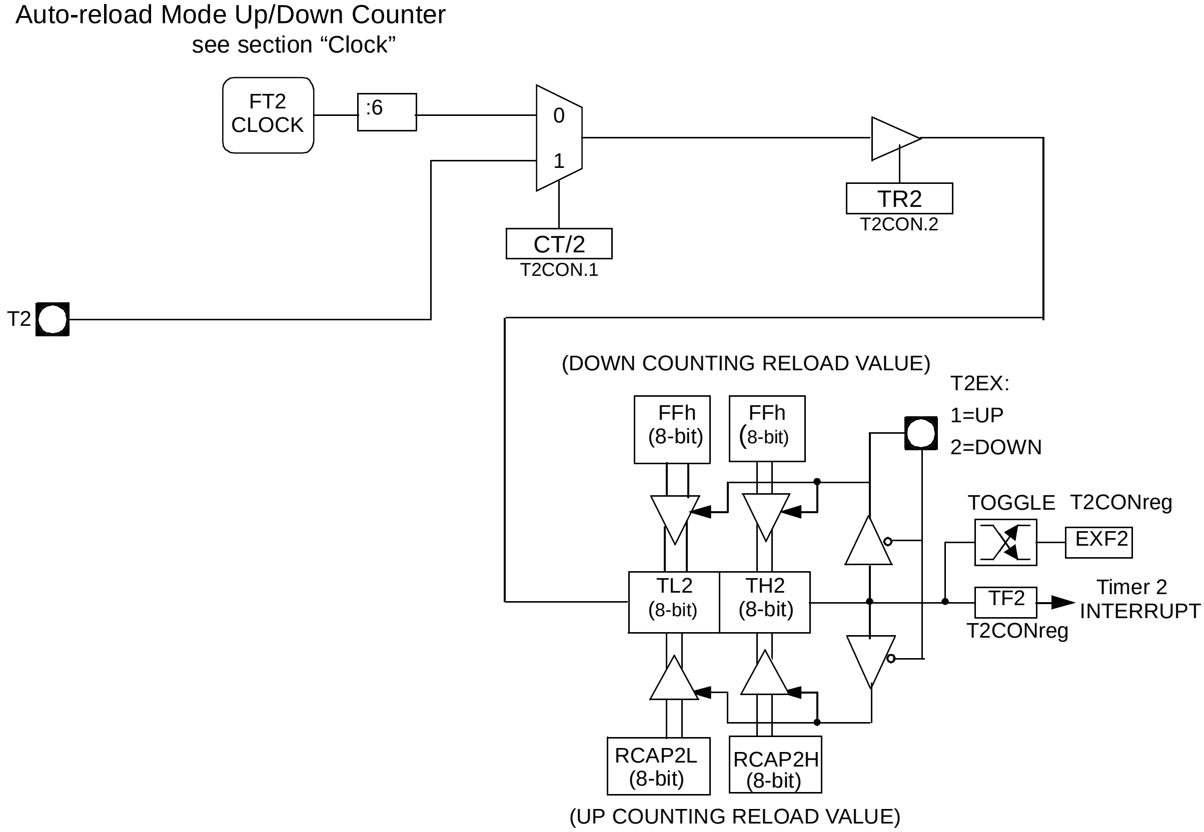
The auto-reload mode configures Timer 2 as a 16-bit timer or event counter with automatic reload. This feature is controlled by the DCEN bit in T2MOD register (See Table 39). Setting the DCEN bit enables Timer 2 to count up or down as shown in Figure 27. In this mode the T2EX pin controls the counting direction.

When T2EX is high, Timer 2 counts up. Timer overflow occurs at FFFFh which sets the TF2 flag and generates an interrupt request. The overflow also causes the 16-bit value in RCAP2H and RCAP2L registers to be loaded into the timer registers TH2 and TL2.

When T2EX is low, Timer 2 counts down. Timer underflow occurs when the count in the timer registers TH2 and TL2 equals the value stored in RCAP2H and RCAP2L registers. The underflow sets TF2 flag and reloads FFFFh into the timer registers.

The EXF2 bit toggles when Timer 2 overflow or underflow, depending on the direction of the count. EXF2 does not generate an interrupt. This bit can be used to provide 17-bit resolution.

Figure 27. Auto-reload Mode Up/Down Counter



Programmable Clock-Output

In clock-out mode, Timer 2 operates as a 50%-duty-cycle, programmable clock generator (See Figure 28). The input clock increments TL2 at frequency $F_{OSC}/2$. The timer repeatedly counts to overflow from a loaded value. At overflow, the contents of RCAP2H and RCAP2L registers are loaded into TH2 and TL2. In this mode, Timer 2 overflows do not generate interrupts. The formula gives the clock-out frequency depending on the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

$$Clock-OutFrequency = \frac{FT2clock}{4 \times (65536 - RCAP2H/RCAP2L)}$$

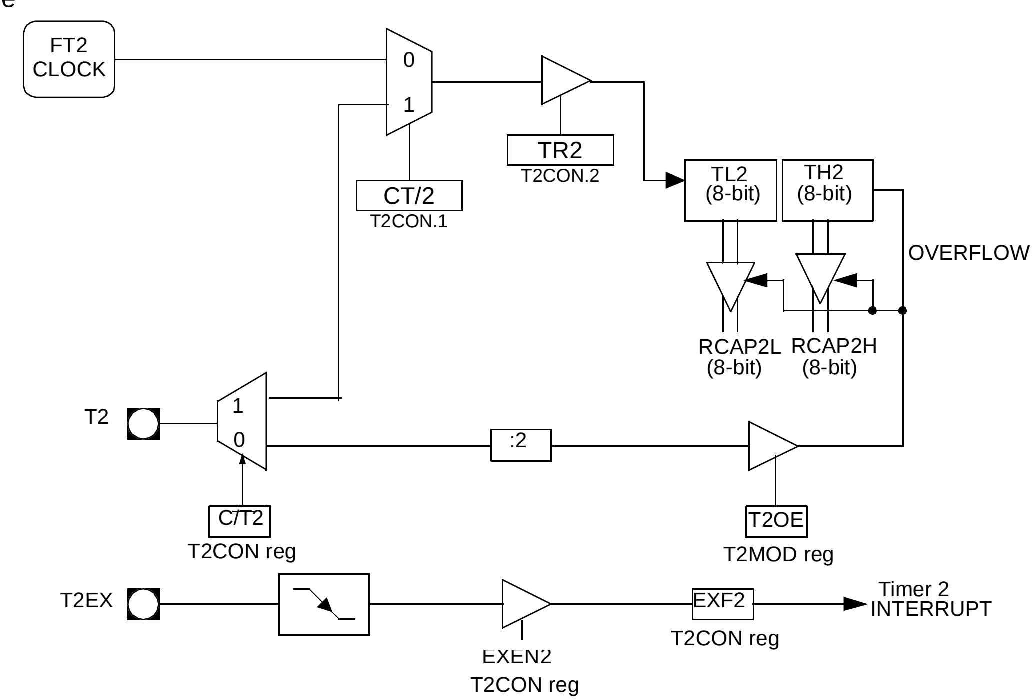
For a 16 MHz system clock in x1 mode, Timer 2 has a programmable frequency range of 61 Hz ($F_{OSC}/2^{16}$) to 4 MHz ($F_{OSC}/4$). The generated clock signal is brought out to T2 pin (P1.0).

Timer 2 is programmed for the clock-out mode as follows:

- Set T2OE bit in T2MOD register.
- Clear C/$\overline{T2}$ bit in T2CON register.
- Determine the 16-bit reload value from the formula and enter it in RCAP2H/RCAP2L registers.
- Enter a 16-bit initial value in timer registers TH2/TL2. It can be the same as the reload value or different depending on the application.
- To start the timer, set TR2 run control bit in T2CON register.

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 28. Clock-out Mode



Registers

Table 37. T2CON Register

T2CON (S:C8h)
Timer 2 Control Register

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** TF2 is not set if RCLK=1 or TCLK = 1. Must be cleared by software. Set by hardware on Timer 2 overflow.
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. Set to cause the CPU to vector to Timer 2 interrupt routine when Timer 2 interrupt is enabled. Must be cleared by software.
5	RCLK	**Receive Clock bit** Clear 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** Clear 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** Clear 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** Clear to turn off Timer 2. Set to turn on Timer 2.
1	C/T2#	**Timer/Counter 2 select bit** Clear for timer operation (input from internal clock system: F_{OSC}). Set for counter operation (input from T2 input pin).
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. Clear to auto-reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2=1. Set to capture on negative transitions on T2EX pin if EXEN2=1.

Reset Value = 0000 0000b
Bit addressable

Table 38. T2MOD Register

T2MOD (S:C9h)
Timer 2 Mode Control Register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	T2OE	DCEN

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	T2OE	**Timer 2 Output Enable bit** Clear to program P1.0/T2 as clock input or I/O port. Set to program P1.0/T2 as clock output.
0	DCEN	**Down Counter Enable bit** Clear to disable Timer 2 as up/down counter. Set to enable Timer 2 as up/down counter.

Reset Value = XXXX XX00b
Not bit addressable

Table 39. TH2 Register

TH2 (S:CDh)
Timer 2 High Byte Register

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

Bit Number	Bit Mnemonic	Description
7-0		High Byte of Timer 2.

Reset Value = 0000 0000b
Not bit addressable

Table 40. TL2 Register

TL2 (S:CCh)
Timer 2 Low Byte Register

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

Bit Number	Bit Mnemonic	Description
7-0		Low Byte of Timer 2.

