HOLTEK

Standard 8051 8-Bit Flash MCU

HT85F2260
HT85F2270
HT85F2280

Revision: V1.00 Date: January 15, 2015

www.holtek.com

Table of Contents

List of Tables

List of Figures

1 Features

CPU Features

- Operating Voltage:
  - f_{SYS}=3.6864MHz: 2.2V~5.5V
  - f_{SYS}=8MHz: 2.2V~5.5V
  - f_{SYS}=12MHz: 2.7V~5.5V
  - f_{SYS}=24MHz: 4.5V~5.5V
- Program Memory Capacity: 16K×8~64K×8
- Data Memory Capacity: 1280×8~2304×8
- High performance 1-T architecture: 8051
- Up to 32MIPS with 32MHz system clock at V_{DD}=5V
- 8051 compatible instruction set
- Flexible Power-down and wake-up functions to reduce power consumption
- Oscillator types:
  - External high frequency crystal
  - Internal high frequency RC
  - External low frequency crystal
  - Internal low frequency RC
- Multi-mode operation: Normal, Idle and Power-Down Modes
- Fully integrated internal 3.6864MHz oscillator requires no external components
- Internal PLL to multiply oscillator frequency up to 1~8 times for high speed system clock
- Watchdog Timer function
- Dual 16-bit data pointers with addition arithmetic operation

Peripheral Features

- Multi-channel 12-bit resolution A/D converter
- Single 12-bit D/A Converter
- Serial SPI Interface
- I^{2}C Interface
- Dual UART Interfaces
- Dual Comparator functions
- Up to 48 bidirectional I/O lines
- 16-bit Programmable Counter Array with 5 Capture/Compare Modules
- 16-bit Programmable Counter Array
- Single Time-Base functions for generation of fixed time interrupt signal
- Internal Temperature Sensor
- Low voltage reset function
- Low voltage detect function
- Package types: 48-LQFP and 64-LQFP

2 General Description

The HT85F22x0 series of devices are Flash Memory A/D type high performance 1-T architecture 8051-Based microcontrollers. Offering users the convenience of Flash Memory multi-programming features, these devices also include a wide range of functions and features.

Analog features include a multi-channel 12-bit A/D converter, a 12-bit D/A converter and dual comparator functions. Multiple timers provide timing, capture, event counter and programmable clock output functions. Communication with the outside world is catered for by including fully integrated SPI, I^2C and UART interface functions, popular interfaces which provide designers with a means of easy communication with external peripheral hardware. Protective features such as an internal Watchdog Timer, Low Voltage Reset and Low Voltage Detector and excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

A full choice of both internal and external high and low speed oscillators are provided with the internal oscillators requiring no external components for its implementation. A fully internal Phase Locked Loop and the ability to operate and switch dynamically between a range of operating modes using different clock sources gives users the ability to optimise microcontroller operation and minimize power consumption.

The inclusion of flexible I/O programming features, Time-Base functions along with many other features ensure that the device will find excellent use in applications such as electronic metering, environmental monitoring, handheld instruments, household appliances, electronically controlled tools, motor driving in addition to many others.

The HT85F22x0 series are Flash devices offering the advantages of easy and effective in-circuit program updates. In addition, an EV chip, HT85V2280, includes an OCDS (On-Chip Debug Support) interface for the In-Circuit Emulator.

3 Selection Table

Most features are common to all devices, the main feature distinguishing them are Program Memory and Data memory capacity, A/D channels, UART numbers and packages. The following table summarises the main features of each device.

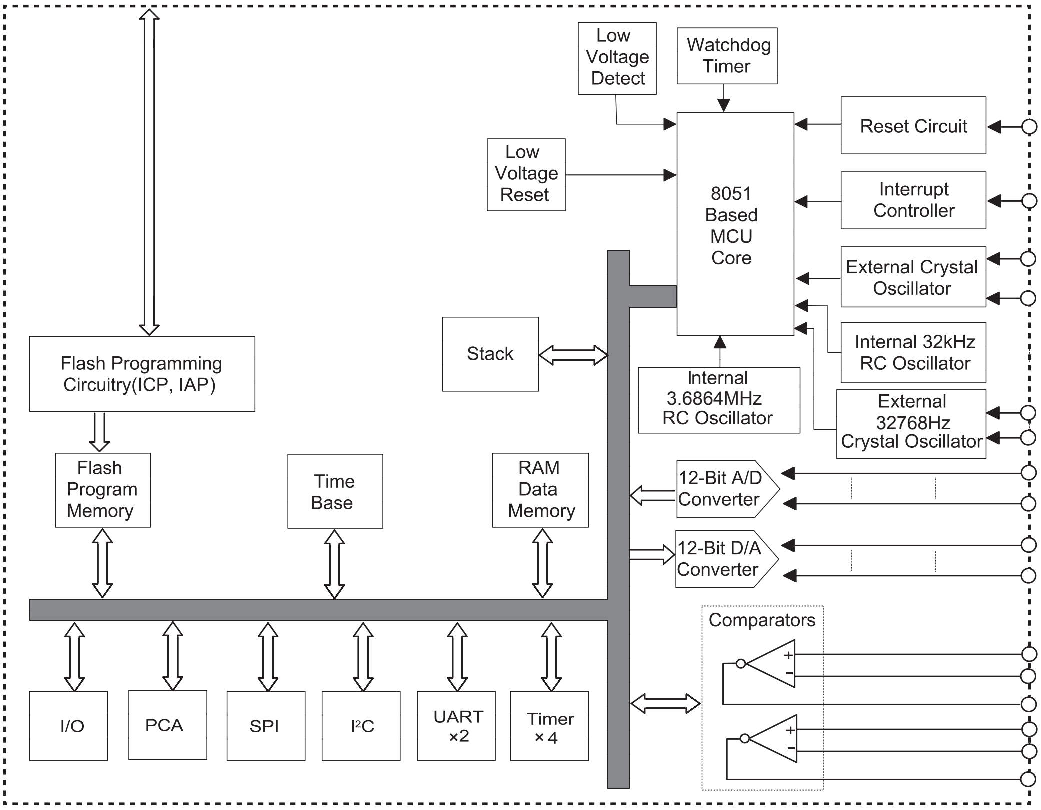
Part No.	V_{DD}	Program Memory	Data Memory	I/O	Ext. Interrupt	16-bit Timer	16-bit PCA	Time Base
HT85F2260	2.2V~5.5V	16K×8	1280×8	32	7	4	CCU×4	1
HT85F2270	2.2V~5.5V	32K×8	2304×8	48	7	4	CCU×4	1
HT85F2280	2.2V~5.5V	64K×8	2304×8	48	7	4	CCU×4	1

Part No.	A/D	D/A	Comparator	I^2C	SPI	UART	Temp. Sensor	package
HT85F2260	12-bit×7	12-bit×1	2	√	√	1	√	48LQFP
HT85F2270	12-bit×9	12-bit×1	2	√	√	2	√	48/64LQFP
HT85F2280	12-bit×9	12-bit×1	2	√	√	2	√	48/64LQFP

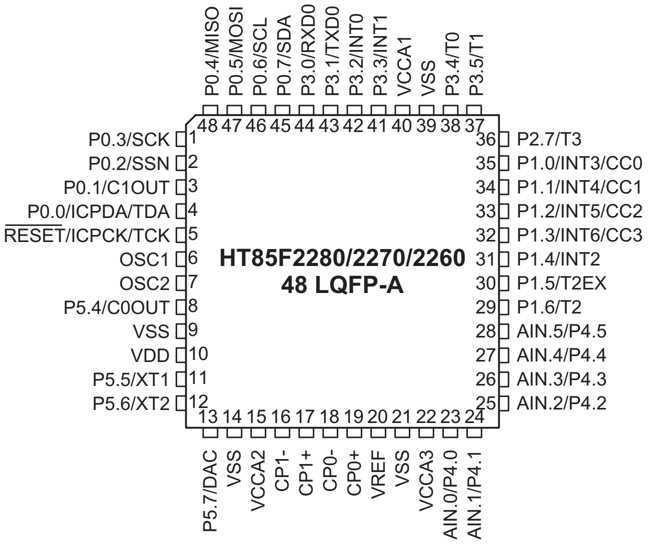
Note: CCU stands for Compare/Capture Unit.