Reset Value = 0000 0000b
Not bit addressable

Table 41. RCAP2H Register

RCAP2H (S:CBh)
Timer 2 Reload/Capture High Byte Register

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

Bit Number	Bit Mnemonic	Description
7-0		High Byte of Timer 2 Reload/Capture.

Reset Value = 0000 0000b
Not bit addressable

Table 42. RCAP2L Register

RCAP2L (S:CAH)
Timer 2 Reload/Capture Low Byte Register

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

Bit Number	Bit Mnemonic	Description
7-0		Low Byte of Timer 2 Reload/Capture.

Reset Value = 0000 0000b
Not bit addressable

WatchDog Timer

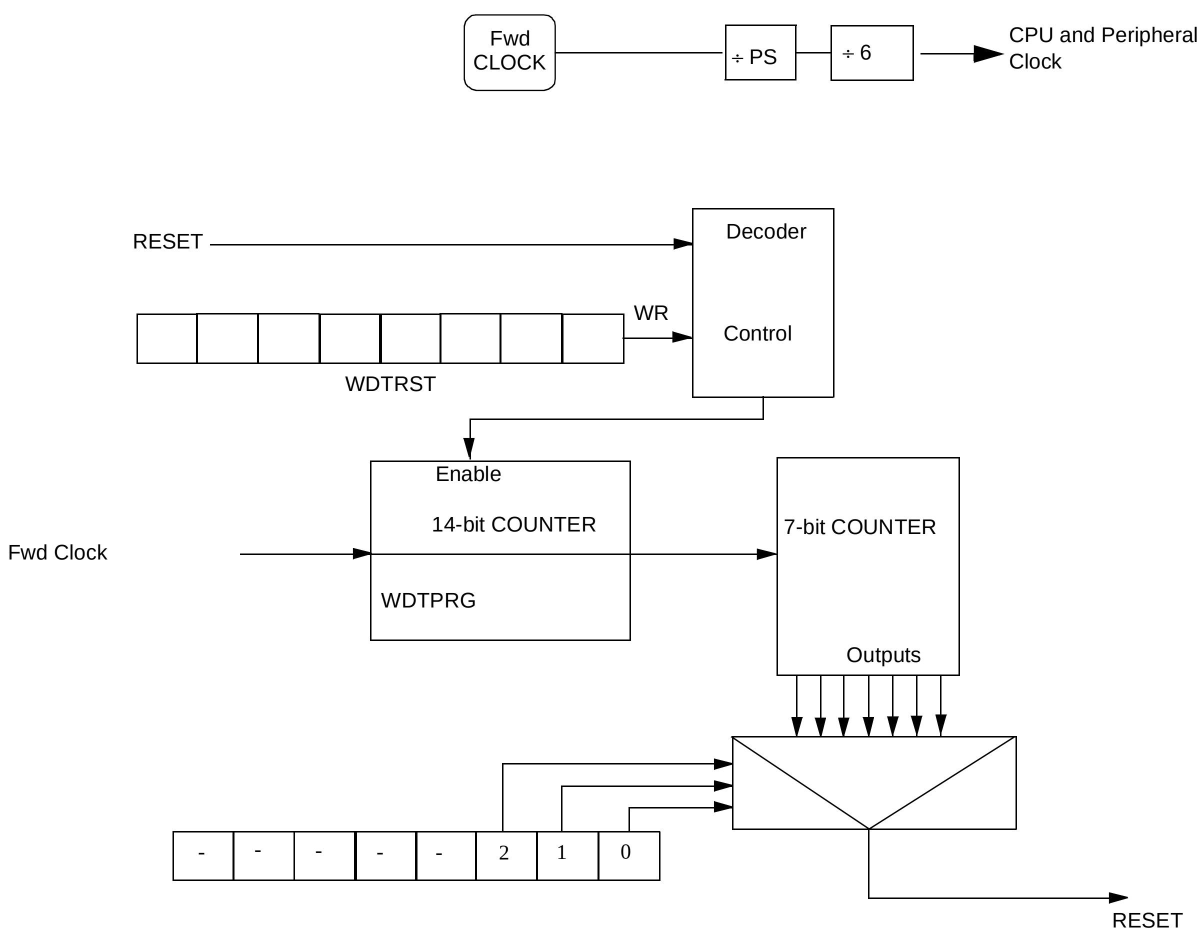
T89C5115 contains a powerful programmable hardware WatchDog Timer (WDT) that automatically resets the chip if it software fails to reset the WDT before the selected time interval has elapsed. It permits large Time-Out ranking from 16ms to 2s @Fosc = 12 MHz in X1 mode.

This WDT consists of a 14-bit counter plus a 7-bit programmable counter, a WatchDog Timer reset register (WDTRST) and a WatchDog Timer programming (WDTPRG) register. When exiting reset, the WDT is -by default- disable.

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

Note: When the WatchDog is enable it is impossible to change its period.

Figure 29. WatchDog Timer



WatchDog Programming

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

Table 43. Machine Cycle Count

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

To compute WD Time-Out, the following formula is applied:

$$FTime-Out = \frac{F_{wd}}{12 \times ((2^{14} \times 2^{Svalue}) - 1)}$$

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

The following table indicates the computed Time-Out value for $Fosc_{XTAL}$ = 12 MHz in X1 mode

Table 44. Time-Out Computation

S2	S1	S0	Fosc = 12 MHz	Fosc = 16 MHz	Fosc = 20 MHz
0	0	0	16.38 ms	12.28 ms	9.82 ms
0	0	1	32.77 ms	24.57 ms	19.66 ms
0	1	0	65.54 ms	49.14 ms	39.32 ms
0	1	1	131.07 ms	98.28 ms	78.64 ms
1	0	0	262.14 ms	196.56 ms	157.28 ms
1	0	1	524.29 ms	393.12 ms	314.56 ms
1	1	0	1.05 sec	786.24 ms	629.12 ms
1	1	1	2.10 sec	1.57 s	1.25 ms

WatchDog Timer during Power-down Mode and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are 2 methods of exiting Power-down mode: by a hardware reset or via a level activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, the WatchDog is disabled. Exiting Power-down with an interrupt is significantly different. The interrupt shall be held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down.

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

In the Idle mode, the oscillator continues to run. To prevent the WDT from resetting T89C5115 while in Idle mode, the user should always set up a timer that will periodically exit Idle, service the WDT, and re-enter Idle mode.

Register

Table 45. WDTPRG Register

WDTPRG (S:A7h)
WatchDog Timer Duration Programming Register

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	S2	**WatchDog Timer Duration selection bit 2** Work in conjunction with bit 1 and bit 0.
1	S1	**WatchDog Timer Duration selection bit 1** Work in conjunction with bit 2 and bit 0.
0	S0	**WatchDog Timer Duration selection bit 0** Work in conjunction with bit 1 and bit 2.

Reset Value = xxxx x000b

Table 46. WDTRST Register

WDTRST (S:A6h Write only)
WatchDog Timer Enable Register

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

Bit Number	Bit Mnemonic	Description
7	-	WatchDog Control Value

Reset Value = 1111 1111b

Note: The WDRST register is used to reset/enable the WDT by writing 1EH then E1H in sequence without instruction between these two sequences.

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 of the following signals:

- PCA clock frequency/6 (see “clock” section)
- PCA clock frequency/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,
- pulse width modulator.

When the compare/capture modules are programmed in 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/Os. These pins are listed below. If the port is not used for the PCA, it can still be used for standard I/O.

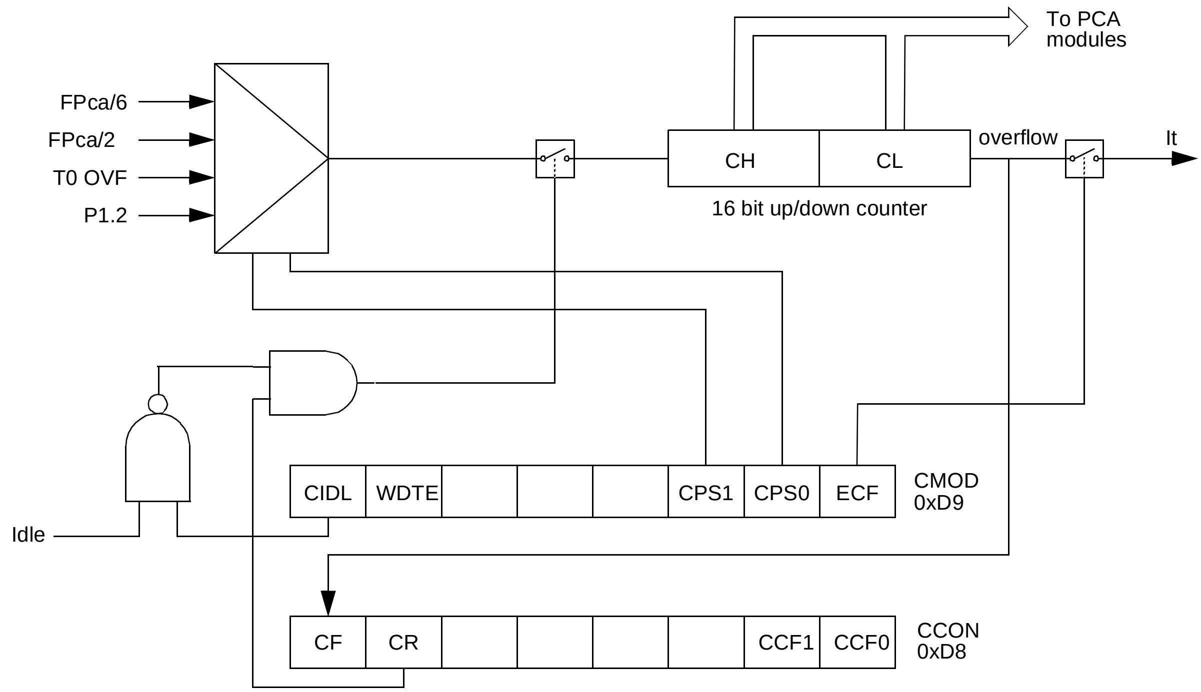
PCA Component	External I/O Pin
16-bit Counter	P1.2/ECI
16-bit Module 0	P1.3/CEX0
16-bit Module 1	P1.4/CEX1

PCA Timer

The PCA timer is a common time base for all five modules (see Figure 9). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR (see Table 8) and can be programmed to run at:

- 1/6 the PCA clock frequency.
- 1/2 the PCA clock frequency.
- the Timer 0 overflow.
- the input on the ECI pin (P1.2).

Figure 30. PCA Timer/Counter



The CMOD register includes three additional bits associated with the PCA.

- The CIDL bit which allows the PCA to stop during idle mode.
- The WDTE bit which enables or disables the WatchDog function on module 4.
- The ECF bit which when set causes an interrupt and the PCA overflow flag CF in CCON register to be set when the PCA timer overflows.

The CCON register contains the run control bit for the PCA and the flags for the PCA timer and each module.

- The CR bit must be set to run the PCA. The PCA is shut off by clearing this bit.
- The CF bit is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in CMOD register is set. The CF bit can only be cleared by software.
- The CCF0:1 bits are the flags for the modules (CCF0 for module0...) and are set by hardware when either a match or a capture occurs. These flags also can be cleared by software.

PCA Modules

Each one of the five compare/capture modules has six possible functions. It can perform:

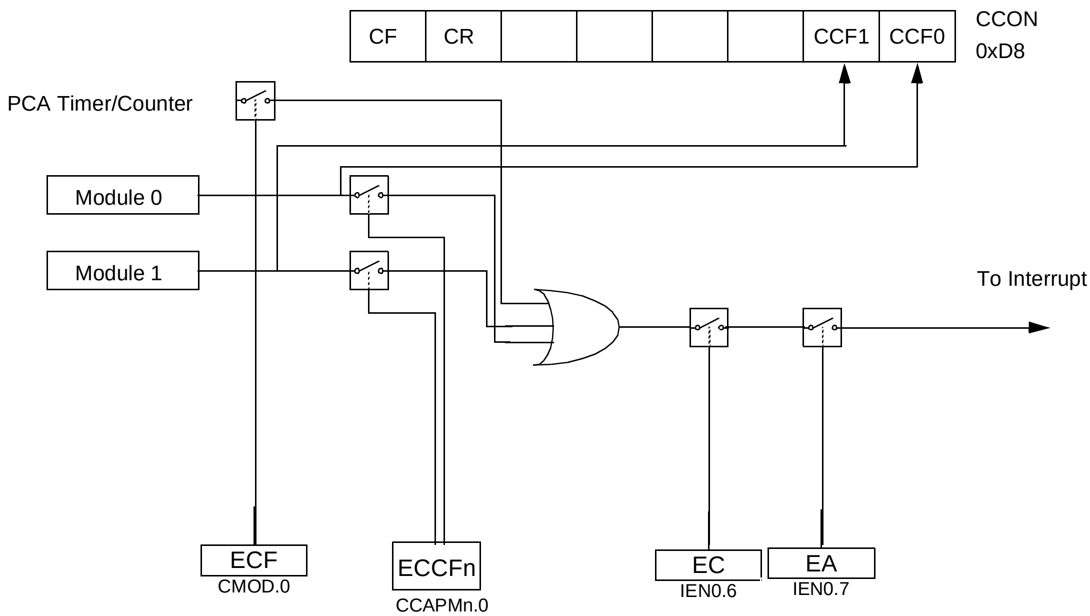
- 16-bit Capture, positive-edge triggered
- 16-bit Capture, negative-edge triggered
- 16-bit Capture, both positive and negative-edge triggered
- 16-bit Software Timer
- 16-bit High Speed Output
- 8-bit Pulse Width Modulator.

Each module in the PCA has a special function register associated with it (CCAPM0 for module 0 ...). The CCAPM0:1 registers contain the bits that control the mode that each module will operate in.

- The ECCF bit enables the CCF flag in the CCON register to generate an interrupt when a match or compare occurs in the associated module.
- The PWM bit enables the pulse width modulation mode.
- The TOG bit when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register.
- The match bit MAT when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.
- The two bits CAPN and CAPP in CCAPMn register determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled.
- The bit ECOM in CCAPM register when set enables the comparator function.

PCA Interrupt

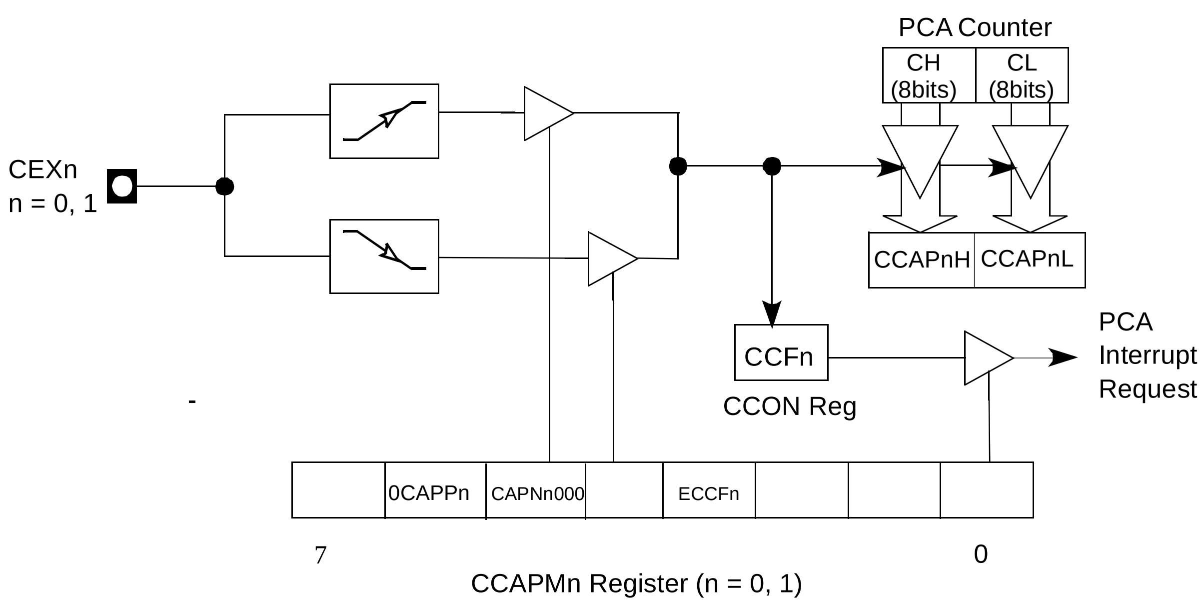
Figure 31. PCA Interrupt System



PCA Capture Mode

To use one of the PCA modules in capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated.

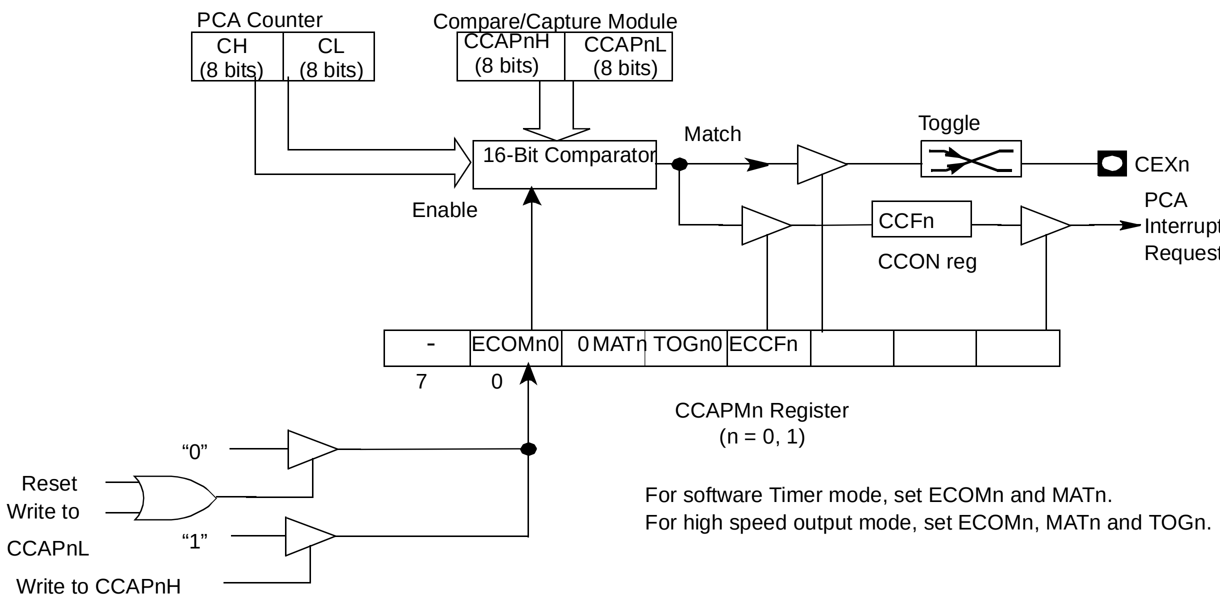
Figure 32. PCA Capture Mode



16-bit Software Timer Mode

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set.