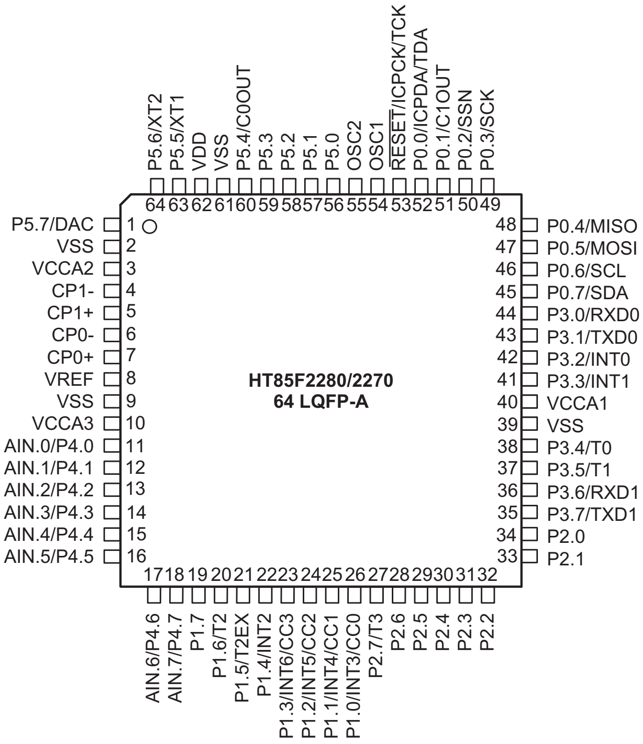
4 Block Diagram

The following block diagram illustrates the main functional blocks.



5 Pin Assignment





Note: 1. If the pin-shared pin functions have multiple outputs simultaneously, its pin names at the right side of the "/" sign can be used for higher priority.

2. For both the 48 LQFP-A and 64 LQFP-A packages, both real IC and OCDS EV IC share the same package.

6 Pin Descriptions

With the exception of the power pins, all pins on these devices can be referenced by their Port name, e.g. P0.0, P0.1 etc, which refer to the digital I/O function of the pins. However these Port pins are also shared with other function such as the Analog to Digital Converter, Serial Port pins etc. The function of each pin is listed in the following table, however the details behind how each pin is configured is contained in other sections of the datasheet.

Pin Name	Function	OPT	I/T	O/T	Description
P0.0/ICPDA/TDA	P0.0	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	ICPDA	—			ICP Data Input/Output
	TDA	—			Debug Data Input/Output
P0.1/C1OUT	P0.1	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	C1OUT	—	—	CMOS	Comparator 1 Output
P0.2/SSN	P0.2	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	SSN	—	ST	—	SPI Slave select Input
P0.3/SCK	P0.3	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	SCK	—	ST	CMOS	SPI Clock
P0.4/MISO	P0.4	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	MISO	—	ST	CMOS	SPI Master In Slave Out pin
P0.5/MOSI	P0.5	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	MOSI	—	ST	CMOS	SPI Master Out Slave In pin
P0.6/SCL	P0.6	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	SCL	—	—	NMOS	I^{2}C Clock
P0.7/SDA	P0.7	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up
	SDA	—	—	NMOS	I^{2}C Data
P1.0/INT3/CC0	P1.0	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT3	—	ST	—	External Interrupt 3 Input
	CC0	—	ST	CMOS	Compare/Capture input/output for PCA module 0
P1.1/INT4/CC1	P1.1	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT4	—	ST	—	External Interrupt 4 Input
	CC1	—	ST	CMOS	Compare/Capture input/output for PCA module 1
P1.2/INT5/CC2	P1.2	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT5	—	ST	—	External Interrupt 5 Input
	CC2	—	ST	CMOS	Compare/Capture input/output for PCA module 2

Pin Name	Function	OPT	I/T	O/T	Description
P1.3/INT6/CC3	P1.3	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT6	—	ST	—	External Interrupt 6 Input
	CC3	—	ST	CMOS	Compare/Capture input/output for PCA module 3
P1.4/INT2	P1.4	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT2	—	ST	—	External Interrupt 2 Input
P1.5/T2EX	P1.5	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	T2EX	—	ST	—	Timer 2 capture trigger
P1.6/T2	P1.6	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	T2	—	ST	CMOS	Timer 2 external input or Timer 2 programmable clock output
P1.7	P1.7	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
P2.0~P2.6	P2.0~P2.6	P2M0 P2M1	ST	CMOS	General purpose I/O. Register selected I/O mode
P2.7/T3	P2.7	P2M0 P2M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	T3	—	ST	—	Timer 3 External Input
P3.0/RXD0	P3.0	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	RXD0	—	ST	—	UART0 Receive Data Input
P3.1/TXD0	P3.1	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	TXD0	—	—	CMOS	UART0 Transmit Data Output
P3.2/INT0	P3.2	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT0	—	ST	—	External Interrupt 0 Input
P3.3/INT1	P3.3	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	INT1	—	ST	—	External Interrupt 1 Input
P3.4/T0	P3.4	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	T0	—	ST	—	Timer 0 External Input
P3.5/T1	P3.5	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	T1	—	ST	—	Timer 1 External Input
P3.6/RXD1	P3.6	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	RXD1	—	ST	—	UART1 Receive Data Input
P3.7/TXD1	P3.7	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	TXD1	—	—	CMOS	UART1 Transmit Data Output
P4.0/AIN.0	P4.0	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.0	—	AN	—	ADC Input Channel 0
P4.1/AIN.1	P4.1	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.1	—	AN	—	ADC Input Channel 1
P4.2/AIN.2	P4.2	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.2	—	AN	—	ADC Input Channel 2

Pin Name	Function	OPT	I/T	O/T	Description
P4.3/AIN.3	P4.3	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.3	—	AN	—	ADC Input Channel 3
P4.4/AIN.4	P4.4	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.4	—	AN	—	ADC Input Channel 4
P4.5/AIN.5	P4.5	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.5	—	AN	—	ADC Input Channel 5
P4.6/AIN.6	P4.6	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.6	—	AN	—	ADC Input Channel 6
P4.7/AIN.7	P4.7	P4M0 P4M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN.7	—	AN	—	ADC Input Channel 7
P5.0~P5.3	P5.0~P5.3	P5M0 P5M1	ST	CMOS	General purpose I/O. Register selected I/O mode
P5.4/C0OUT	P5.4	P5M0 P5M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	C0OUT	—	—	CMOS	Comparator 0 Output
P5.5/XT1	P5.5	P5M0 P5M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	XT1	—	LXT	—	Low Frequency Crystal Oscillator
P5.6/XT2	P5.6	P5M0 P5M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	XT2	—	—	LXT	Low Frequency Crystal Oscillator
P5.7/DAC	P5.7	P5M0 P5M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	DAC	—	—	CMOS	DAC Output
CP0-/CP0+	CP0-	—	AN	—	Comparator 0 Inverting Input
	CP0+	—	AN	—	Comparator 0 Non-Inverting Input
CP1-/CP1+	CP1-	—	AN	—	Comparator 1 Inverting Input
	CP1+	—	AN	—	Comparator 1 Non-Inverting Input
OSC1	OSC1	—	HXT	—	High Frequency Crystal Oscillator
OSC2	OSC2	—	—	HXT	High Frequency Crystal Oscillator
$\overline{\text{RESET}}$/ICPCK/TCK	$\overline{\text{RESET}}$	—	ST	—	RESET pin
	ICPCK	—	ST	—	ICP Clock Input
	TCK	—	ST	—	Debug Clock Input
VREF	VREF	—	AN	—	Reference Voltage for ADC/DAC
VDD	VDD	—	PWR	—	Positive Power supply for CORE
VCCA1	VCCA1	—	PWR	—	Positive Power supply for I/O pad
VCCA2	VCCA2	—	PWR	—	Positive Power supply for DAC
VCCA3	VCCA3	—	PWR	—	Positive Power supply for ADC
VSS	VSS	—	PWR	—	Negative Power supply

Note: I/T: Input type; O/T: Output type; ST: Schmitt Trigger input
OPT: Optional by configuration option (CO) or register option
PWR: Power; NMOS: NMOS output
CMOS: CMOS output; AN: Analog input pin
LXT: low frequency crystal oscillator; HXT: high frequency crystal oscillator

Where devices exist in more than one package type the table reflects the situation for the package with the largest number of pins. For this reason not all pins described in the table may exist on all package types.

7 Absolute Maximum Ratings

Supply Voltage	V_{SS}-0.3V to V_{DD}+6.0V
Input Voltage	V_{SS}-0.3V to V_{DD}+0.3V
Storage Temperature	-50°C to 125°C
Operating Temperature	-40°C to 85°C
I_{OL} Total	150mA
I_{OH} Total	-100mA
Total Power Dissipation	500mW

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

8 D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD1}	Operating Voltage (High Frequency Internal RC OSC)	—	f_{OSC}=f_{SYS}=3.6864MHz (PLL disabled)	2.2	—	5.5	V
V_{DD2}	Operating Voltage (Crystal OSC)	—	f_{OSC}=f_{SYS}=8MHz (PLL disabled)	2.2	—	5.5	V
V_{DD3}	Operating Voltage (PLL)	—	f_{OSC}=4MHz (Crystal OSC) f_{SYS}=12MHz (PLL × 3)	2.7	—	5.5	V
V_{DD4}	Operating Voltage (PLL)	—	f_{OSC}=4MHz (Crystal OSC) f_{SYS}=16MHz (PLL × 4)	3.3	—	5.5	V
V_{DD5}	Operating Voltage (PLL)	—	f_{OSC}=4MHz (Crystal OSC) f_{SYS}=24MHz (PLL × 6)	4.5	—	5.5	V
I_{DD1}	Operating Current (High Frequency Internal RC OSC)	3V	No load, f_{OSC}=f_{SYS}=3.6864MHz , (PLL disabled) ADC off, DAC off, WDT enable	—	5.0	8.0	mA
		5V		—	10.0	15.0	
I_{DD2}	Operating Current (Crystal OSC)	3V	No load, f_{OSC}=f_{SYS}=8MHz , (PLL disabled) ADC off, DAC off, WDT enable	—	6.0	8.5	mA
		5V		—	12.5	20	
I_{DD3}	Operating Current (PLL)	3V	No load, f_{OSC}=4MHz (Crystal OSC) f_{SYS}=12MHz (PLL × 3) ADC off, DAC off, WDT enable	—	8.0	12.0	mA
		5V		—	16	25	
I_{DD4}	Operating Current (PLL)	5V	No load, f_{OSC}=4MHz (Crystal OSC) f_{SYS}=16MHz (PLL × 4) ADC off, DAC off, WDT enable	—	20	30	mA
I_{DD5}	Operating Current (PLL)	5V	No load, f_{OSC}=4MHz (Crystal OSC) f_{SYS}=24MHz (PLL × 6) ADC off, DAC off, WDT enable	—	28	40	mA

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{STB1}	Stanby Current (Power-Down mode)(HIRC off, HXT off)	3V	No load, All peripherals off	—	—	1.5	μA
		5V		—	—	2.5	
I_{STB2}	Stanby Current (Idle) (HIRC off, HXT on)	3V	No load, f_{OSC}=4MHz (Crystal OSC) f_{SYS} off, ADC off, DAC off, LVD/LVR disable, WDT enable,	—	1.5	2.5	mA
		5V		—	3.5	5.0	
V_{IL1}	Input Low Voltage (except $\overline{RESET}$ pin)	—	quasi-bidirection mode	0	—	$0.2V_{DD}$	V
V_{IH1}	Input High Voltage (except $\overline{RESET}$ pin)	—	quasi-bidirection mode	$0.8V_{DD}$	—	V_{DD}	V
V_{IL2}	Input Low Voltage ($\overline{RESET}$ pin)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	Input High Voltage ($\overline{RESET}$ pin)	—	—	$0.9V_{DD}$	—	V_{DD}	V
I_{OL}	I/O Port Sink Current	2.2V	V_{OL}=0.4V	—	6.0	—	mA
		3.3V		—	9.0	—	
		5.0V		—	12.0	—	
I_{OH1}	I/O Port Source Current (push-pull mode for Ports 0, 1, 2, 3)	2.2V	V_{OH}=$0.9V_{DD}$	—	-1.0	—	mA
		3.3V		—	-2.0	—	
		5.0V		—	-4.0	—	
I_{OH2}	I/O Port Source Current (quasi-bidirection mode for Ports 0, 1, 2, 3, 4, 5)	2.2V	V_{OH}=$0.9V_{DD}$	—	-40	—	μA
		3.3V		—	-80	—	
		5.0V		—	-160	—	
I_{IL}	Logical 0 input current, Ports 0, 1, 2, 3, 4, 5 (quasi-bidirection mode)	5V	V_{IN}=0.4V	—	—	-50	μA
I_{TL}	Logical 1 to 0 transition Current, Ports 0, 1, 2, 3, 4, 5 (quasi-bidirection mode)	5V	V_{IN}=2.4V	—	—	-950	μA
I_{LI}	Input Leakage current, Ports 0, 1, 2, 3 (input mode)	5V	$0.45V<V_{IN}<V_{DD}-0.3$	—	—	±10	μA
V_{BG}	Bandgap reference with buffer voltage (for A/D type MCU Tiny Power IP)	—	—	-3%	1.1	+3%	V
I_{BG}	Additional Power Consumption if Reference with Buffer is used (for A/D type MCU)	—	—	—	200	300	μA
I_{LVR}	Additional Power Consumption if LVR is used (for Tiny Power IP)	3V	LVR enable	—	75	100	μA
		5V		—	75	100	
I_{LVD}	Additional Power Consumption if LVD is used (for Tiny Power IP)	3V	LVD enable	—	75	100	μA
		5V		—	75	100	
V_{LVR1}	Low Voltage Reset Voltage	—	LVR Enable, 2.1V select	-5%	2.1	+5%	V
V_{LVR2}			LVR Enable, 2.55V select		2.55		
V_{LVR3}			LVR Enable, 3.15V select		3.15		
V_{LVR4}			LVR Enable, 4.0V select		4.0		
V_{LVD1}	Low Voltage Detector Voltage	—	LVD Enable, 2.0V Select	-5%.	2.0	+5%	V
V_{LVD2}			LVD Enable, 2.2V Select		2.2		
V_{LVD3}			LVD Enable, 2.4V Select		2.4		
V_{LVD4}			LVD Enable, 2.7V Select		2.7		
V_{LVD5}			LVD Enable, 3.0V Select		3.0		
V_{LVD6}			LVD Enable, 3.3V Select		3.3		
V_{LVD7}			LVD Enable, 3.6V Select		3.6		
V_{LVD8}			LVD Enable, 4.2V Select		4.2		