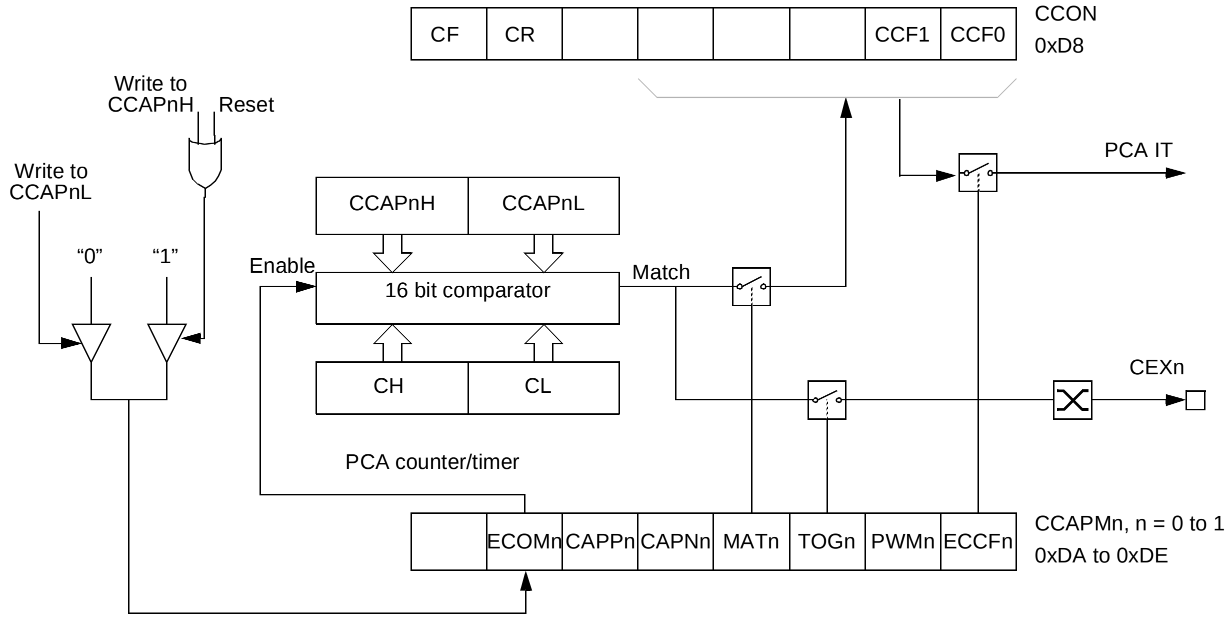
Figure 33. PCA 16-bit Software Timer and High Speed Output Mode



High Speed Output Mode

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set.

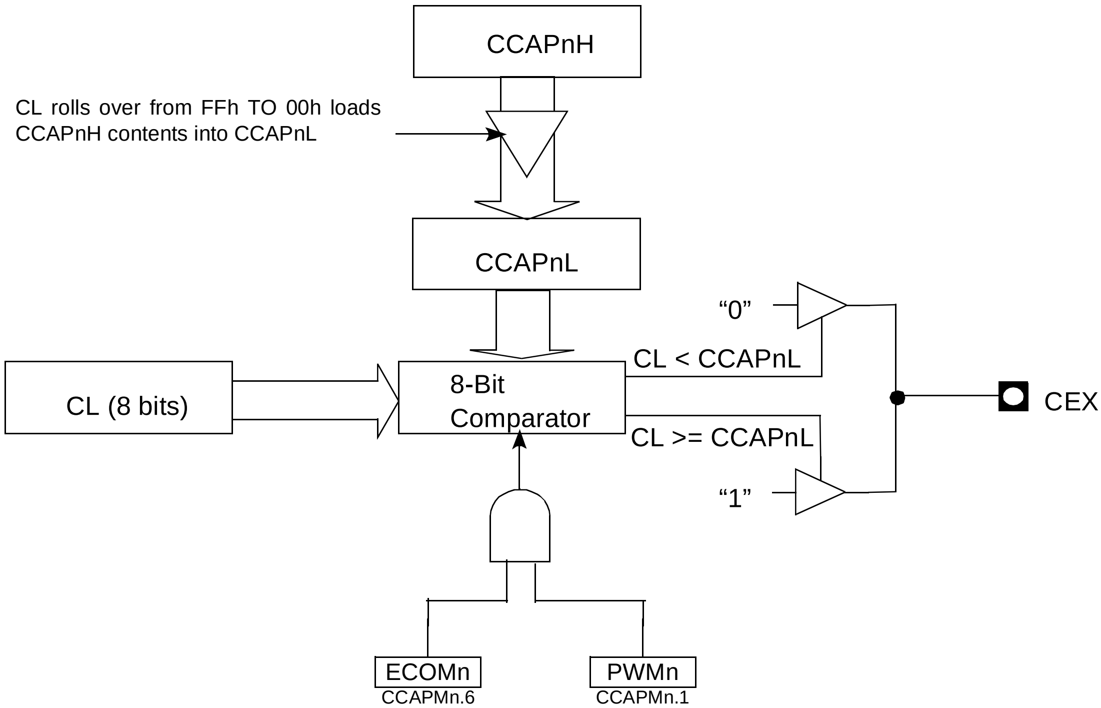
Figure 34. PCA High Speed Output Mode



Pulse Width Modulator Mode

All the PCA modules can be used as PWM outputs. The output frequency depends on the source for the PCA timer. All the modules will have the same output frequency 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 it, the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. the allows the PWM to be updated without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

Figure 35. PCA PWM Mode



PCA Registers

Table 47. CMOD Register

CMOD (S:D9h)
PCA Counter Mode Register

7	6	5	4	3	2	1	0
CIDL	WDTE	-	-	-	CPS1	CPS0	ECF

Bit Number	Bit Mnemonic	Description
7	CIDL	**PCA Counter Idle Control bit** Clear to let the PCA run during Idle mode. Set to stop the PCA when Idle mode is invoked.
6	WDTE	**WatchDog Timer Enable** Clear to disable WatchDog Timer function on PCA Module 4, Set to enable it.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	CPS1	**EWC Count Pulse Select bits** **CPS1 CPS0 Clock Source** 0 0 Internal Clock, FPca/6 0 1 Internal Clock, FPca/2 1 0 Timer 0 overflow 1 1 External clock at ECI/P1.2 pin (Max. Rate = FPca/4)
1	CPS0	
0	ECF	**Enable PCA Counter Overflow Interrupt bit** Clear to disable CF bit in CCON register to generate an interrupt. Set to enable CF bit in CCON register to generate an interrupt.

Reset Value = 00XX X000b

Table 48. CCON Register

CCON (S:D8h)
PCA Counter Control Register

7	6	5	4	3	2	1	0
CF	CR	-	-	-	-	CCF1	CCF0

Bit Number	Bit Mnemonic	Description
7	CF	**PCA Timer/Counter Overflow flag** Set by hardware when the PCA Timer/Counter rolls over. This generates a PCA interrupt request if the ECF bit in CMOD register is set. Must be cleared by software.
6	CR	**PCA Timer/Counter Run Control bit** Clear to turn the PCA Timer/Counter off. Set to turn the PCA Timer/Counter on.
5-2	-	**Reserved** The value read from these bist are indeterminate. Do not set these bits.
1	CCF1	**PCA Module 1 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF 1 bit in CCAPM 1 register is set. Must be cleared by software.
0	CCF0	**PCA Module 0 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF 0 bit in CCAPM 0 register is set. Must be cleared by software.

Reset Value = 00xx xx00b

Table 49. CCAPnH Registers

CCAP0H (S:FAh)
CCAP1H (S:FBh)
PCA High Byte Compare/Capture Module n Register (n=0..1)

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

Bit Number	Bit Mnemonic	Description
7:0	CCAPnH 7:0	High byte of EWC-PCA comparison or capture values

Reset Value = 0000 0000b

Table 50. CCAPnL Registers

CCAP0L (S:EAh)
CCAP1L (S:EBh)
PCA Low Byte Compare/Capture Module n Register (n=0..1)

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

Bit Number	Bit Mnemonic	Description
7:0	CCAPnL 7:0	Low byte of EWC-PCA comparison or capture values

Reset Value = 0000 0000b

Table 51. CCAPMn Registers

CCAPM0 (S:DAh)
CCAPM1 (S:DBh)
PCA Compare/Capture Module n Mode registers (n=0..1)

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 Compare Mode Module x bit** Clear to disable the Compare function. Set to enable the Compare function. The Compare function is used to implement the software Timer, the high-speed output, the Pulse Width Modulator (PWM) and the WatchDog Timer (WDT).
5	CAPPn	**Capture Mode (Positive) Module x bit** Clear to disable the Capture function triggered by a positive edge on CEXx pin. Set to enable the Capture function triggered by a positive edge on CEXx pin
4	CAPNn	**Capture Mode (Negative) Module x bit** Clear to disable the Capture function triggered by a negative edge on CEXx pin. Set to enable the Capture function triggered by a negative edge on CEXx pin.
3	MATn	**Match Module x bit** Set when a match of the PCA Counter with the Compare/Capture register sets CCFx bit in CCON register, flagging an interrupt.
2	TOGn	**Toggle Module x bit** The toggle mode is configured by setting ECOMx, MATx and TOGx bits. Set when a match of the PCA Counter with the Compare/Capture register toggles the CEXx pin.
1	PWMn	**Pulse Width Modulation Module x Mode bit** Set to configure the module x as an 8-bit Pulse Width Modulator with output waveform on CEXx pin.
0	ECCFn	**Enable CCFx Interrupt bit** Clear to disable CCFx bit in CCON register to generate an interrupt request. Set to enable CCFx bit in CCON register to generate an interrupt request.

Reset Value = X000 0000b

Table 52. CH Register

CH (S:F9h)
PCA Counter Register High Value

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

Bit Number	Bit Mnemonic	Description
7:0	CH 7:0	High byte of Timer/Counter

Reset Value = 0000 00000b

Table 53. CL Register

CL (S:E9h)
PCA counter Register Low Value

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

Bit Number	Bit Mnemonic	Description
7:0	CL0 7:0	Low byte of Timer/Counter

Reset Value = 0000 00000b

Analog-to-Digital Converter (ADC)

This section describes the on-chip 10-bit analog-to-digital converter of the T89C5115. Eight ADC channels are available for sampling of the external sources AN0 to AN7. An analog multiplexer allows the single ADC converter to select one from the 8 ADC channels as ADC input voltage (ADCIN). ADCIN is converted by the 10 bit-cascaded potentiometric ADC.

Two kind of conversion are available:
- Standard conversion (8 bits).
- Precision conversion (10 bits).

For the precision conversion, set bit PSIDLE in ADCON register and start conversion. The device is in a pseudo-idle mode, the CPU does not run but the peripherals are always running. This mode allows digital noise to be as low as possible, to ensure high precision conversion.

For this mode it is necessary to work with end of conversion interrupt, which is the only way to wake the device up.

If another interrupt occurs during the precision conversion, it will be treated only after this conversion is ended.

Features

- 8 channels with multiplexed inputs
- 10-bit cascaded potentiometric ADC
- Conversion time 16 micro-seconds (typ.)
- Zero Error (offset) ± 2 LSB max
- Positive External Reference Voltage Range (VREF) 2.4V to 3.0V (typ.)
- ADCIN Range 0V to 3V
- Integral non-linearity typical 1 LSB, max. 2 LSB
- Differential non-linearity typical 0.5 LSB, max. 1 LSB
- Conversion Complete Flag or Conversion Complete Interrupt
- Selectable ADC Clock

ADC Port1 I/O Functions

Port 1 pins are general I/O that are shared with the ADC channels. The channel select bit in ADCF register define which ADC channel/port1 pin will be used as ADCIN. The remaining ADC channels/port1 pins can be used as general-purpose I/O or as the alternate function that is available.

A conversion launched on a channel which are not selected on ADCF register will not have any effect.

Figure 36. ADC Description

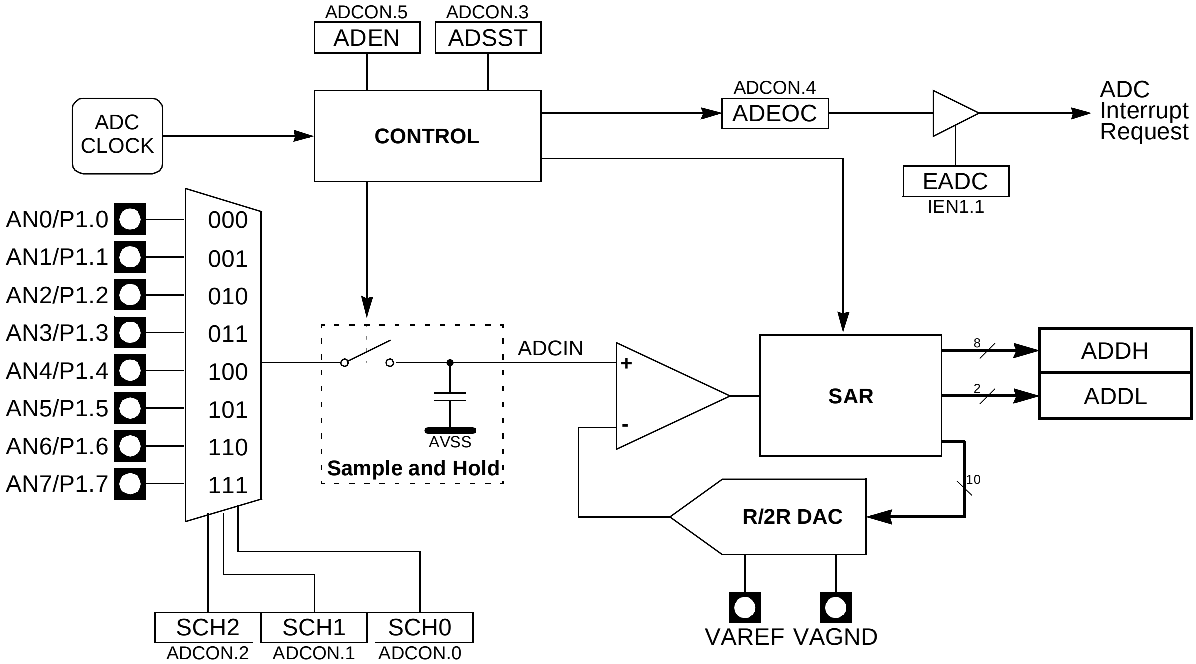
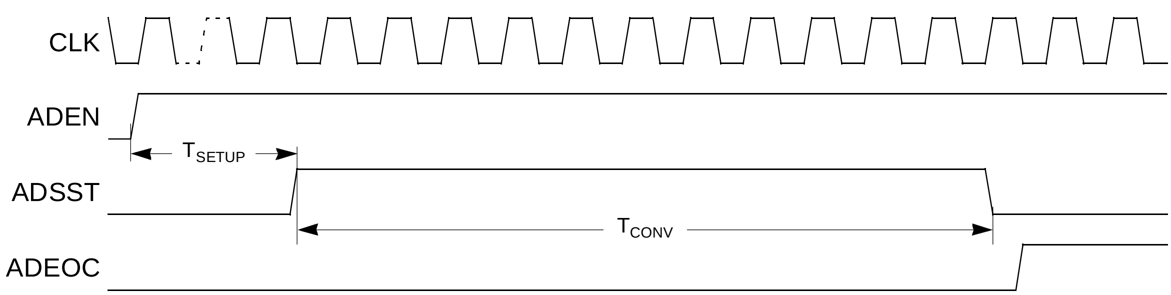


Figure 37 shows the timing diagram of a complete conversion. For simplicity, the figure depicts the waveforms in idealized form and do not provide precise timing information. For ADC characteristics and timing parameters refer to the Section “AC Characteristics” of the T89C5115 datasheet.

Figure 37. Timing Diagram



Note: Tsetup min = 4 us
Tconv=11 clock ADC = 1sample and hold + 10 bit conversion
The user must ensure that 4 us minimum time between setting ADEN and the start of the first conversion.

ADC Converter Operation

A start of single A/D conversion is triggered by setting bit ADSST (ADCON.3).

After completion of the A/D conversion, the ADSST bit is cleared by hardware.

The end-of-conversion flag ADEOC (ADCON.4) is set when the value of conversion is available in ADDH and ADDL, it must be cleared by software. If the bit EADC (IEN1.1) is set, an interrupt occur when flag ADEOC is set (see Figure 39). Clear this flag for re-arming the interrupt.

The bits SCH0 to SCH2 in ADCON register are used for the analog input channel selection.

Table 54. Selected Analog Input

SCH2	SCH1	SCH0	Selected Analog input
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

Voltage Conversion

When the ADCIN is equals to VAREF the ADC converts the signal to 3FFh (full scale). If the input voltage equals VAGND, the ADC converts it to 000h. Input voltage between VAREF and VAGND are a straight-line linear conversion. All other voltages will result in 3FFh if greater than VAREF and 000h if less than VAGND.

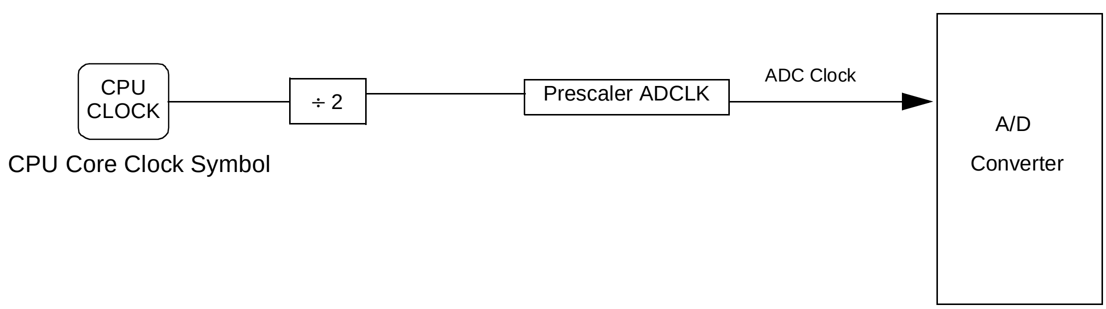
Note that ADCIN should not exceed VAREF absolute maximum range! (see section "AC-DC")

Clock Selection

The ADC clock is the same as CPU.

The maximum clock frequency for ADC is 700 KHz. A prescaler is featured (ADCCLK) to generate the ADC clock from the oscillator frequency.

Figure 38. A/D Converter Clock



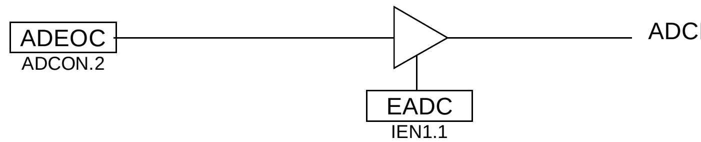
ADC Standby Mode

When the ADC is not used, it is possible to set it in standby mode by clearing bit ADEN in ADCON register. In this mode its power dissipation is about 1 uW.