9 A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS1}	System clock (Crystal OSC)	2.2V~5.5V	PLL Disable	0.4	—	8	MHz
		2.7V~5.5V		0.4	—	12	
		4.5V~5.5V		0.4	—	24	
f_{SYS2}	System clock (PLL)	4.5V~5.5V	Crystal OSC=4MHz, PLL Enable	4	—	32	MHz
f_{LIRC}	32kHz Internal RC oscillator	5V	Ta=25°C	-10%	32	+10%	kHz
		2.2V~5.5V	Ta=-40°C~85°C	-50%	32	+60%	
f_{HIRC}	3.6864MHz Internal RC oscillator	3V	Ta=25°C	-3%	3.6864	+3%	MHz
		5V	Ta=25°C	-3%	3.6864	+3%	
f_{TIMER}	Timer Input Frequency (T0~T3)	2.2V~5.5V	f_{SYS}=8MHz	0	—	2	MHz
		2.7V~5.5V	f_{SYS}=12MHz	0	—	3	
		4.5V~5.5V	f_{SYS}=24MHz	0	—	6	
t_{RES}	External Reset Minimum Low Pulse width	—	—	1	3.3	5	μs
t_{SST}	System start-up timer period (Power-up or wake-up from Power-Down mode when the main oscillator is off or system clock is switching between HXT and HIRC)	—	f_{SYS}=HXT or HIRC	—	1024	—	t_{SYS}
t_{RSTD}	System Reset Delay Time (LVR reset)	—	—	16	32	64	ms
t_{SRESET}	Software Reset Width to Reset	—	—	45	90	120	μs
t_{HTO}	HIRC Turn On Period	2.2V~5.5V	HIRC OFF → ON	—	200	—	μs
t_{INT}	External Interrupt Minimum Pulse Width	—	MCU is in Normal mode or Idle mode	—	4	—	t_{SYS}
t_{LVR}	Low Voltage Width to Reset	—	—	120	240	480	μs

10 ADC Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
AV_{DD}	A/D Converter Operating Voltage	—	—	2.7	—	5.5	V
V_{ADI}	A/D Converter Input Voltage	—	VREF available	0	—	V_{REF}	V
		—	VREF not available	0	—	AV_{DD}	
V_{REF}	A/D Converter Reference Voltage	—	—	2	—	AV_{DD}	V
DNL	Differential Non-linearity	—	AV_{DD}=5V V_{REF}=AV_{DD} t_{ADCK}=1μs	-2	—	+2	LSB
INL	Integral Non-linearity	—	AV_{DD}=5V V_{REF}=AV_{DD} t_{ADCK} =1μs	-4	—	+4	LSB
I_{ADC}	Additional Power Consumption if A/D Converter is used	3V	No load, t_{ADCK}=0.5μs	—	1.00	1.40	mA
		5V		—	1.30	2.00	mA
t_{ADCK}	A/D Converter Clock Period	—	—	0.5	—	10	μs
t_{ADC}	A/D Conversion Time (Include Sample and Hold Time)	—	12 bit ADC	—	16	—	t_{ADCK}
t_{ADS}	A/D Converter Sampling Time	—	—	—	4	—	t_{ADCK}
t_{ON2ST}	ADC on to ADC start	—	—	2	—	—	μs

11 DAC Electrical Characteristics

V_{DD}=3V, AV+=3.0V, V_{REF}=2.4V, no output load unless otherwise specified

Parameter	Test Conditions	Min.	Typ.	Max.	Units
Static Performance					
Resolution	—		12		bits
Integral Nonlinearity	—	—	±2	—	LSB
Differential Nonlinearity	—	—	—	±1	LSB
Offset Error	Data Word=0x014	—	±3	±30	mV
Gain Error	—	—	±20	±60	mV
Output Sink Current	—	—	300	—	μA
Output Short-Circuit Current	Data Word=0xFFF	—	15	—	mA

12 Comparator Electrical Characteristics

Ta=25°C

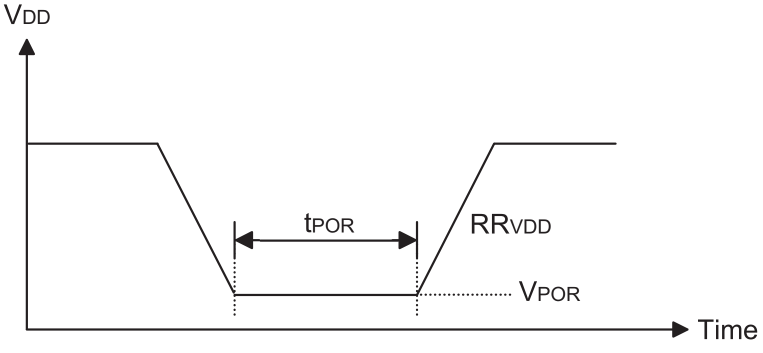
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
—	Comparator operating voltage	—	—	2.2	—	5.5	V
—	Comparator operating current	5V	LVDCR=00h, ADCR1=08h, i.e. select internal bandgap voltage output (x2) as VREFI	—	—	10	μA
—	Comparator power-down current	5V	Comparator disable	—	—	0.1	μA
V_{CMPOS}	Comparator input offset voltage	5V	—	-10	—	+10	mV
V_{HP1}	Positive Hysteresis 1	5V	CP(n)HP[1:0]=00b	—	0	1	mV
V_{HP2}	Positive Hysteresis 2	5V	CP(n)HP[1:0]=01b	3	6	10	mV
V_{HP3}	Positive Hysteresis 3	5V	CP(n)HP[1:0]=10b	6	13	20	mV
V_{HP4}	Positive Hysteresis 4	5V	CP(n)HP[1:0]=11b	12	25	40	mV
V_{HN1}	Negative Hysteresis 1	5V	CP(n)HN[1:0]=00b	—	0	1	mV
V_{HN2}	Negative Hysteresis 2	5V	CP(n)HN[1:0]=01b	3	6	10	mV
V_{HN3}	Negative Hysteresis 3	5V	CP(n)HN[1:0]=10b	6	13	20	mV
V_{HN4}	Negative Hysteresis 4	5V	CP(n)HN[1:0]=11b	12	25	40	mV
V_{CM}	Comparator common mode voltage range	—	—	V_{SS}	—	V_{DD}-1.4V	V
A_{OL}	Comparator open loop gain	—	—	60	80	—	dB
t_{PD}	Comparator response time	3V 5V	With 100mV overdrive	—	4	—	μs

Note: Measured with comparator one input pin at V_{CM}=(V_{DD}-1.4)/2 while the other pin input transition from V_{SS} to (V_{CM} +100mV) or from V_{DD} to (V_{CM} -100mV).