IT ADC Management

An interrupt end-of-conversion will occurs when the bit ADEOC is activated and the bit EADC is set. For re-arming the interrupt the bit ADEOC must be cleared by software.

Figure 39. ADC interrupt structure



Routine Examples

1. Configure P1.2 and P1.3 in ADC channels

```
// configure channel P1.2 and P1.3 for ADC
 ADCF = 0Ch

// Enable the ADC
 ADCON = 20h
```

2. Start a standard conversion

```
// The variable "channel" contains the channel to convert
// The variable "value_converted" is an unsigned int
// Clear the field SCH[2:0]
 ADCON &= F8h
// Select channel
 ADCON |= channel
// Start conversion in standard mode
 ADCON |= 08h
// Wait flag End of conversion
 while((ADCON & 01h)!= 01h)
// Clear the End of conversion flag
 ADCON &= EFh
// read the value
 value_converted = (ADDH << 2)+(ADDL)
```

3. Start a precision conversion (need interrupt ADC)

```
// The variable "channel" contains the channel to convert
// Enable ADC
 EADC = 1
// clear the field SCH[2:0]
 ADCON &= F8h
// Select the channel
 ADCON |= channel
// Start conversion in precision mode
 ADCON |= 48h
```

Note: to enable the ADC interrupt:
EA = 1

Registers

Table 55. ADCF Register
ADCF (S:F6h)
ADC Configuration

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

Bit Number	Bit Mnemonic	Description
7-0	CH 0:7	**Channel Configuration** Set to use P1.x as ADC input. Clear to use P1.x as standart I/O port.

Reset Value = 0000 0000b

Table 56. ADCON Register
ADCON (S:F3h)
ADC Control Register

7	6	5	4	3	2	1	0
-	PSIDLE	ADEN	ADEOC	ADSST	SCH2	SCH1	SCH0

Bit Number	Bit Mnemonic	Description
7	-	
6	PSIDLE	**Pseudo Idle mode (best precision)** Set to put in idle mode during conversion Clear to convert without idle mode.
5	ADEN	**Enable/Standby Mode** Set to enable ADC Clear for Standby mode (power dissipation 1 uW).
4	ADEOC	**End Of Conversion** 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** Set to start an A/D conversion. Cleared by hardware after completion of the conversion
2-0	SCH2:0	**Selection of channel to convert** see Table 54

Reset Value = X000 0000b

Table 57. ADCLK Register
ADCLK (S:F2h)
ADC Clock Prescaler

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

Bit Number	Bit Mnemonic	Description
7-5	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
4-0	PRS4:0	**Clock Prescaler** f_{ADC} = fcpu clock/ (4 (or 2 in X2 mode)* (PRS +1))

Reset Value = XXX0 0000b

Table 58. ADDH Register
ADDH (S:F5h Read Only)
ADC Data High byte register

7	6	5	4	3	2	1	0
ADAT 9	ADAT 8	ADAT 7	ADAT 6	ADAT 5	ADAT 4	ADAT 3	ADAT 2

Bit Number	Bit Mnemonic	Description
7-0	ADAT9:2	**ADC result** bits 9-2

Reset Value = 00h

Table 59. ADDL Register
ADDL (S:F4h Read Only)
ADC Data Low byte register

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

Bit Number	Bit Mnemonic	Description
7-2	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
1-0	ADAT1:0	**ADC result** bits 1-0

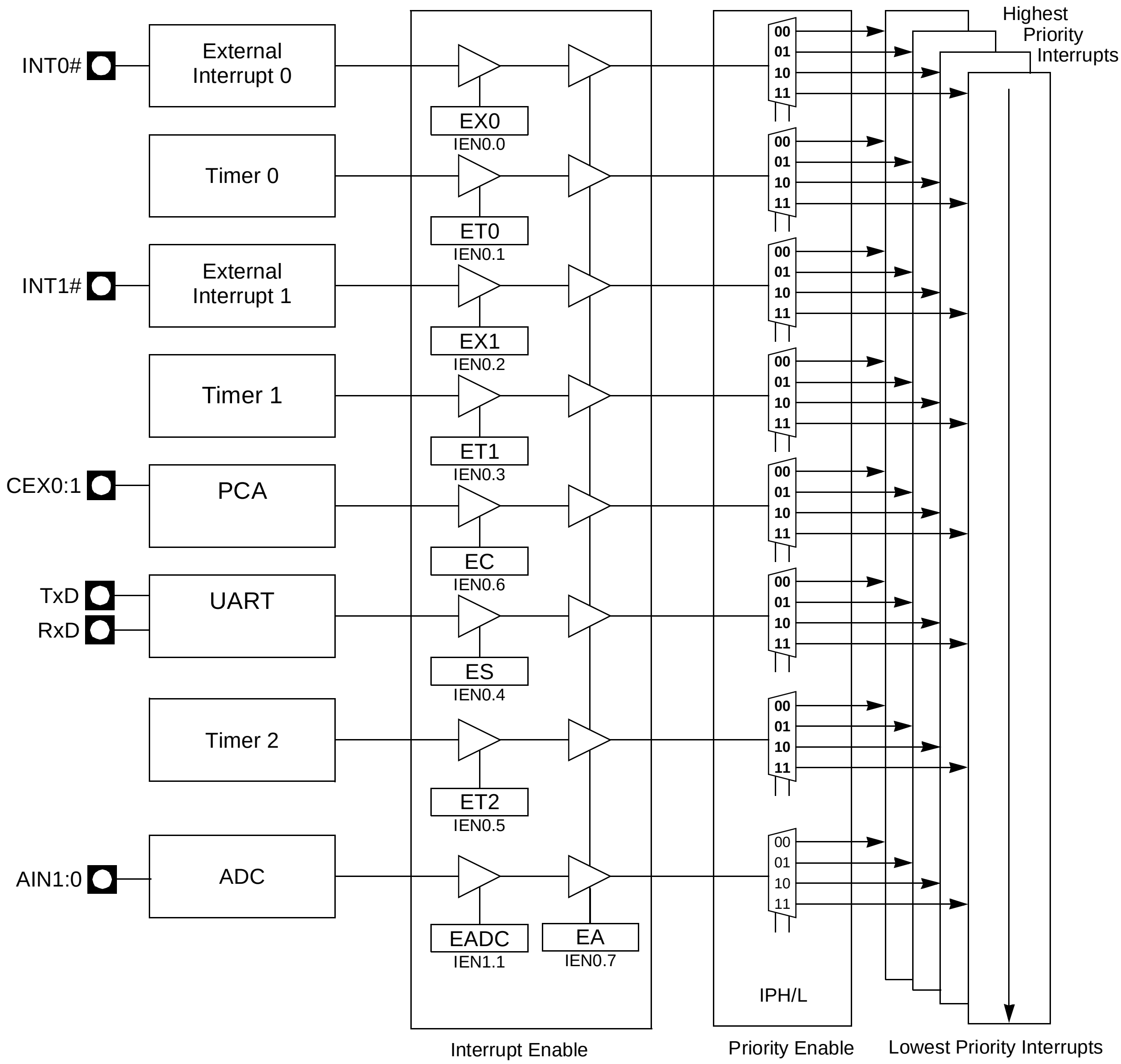
Reset Value = 00h

Interrupt System

Introduction

The T89C5115 has a total of 8 interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (timers 0, 1 and 2), a serial port interrupt, a PCAand an ADC. These interrupts are shown below.

Figure 40. Interrupt Control System



Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable register. This register also contains a global disable bit which must be cleared to disable all the interrupts at the same time.

Each interrupt source can also be individually programmed to one of four priority levels by setting or clearing a bit in the Interrupt Priority registers. The Table below shows the bit values and priority levels associated with each combination.

Table 60. Priority Level Bit Values

IPH.x	IPL.x	Interrupt Level Priority
0	0	0 (Lowest)
0	1	1
1	0	2
1	1	3 (Highest)

A low-priority interrupt can be interrupted by a high priority interrupt but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of the higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence, see Table 61.

Table 61. Interrupt Priority Within Level

Interrupt Name	Interrupt Address Vector	Priority Number
external interrupt (INT0)	0003h	1
Timer0 (TF0)	000Bh	2
external interrupt (INT1)	0013h	3
Timer1 (TF1)	001Bh	4
PCA (CF or CCFn)	0033h	5
UART (RI or TI)	0023h	6
Timer2 (TF2)	002Bh	7
ADC (ADCI)	0043h	8

Registers

Table 62. IEN0 Register

IEN0 (S:A8h)
Interrupt Enable Register

7	6	5	4	3	2	1	0
EA	EC	ET2	ES	ET1	EX1	ET0	EX0

Bit Number	Bit Mnemonic	Description
7	EA	**Enable All interrupt bit** Clear to disable all interrupts. Set to enable all interrupts. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its interrupt enable bit.
6	EC	**PCA Interrupt Enable** Clear to disable the PCA interrupt. Set to enable the PCA interrupt.
5	ET2	**Timer 2 overflow interrupt Enable bit** Clear to disable Timer 2 overflow interrupt. Set to enable Timer 2 overflow interrupt.
4	ES	**Serial port Enable bit** Clear to disable serial port interrupt. Set to enable serial port interrupt.
3	ET1	**Timer 1 overflow interrupt Enable bit** Clear to disable timer 1 overflow interrupt. Set to enable timer 1 overflow interrupt.
2	EX1	**External interrupt 1 Enable bit** Clear to disable external interrupt 1. Set to enable external interrupt 1.
1	ET0	**Timer 0 overflow interrupt Enable bit** Clear to disable timer 0 overflow interrupt. Set to enable timer 0 overflow interrupt.
0	EX0	**External interrupt 0 Enable bit** Clear to disable external interrupt 0. Set to enable external interrupt 0.