13 Power on Reset Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{POR}	V_{DD} Start Voltage to ensure Power-on Reset	—	—	—	—	100	mV
RR_{POR}	V_{DD} Rising Rate to ensure Power-on Reset	—	—	0.035	—	—	V/ms
t_{POR}	Minimum Time for V_{DD} stays at V_{POR} to ensure Power-on Reset	—	—	1	—	—	ms



14 System Architecture

A key factor in the high-performance features of the Holtek range of microcontrollers is attributed to their internal system architecture. The range of devices take advantage of the usual features found within 8051-based microcontrollers providing increased speed of operation and enhanced performance. The pipelining scheme is implemented in such a way that instruction fetching and instruction execution are overlapped, hence most instructions are effectively executed in one clock cycle, with the exception of branch or call instructions. Compared with classic MCU architecture, the 8051-based core runs at a much higher speed and with greatly reduced power consumption. An 8-bit wide ALU is used in practically all operations of the 8051 compatible instruction set. It carries out arithmetic operations, logic operations, rotation, increment, decrement, branch decisions, etc. The internal data path is simplified by moving data through the Accumulator and the ALU. Certain internal registers are implemented in the Data Memory and can be directly or indirectly addressed. The simple addressing methods of these registers along with additional architectural features ensure that a minimum of external components is required to provide a functional I/O control system with maximum reliability and flexibility.

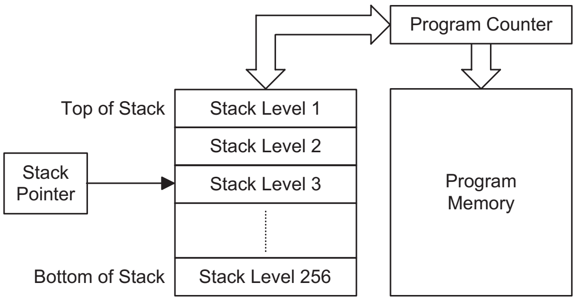
15 Program Counter

During program execution, the Program Counter is used to keep track of the address of the next instruction to be executed. It is automatically incremented by one each time an instruction is executed except for instructions, such as "JMP" or "CALL" that demand a jump to a non-consecutive Program Memory address.

When executing instructions requiring jumps to non-consecutive addresses such as a jump instruction, a subroutine call, interrupt or reset, etc., the microcontroller manages program control by loading the required address into the Program Counter. For conditional skip instructions, once the condition has been met, the next instruction, which has already been fetched during the present instruction execution, is discarded and a dummy cycle takes its place while the correct instruction is obtained.

16 Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack is located in the 256 byte Data memory; therefore, the depth can be extended up to 256 levels. The activated level is indexed by the Stack Pointer, SP, and is neither readable nor writeable. At a subroutine call or interrupt acknowledge signal, the contents of the Program Counter are pushed onto the stack. At the end of a subroutine or an interrupt routine, signaled by a return instruction, RET or RETI, the Program Counter is restored to its previous value from the stack. After a device reset, the Stack Pointer will point to the location 0x07, the top of the stack. Note that if the data memory has been used as the stack area, it should not be used as general purpose Data RAM.



Stack Block Diagram

If the stack is full and an enabled interrupt takes place, the interrupt request flag will be recorded but the acknowledge signal will be inhibited. When the Stack Pointer is decremented, by RET or RETI, the interrupt will be serviced. This feature prevents stack overflow allowing the programmer to use the structure more easily. However, when the stack is full, a CALL subroutine instruction can still be executed which will result in a stack overflow. Precautions should be taken to avoid such cases which might cause unpredictable program branching.

17 Arithmetic and Logic Unit – ALU

The arithmetic-logic unit or ALU is a critical area of the microcontroller that carries out arithmetic and logic operations of the instruction set. Connected to the main microcontroller data bus, the ALU receives related instruction codes and performs the required arithmetic or logical operations after which the result will be placed in the specified register. As these ALU calculation or operations may result in carry, borrow or other status changes, the status register will be correspondingly updated to reflect these changes. The ALU supports the following functions:

- Arithmetic operations: ADD, ADDC, SUBB, DA, MUL, DIV
- Logic operations: ANL, ORL, XRL, CLR, CPL
- Rotation: RL, RLC, RR, RRC, SWAP
- Increment and Decrement: INC, DEC
- Branch decision: JC, JNC, JB, JNB, JBC, ACALL, LCALL, RET, RETI, AJMP, SJMP, JMP, JZ, JNZ, CJNE, DJNZ

18 Flash Program Memory

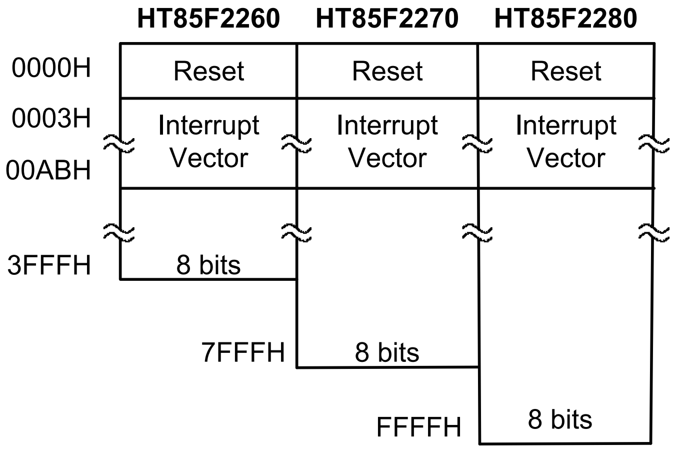
The Program Memory is the location where the user code or program is stored. For these devices the Program Memory is Flash type, which means it can be programmed and re-programmed a large number of times, allowing the user the convenience of code modification on the same device. By using the appropriate programming tools, these Flash devices offer users the flexibility to conveniently debug and develop their applications while also offering a means of field programming and updating.

Structure

The Program Memory has a capacity from 16K×8 to 64K×8. The Program Memory is addressed by the Program Counter and also contains data, table information and interrupt entries. Table data, which can be setup in any location within the Program Memory, is addressed by a separate table pointer register.

Special Vectors

Within the Program Memory, certain locations are reserved for the reset and interrupts. The location 000H is reserved for use by the device reset for program initialisation. After a device reset is initiated, the program will jump to this location and begin execution.



Program Memory Structure

In-Circuit Programming – ICP

The provision of Flash type Program Memory provides the user with a means of convenient and easy upgrades and modifications to their programs on the same device. As an additional convenience, Holtek has provided a means of programming the microcontroller in-circuit using a four-line serial interface. This provides manufacturers with the possibility of manufacturing their circuit boards complete with a programmed or un-programmed microcontroller, and then programming or upgrading the program at a later stage. This enables product manufacturers to easily keep their manufactured products supplied with the latest program releases without removal and re-insertion of the device.

The Holtek Flash MCU to Writer Programming Pin correspondence table is as follows:

Holtek Writer Pins	MCU Programming Pins	Function
ICPDA	P0.0/ICPDA	Programming Serial Data/Address
ICPCK	$\overline{\text{RESET}}$/ICPCK	Programming Serial Clock
VDD	VDD	Power Supply
VSS	VSS	Ground

The Program Memory can be programmed serially in-circuit using the interface on pins ICPDA and ICPCK. Data is downloaded and uploaded serially on a single pin with an additional line for the clock. Two additional lines are required for the power supply. The technical details regarding the in-circuit programming of the device are beyond the scope of this document and will be supplied in supplementary literature. The Flash Program Memory Read/Write function is implemented using a series of registers.

On-Chip Debug Support – OCDS

An EV chip, HT85V2280, is provided which includes all the HT85F2280 functions as well as an “On-Chip Debug” interface for emulation of the HT85F2280/2270/2260 devices. To minimise the difference between the real IC (the volume-production version) and the EV chip (the device with the debug interface), a protocol converter is implemented to translate the external 2-wire connections (TCK and TDA) into 4 internal JTAG signals (TCK, TMS, TDI, and TDO) and vice versa. Users can use the EV chip device to emulate the real chip device behavior by connecting the TDA and TCK pins to the related Holtek development tools. The TDA pin is the OCDS Data/Address input/output pin while the TCK pin is the OCDS clock input pin. When users use the EV chip for debugging, other functions which are shared with the TDA and TCK pins in the actual MCU device will have no effect in the EV chip. However, the two OCDS pins which are pin-shared with the ICP programming pins are still used as the Flash Memory programming pins for ICP. For a more detailed OCDS description, refer to the corresponding user’s guide.

In-Application Programming – IAP

An In-Application Programming interface is provided to allow the end user's application to erase and reprogram the user code memory. No extra code memory block (bootloader) is required to update the firmware or non-volatile data. Firmware for the IAP and the code memory to be updated are physically on the same IP. Users could update firmware or non-volatile data except for the sector where IAP is located and run. A firmware library is used to provide APIs for flash programming.

Flash Program Memory Resisters

With regard to the Flash Program Memory registers, there are three address registers, one 8-bit data register and three control registers, located in the Special Function Registers. Read and Write operations to the Flash memory are carried out in 8-bit data operations using the address and data registers and the control registers. The address registers are named FMAR0, FMAR1 and FMAR2, the data register is named FMDR, and the three control registers are named FMKEY, FMCR and FMSR. As these registers are located in Special Function Register area, they can be directly accessed in the same was as any other Special Function Register.

Program Memory Register List

Name	Bit							
	7	6	5	4	3	2	1	0
FMAR0	FADDR7	FADDR6	FADDR5	FADDR4	FADDR3	FADDR2	FADDR1	FADDR0
FMAR1	FADDR15	FADDR14	FADDR13	FADDR12	FADDR11	FADDR10	FADDR9	FADDR8
FMAR2	INBLK	FADDR22	FADDR21	FADDR20	FADDR19	FADDR18	FADDR17	FADDR16
FMDR	FDAT7	FDAT6	FDAT5	FDAT4	FDAT3	FDAT2	FDAT1	FDAT0
FMKEY	FMKEY7	FMKEY6	FMKEY5	FMKEY4	FMKEY3	FMKEY2	FMKEY1	FMKEY0
FMCR	FMCR.7	FMCR.6	—	—	—	FMCR.2	FMCR.1	FMCR.0
FMSR	UNLOCK	—	—	—	FMPF	FMSEF	FMBF	FMBUSY

FMAR0 Register – Flash Program Memory Address Register 0
SFR Address: FAh

Bit	7	6	5	4	3	2	1	0
Name	FADDR7	FADDR6	FADDR5	FADDR4	FADDR3	FADDR2	FADDR1	FADDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 Flash Program Memory address
Flash Program Memory address bit 7~bit 0

FMAR1 Register – Flash Program Memory Address Register 1
SFR Address: FBh

Bit	7	6	5	4	3	2	1	0
Name	FADDR15	FADDR14	FADDR13	FADDR12	FADDR11	FADDR10	FADDR9	FADDR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 Flash Program Memory address
Flash Program Memory address bit 15~bit 8

FMAR2 Register – Flash Program Memory Address Register 2
SFR Address: FCh

Bit	7	6	5	4	3	2	1	0
Name	INBLK	FADDR22	FADDR21	FADDR20	FADDR19	FADDR18	FADDR17	FADDR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **INBLK:** Flash memory access block selection
0: Main Flash program memory area
1: Information block area

Bit 6~0 Flash Program Memory address
Flash Program Memory address bit 22~bit 16

FMDR Register – Flash Program Memory Data Register
SFR Address: FDh

Bit	7	6	5	4	3	2	1	0
Name	FDAT7	FDAT6	FDAT5	FDAT4	FDAT3	FDAT2	FDAT1	FDAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 Flash Program Memory Data register
Flash Program Memory Data bit 7~bit 0

FMCR Register – Flash Program Memory Control Register
SFR Address: F8h

Bit	7	6	5	4	3	2	1	0
Name	FMCR.7	FMCR.6	—	—	—	FMCR.2	FMCR.1	FMCR.0
R/W	R/W	R/W	—	—	—	R/W	R/W	R/W
POR	0	1	—	—	—	0	0	0

Bit 7 **FMCR.7:** Flash Memory Read/Write/Erase enable control bit
0: Disable
1: Enable
As this bit is cleared automatically by hardware soon after a command is initiated, when the MCU reads this bit it will always obtain a zero value.

Bit 6 **FMCR.6:** Flash Memory Byte Write/Page Erase control bit
0: For an un-written byte (0xFF) within a page, a write operation is allowed. But for those written bytes (except for 0xFF), a re-write operation is prohibited to avoid Flash errors. The writing time is shorter.
1: Before the main program executes a byte write operation, a page erase operation is automatically executed. Any location within the page is then rewritable, but the write time is longer. Note that the security bytes 00h~1Fh in the ID block page 0 can only be written once.

Bit 5~3 Unimplemented, read as "0"

Bit 2 **FMCR.2:** Flash Memory Page Erase control bit
0: Disable
1: Enable
This bit should be cleared manually.

Bit 1 **FMCR.1:** Flash Memory Byte Read control bit
0: Disable
1: Enable
This bit should be cleared manually.

Bit 0 **FMCR.0:** Flash Memory Byte Write control bit
0: Disable
1: Enable
This bit should be cleared manually.

FMKEY Register – Flash Program Memory Unlock Key Data Register
SFR Address: F9h

Bit	7	6	5	4	3	2	1	0
Name	FMKEY7	FMKEY6	FMKEY5	FMKEY4	FMKEY3	FMKEY 2	FMKEY 1	FMKEY 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 Flash Memory Unlock Key Data register Unlock Data bits 7~bit 0

The FMKEY register is the Flash Memory Unlock key data register. If a correct key data sequence has been written into this register, the Flash memory will release its locked status; otherwise, the Flash memory will remain in its locked status. The correct sequence to be written is 55H, AAH, 00H and then FFH. It is recommended to write the key data sequence to the FMKEY register in four consecutive instructions. When the program memory is in an unlocked status, writing any data to the FMKEY register will result in the program memory being locked again. If there is no need to update the program memory, it's strongly recommended to lock the program memory at all times.