Reset Value = 0000 0000b
bit addressable

Table 63. IEN1 Register

IEN1 (S:E8h)
Interrupt Enable Register

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	EADC	**ADC Interrupt Enable bit** Clear to disable the ADC interrupt. Set to enable the ADC interrupt.
0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value = xxxx xx0xb
bit addressable

Table 64. IPL0 Register

IPL0 (S:B8h)
Interrupt Enable Register

7	6	5	4	3	2	1	0
-	PPC	PT2	PS	PT1	PX1	PT0	PX0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	PPC	**PCA Interrupt Priority bit** Refer to PPCH for priority level.
5	PT2	**Timer 2 overflow interrupt Priority bit** Refer to PT2H for priority level.
4	PS	**Serial port Priority bit** Refer to PSH for priority level.
3	PT1	**Timer 1 overflow interrupt Priority bit** Refer to PT1H for priority level.
2	PX1	**External interrupt 1 Priority bit** Refer to PX1H for priority level.
1	PT0	**Timer 0 overflow interrupt Priority bit** Refer to PT0H for priority level.
0	PX0	**External interrupt 0 Priority bit** Refer to PX0H for priority level.

Reset Value = X000 0000b
bit addressable

Table 65. IPL1 Register

IPL1 (S:F8h)
Interrupt Priority Low Register 1

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	PADCL	**ADC Interrupt Priority level less significant bit.** Refer to PSPIH for priority level.
0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value = xxxx xx0xb
bit addressable

Table 66. IPL0 Register

IPH0 (B7h)
Interrupt High Priority Register

7	6	5	4	3	2	1	0
-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	PPCH	**PCA Interrupt Priority level most significant bit** PPCH PPC Priority level 0 0 Lowest 0 1 1 0 1 1 Highest priority
5	PT2H	**Timer 2 overflow interrupt High Priority bit** PT2H PT2 Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
4	PSH	**Serial port High Priority bit** PSH PS Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
3	PT1H	**Timer 1 overflow interrupt High Priority bit** PT1H PT1 Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
2	PX1H	**External interrupt 1 High Priority bit** PX1H PX1 Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
1	PT0H	**Timer 0 overflow interrupt High Priority bit** PT0H PT0 Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest
0	PX0H	**External interrupt 0 high priority bit** PX0H PX0 Priority Level 0 0 Lowest 0 1 1 0 1 1 Highest

Reset Value = X000 0000b

Table 67. IPH1 Register

IPH1 (S:F7h)
Interrupt high priority Register 1

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

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	PADCH	**ADC Interrupt Priority level most significant bit** PADCH PADCL Priority level 0 0 Lowest 0 1 1 0 1 1 Highest
0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value = xxxx xx0xb

Electrical Characteristics

Absolute Maximum Ratings

Ambiant Temperature Under Bias:
I = industrial .... -40°C to 85°C
Storage Temperature .... -65°C to + 150°C
Voltage on V_{CC} from V_{SS}.....................................-0.5V to + 6V
Voltage on Any Pin from V_{SS} ....-0.5V to V_{CC} + 0.2 V
Power Dissipation .... 1 W

Note: Stresses at or above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.
The power dissipation is based on the maximum allowable die temperature and the thermal resistance of the package.

DC Parameters for Standard Voltage

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

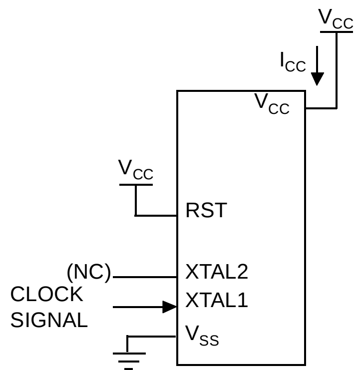
Table 68. DC Parameters in Standard Voltage

Symbol	Parameter	Min	Typ(5)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5		0.2 Vcc - 0.1	V	
V_{IH}	Input High Voltage except XTAL1, RST	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage, XTAL1, RST	0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage, ports 1, 2, 3 and 4(6)			0.3 0.45 1.0	V V V	I_{OL} = 100 µA I_{OL} = 1.6 mA I_{OL} = 3.5 mA
V_{OH}	Output High Voltage, ports 1, 2, 3, 4	V_{CC} - 0.3 V_{CC} - 0.7 V_{CC} - 1.5			V V V	I_{OH} = -10 µA I_{OH} = -30 µA I_{OH} = -60 µA V_{CC} = 5V ± 10%
R_{RST}	RST Pulldown Resistor	50	90	200	kΩ	
I_{IL}	Logical 0 Input Current ports 1, 2, 3 and 4			-50	µA	Vin = 0.45V
I_{LI}	Input Leakage Current			±10	µA	0.45V < Vin < V_{CC}
I_{TL}	Logical 1 to 0 Transition Current, ports 1, 2, 3 and 4			-650	µA	Vin = 2.0V
C_{IO}	Capacitance of I/O Buffer			10	pF	Fc = 1 MHz T_A = 25°C
I_{PD}	Power-down Current		160	350	µA	4.5V < V_{CC} < 5.5V(3)
I_{CC}	Power Supply Current	I_{CCOP} = 0.7 Freq (MHz) + 3 mA I_{CCIDLE} = 0.6 Freq (MHz) + 2 mA				Vcc = SS(1) (2)

Notes:
1. Operating I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns (see Figure 44.), V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} - 0.5V; XTAL2 N.C.; RST = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator used (see Figure 41.).
2. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} - 0.5V; XTAL2 N.C; Port 0 = V_{CC}; RST = V_{SS} (see Figure 42.).
3. Power-down I_{CC} is measured with all output pins disconnected; XTAL2 NC.; RST = V_{SS} (see Figure 43.). In addition, the WDT must be inactive and the POF flag must be set.
4. Typicals are based on a limited number of samples and are not guaranteed. The values listed are at room temperature.

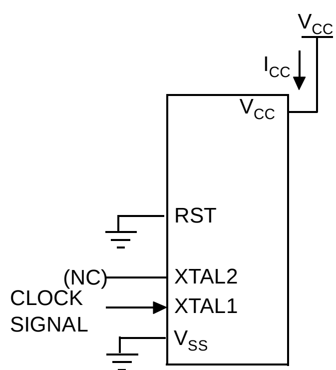
5. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Ports 1, 2 and 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

Figure 41. I_{CC} Test Condition, Active Mode



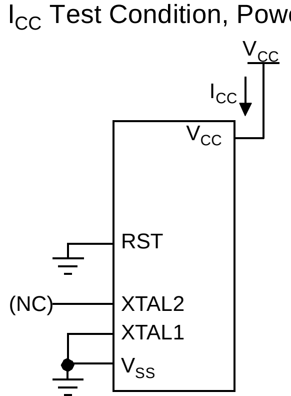
All other pins are disconnected.

Figure 42. I_{CC} Test Condition, Idle Mode



All other pins are disconnected.

Figure 43. I_{CC} Test Condition, Power-down Mode



All other pins are disconnected

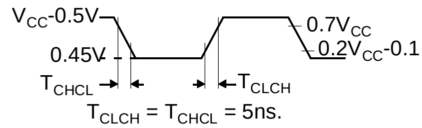


Figure 44. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes

DC Parameters for A/D Converter

Table 69. DC Parameters for AD Converter in Precision conversion

Symbol	Parameter	Min	Typ[1]	Max	Unit	Test Conditions
AVin	Analog input voltage	Vss - 0.2		Vref + 0.6	V	
Rref	Resistance between Vref and Vss	12	16	24	kΩ	
Vref	Reference voltage	2.40		3.00	V	
Cai	Analog input Capacitance		60		pF	During sampling
INL	Integral non linearity		1	2	lsb	
DNL	Differential non linearity		0.5	1	lsb	
OE	Offset error	-2		2	lsb	

Note: 1. Typicals are based on a limited number of samples and are not guaranteed.

AC Parameters

Explanation of the AC Symbols

Each timing symbol has 5 characters. The first character is always a “T” (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example: T_{AVLL} = Time for Address Valid to ALE Low.
T_{LLPL} = Time for ALE Low to $\overline{PSEN}$ Low.

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

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

(Load Capacitance for all outputs = 60 pF.)

Table 70 and Table 74 give the description of each AC symbols.

Table 71, Table 72 and Table 73 give for each range the AC parameter.

Table 72 gives the frequency derating formula of the AC parameter for each speed range description. To calculate each AC symbols. Take the x value and use this value in the formula.