FMSR Register – Flash Program Memory Status Register
SFR Address: E2h

Bit	7	6	5	4	3	2	1	0
Name	UNLOCK	—	—	—	FMPF	FMSEF	FMBF	FMBUSY
R/W	R	—	—	—	R	R	R	R
POR	0	—	—	—	0	0	0	0

Bit 7 **UNLOCK:** Flash memory Control Registers Unlock flag
0: Indicated Flash Memory Controller is locked
1: Indicated Flash Memory Controller is unlocked

Bit 6~4 Unimplemented, read as "0"

Bit 3 **FMPF:** Flash Memory Controller Procedure flag
0: The Flash Memory Controller Procedure Flag is cleared to 0 if FMSEF=1, or if FMBF=1 or if the IAP Procedure has ended.
1: Flash Memory Controller Procedure is corrected

Bit 2 **FMSEF:** Flash Memory Controller Security Error Flag
0: Manipulation of Flash Memory does not violate the security rules
1: Manipulation of Flash Memory violates the security rules

After a flash memory manipulation, this bit must be checked to determine if the Flash Memory manipulation has violated the security rules or not.

Bit 1 **FMBF:** Flash Memory Controller Break Flag
0: Manipulation of Flash Memory does not violate the security rules or lock rules or FMCR mode change
1: Manipulation of Flash Memory violates the security rules or lock rules or FMCR mode change

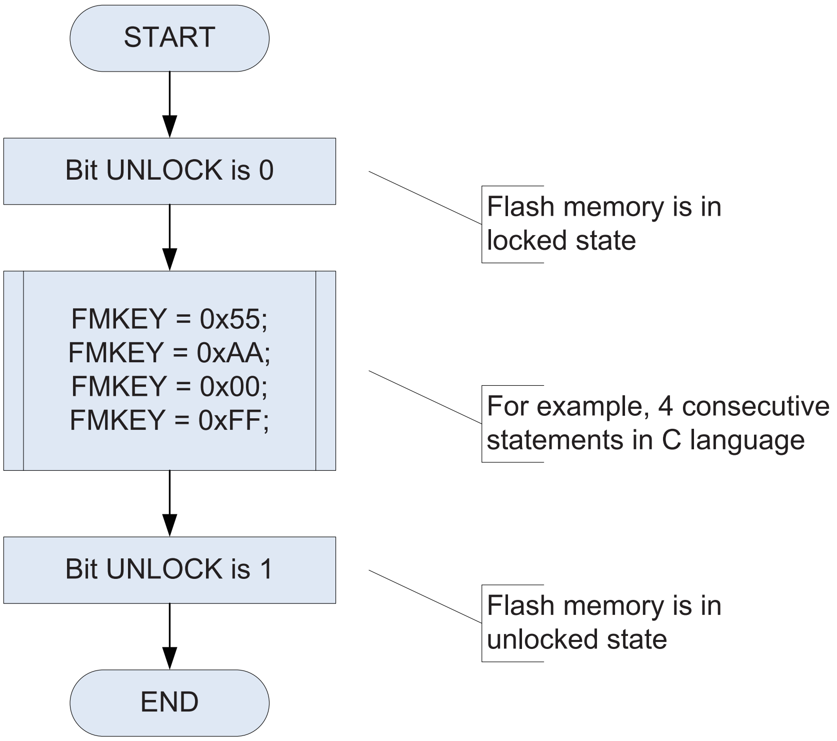
Bit 0 **FMBUSY:** Flash Memory Controller Status indication bit
0: Not erasing or rewriting
1: Busy

Flash Memory Read/Write Operations

The flash memory can be read from and written to using register operations. To ensure protection of application data certain protection measures have to be first carried out before any read and write operations are executed on the Flash Memory.

Unlocking the Flash Memory

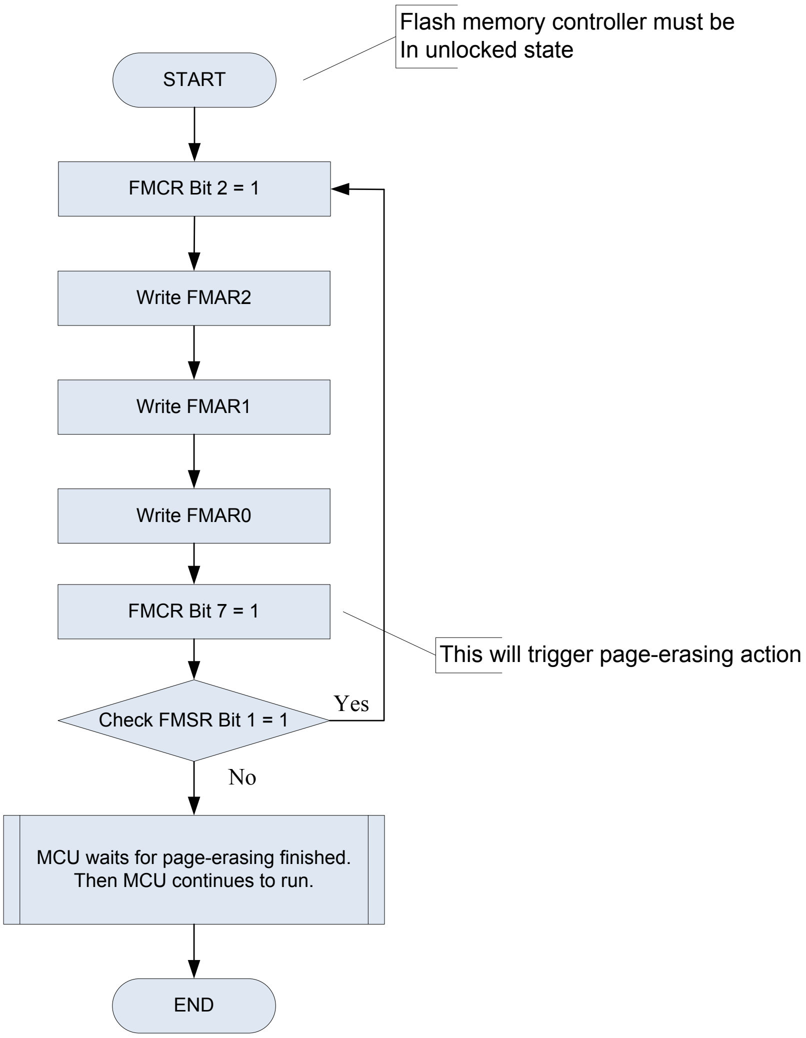
Before writing data to the Flash Memory it must first be unlocked. This is implemented by writing a correct data sequence to the Flash Memory Unlock key register, FMKEY. It is recommended to write the data sequence to the FMKEY register in 4 consecutive instructions. The following flowchart illustrates the unlock procedure.