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

Serial Port Timing – Shift Register Mode

Table 70. Symbol Description (F = 40 MHz)

Symbol	Parameter
T_{XLXL}	Serial port clock cycle time
T_{QVHX}	Output data set-up to clock rising edge
T_{XHQX}	Output data hold after clock rising edge
T_{XHDX}	Input data hold after clock rising edge
T_{XHDV}	Clock rising edge to input data valid

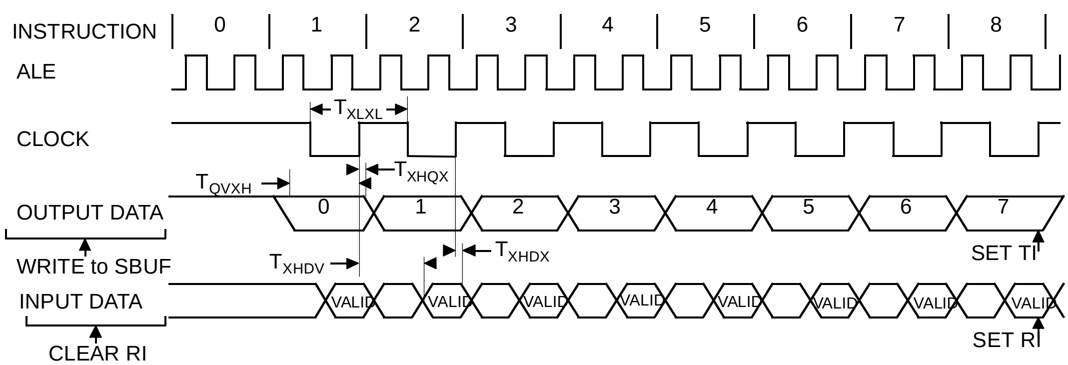
Table 71. AC Parameters for a Fix Clock (F = 40 MHz)

Symbol	Min	Max	Units
T_{XLXL}	300		ns
T_{QVHX}	200		ns
T_{XHQX}	30		ns
T_{XHDX}	0		ns
T_{XHDV}		117	ns

Table 72. AC Parameters for a Variable Clock

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

Shift Register Timing Waveforms

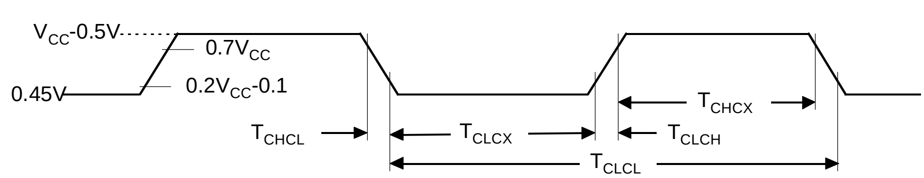


External Clock Drive Characteristics (XTAL1)

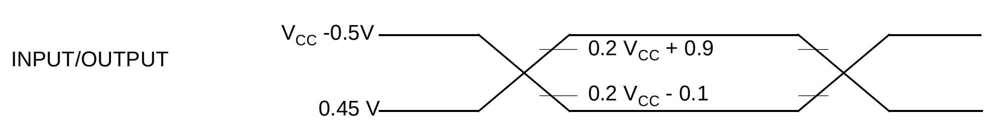
Table 73. AC Parameters

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

External Clock Drive Waveforms

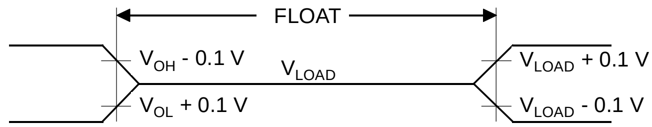


AC Testing Input/Output Waveforms



AC inputs during testing are driven at V_{CC} - 0.5 for a logic “1” and 0.45V for a logic “0”. Timing measurement are made at V_{IH} min for a logic “1” and V_{IL} max for a logic “0”.

Float Waveforms



For timing purposes as port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm$ 20 mA.

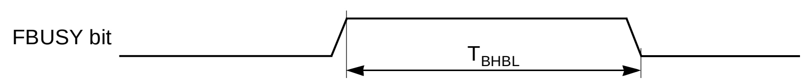
Flash Memory

Table 74. Memory AC Timing

VDD = 5V ± 10% , TA = -40 to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
T_{BHBL}	Flash Internal Busy (Programming) Time		10		ms

Figure 45. Flash Memory - Internal Busy Waveforms



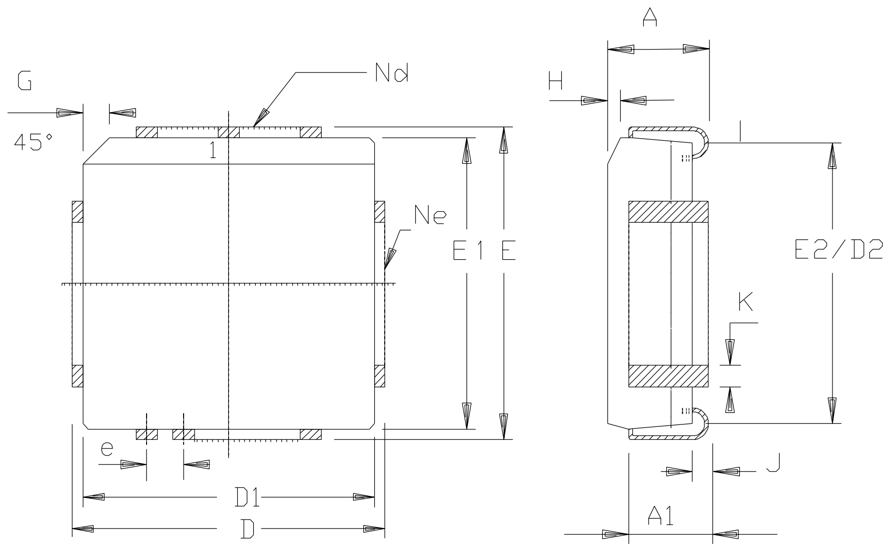
Ordering Information

Table 75. Possible Order Entries

Part-Number	Memory Size	Supply Voltage	Temperature Range	Max Frequency	Package	Packing
T89C5115-SISIM	16K	5V	Industrial	40 MHz	PLCC28	Stick
T89C5115-TISIM	16K	5V	Industrial	40 MHz	SOIC28	Stick
T89C5115-RATIM	16K	5V	Industrial	40 MHz	VQFP32	Tray

Package Drawing

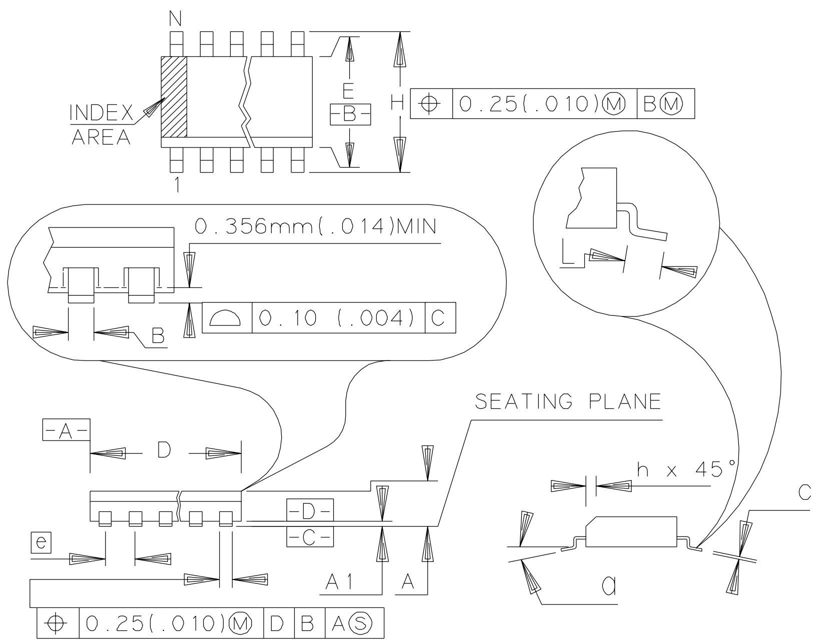
PLCC28



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

Package Drawing

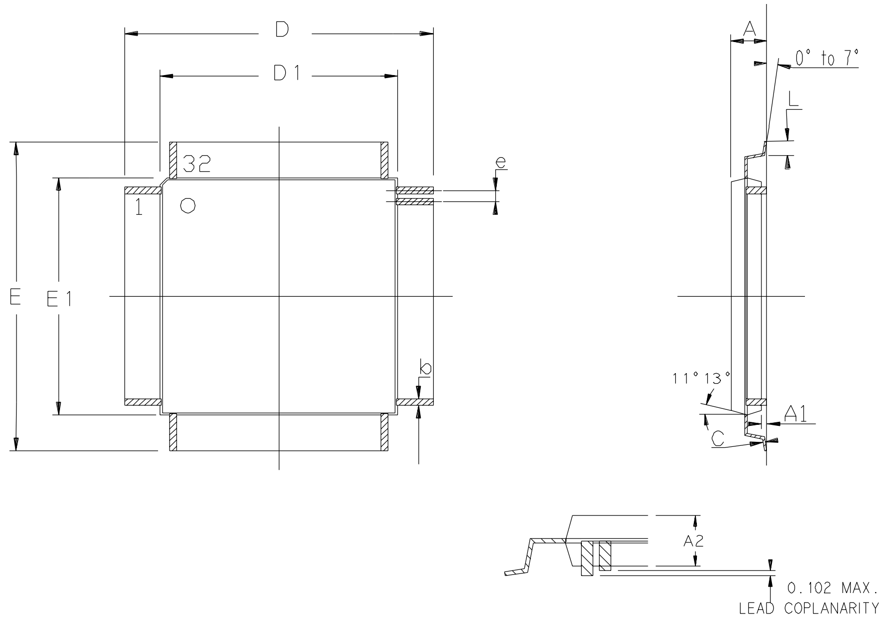
SOIC28



	MM		INCH	
A	2.35	2.65	.093	.104
A1	0.10	0.30	.004	.012
B	0.35	0.49	.014	.019
C	0.23	0.32	.009	.013
D	17.70	18.10	.697	.713
E	7.40	7.60	.291	.299
e	1.27	BSC	.050	BSC
H	10.00	10.65	.394	.419
h	0.25	0.75	.010	.029
L	0.40	1.27	.016	.050
N	28		28	
a	0°		8°	

Package Drawing

VQFP32



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

Table of Contents

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

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 Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

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

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

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

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

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

4128A–8051–04/02 /xM