START
Bit UNLOCK is 0
Flash memory is in locked state
FMKEY = 0x55;
FMKEY = 0xAA;
FMKEY = 0x00;
FMKEY = 0xFF;
For example, 4 consecutive statements in C language
Bit UNLOCK is 1
Flash memory is in unlocked state
END

Unlock Procedure Flowchart

Page Erase Operation

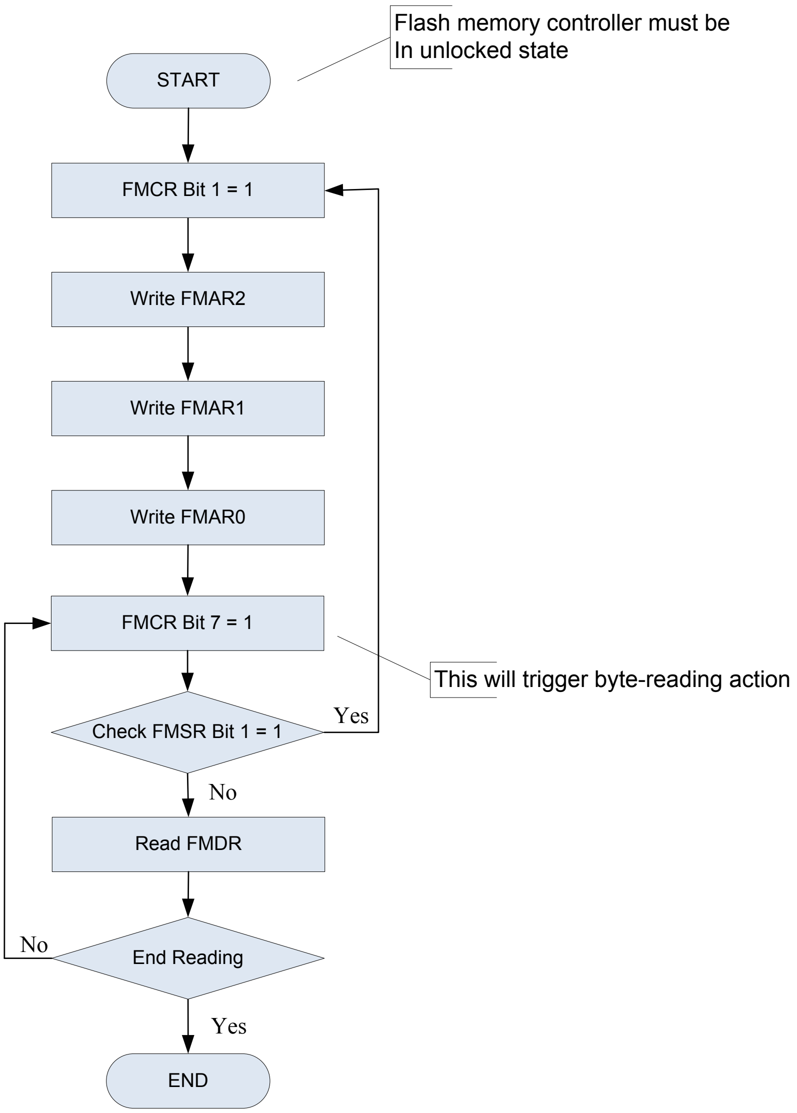
The Flash Memory must be first unlocked before implementing a page erase procedure. The flash memory address is setup using the control registers, FMAR0, FMAR1 and FMAR2. The Flash Memory Page Erase function is selected by the control bit, FMCR.2, in the FMCR register. Setting the FMCR.7 bit high will start the Page Erase procedure. When the procedure has finished, the MCU will continue to run automatically. The following flowchart illustrates the Page Erase procedure.


Flash memory controller must be
In unlocked state
START
FMCR Bit 2 = 1
Write FMAR2
Write FMAR1
Write FMAR0
FMCR Bit 7 = 1
This will trigger page-erasing action
Check FMSR Bit 1 = 1
Yes
No
MCU waits for page-erasing finished.
Then MCU continues to run.
END

Page Erase Flowchart

Byte Read Operation

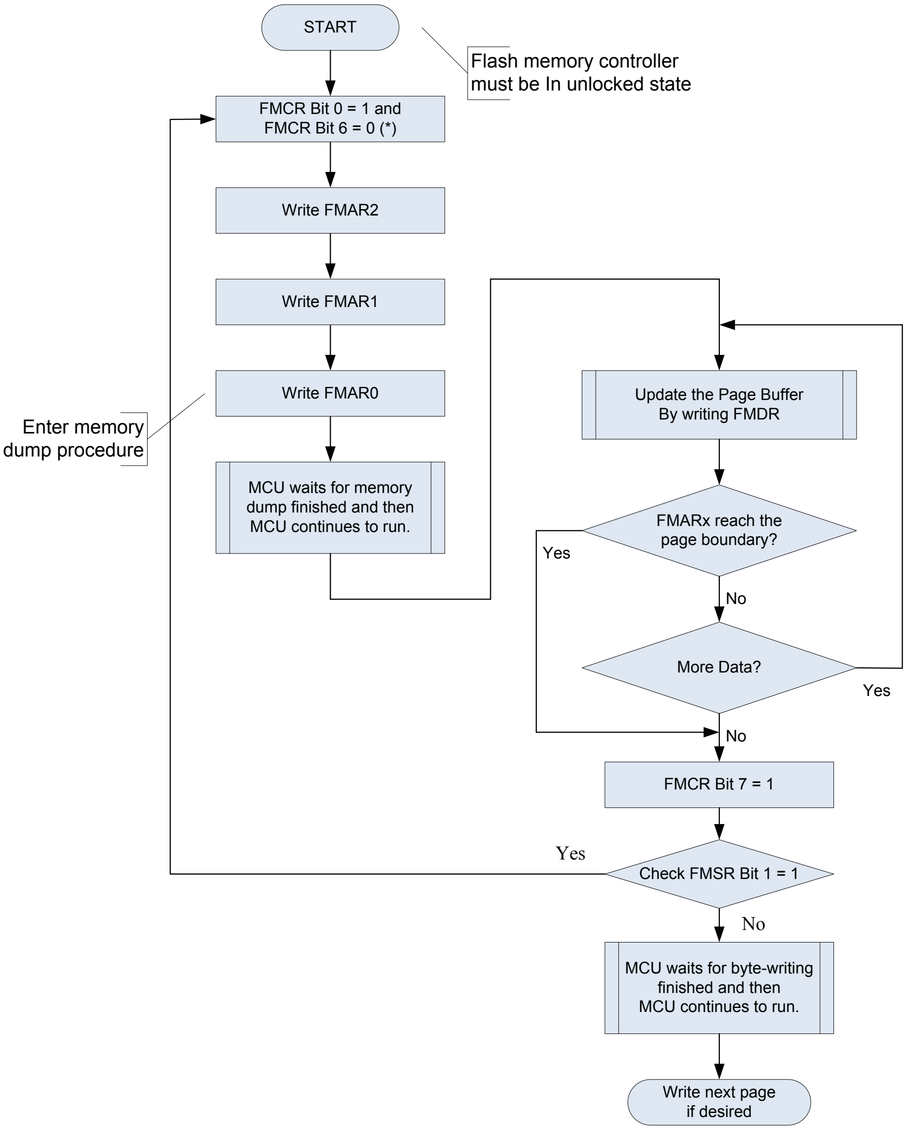
The Flash memory must be first unlocked before implementing a byte read procedure. The flash memory address is setup using the control registers, FMAR0, FMAR1 and FMAR2. The Flash Memory Page Read function is selected by the control bit, FMCR.1, in the FMCR register. When the FMCR.7 bit is set high the Byte Read procedure will be initiated. When the procedure is ready, the MCU will continue to run automatically. The following flowchart illustrates the Byte Read procedure.



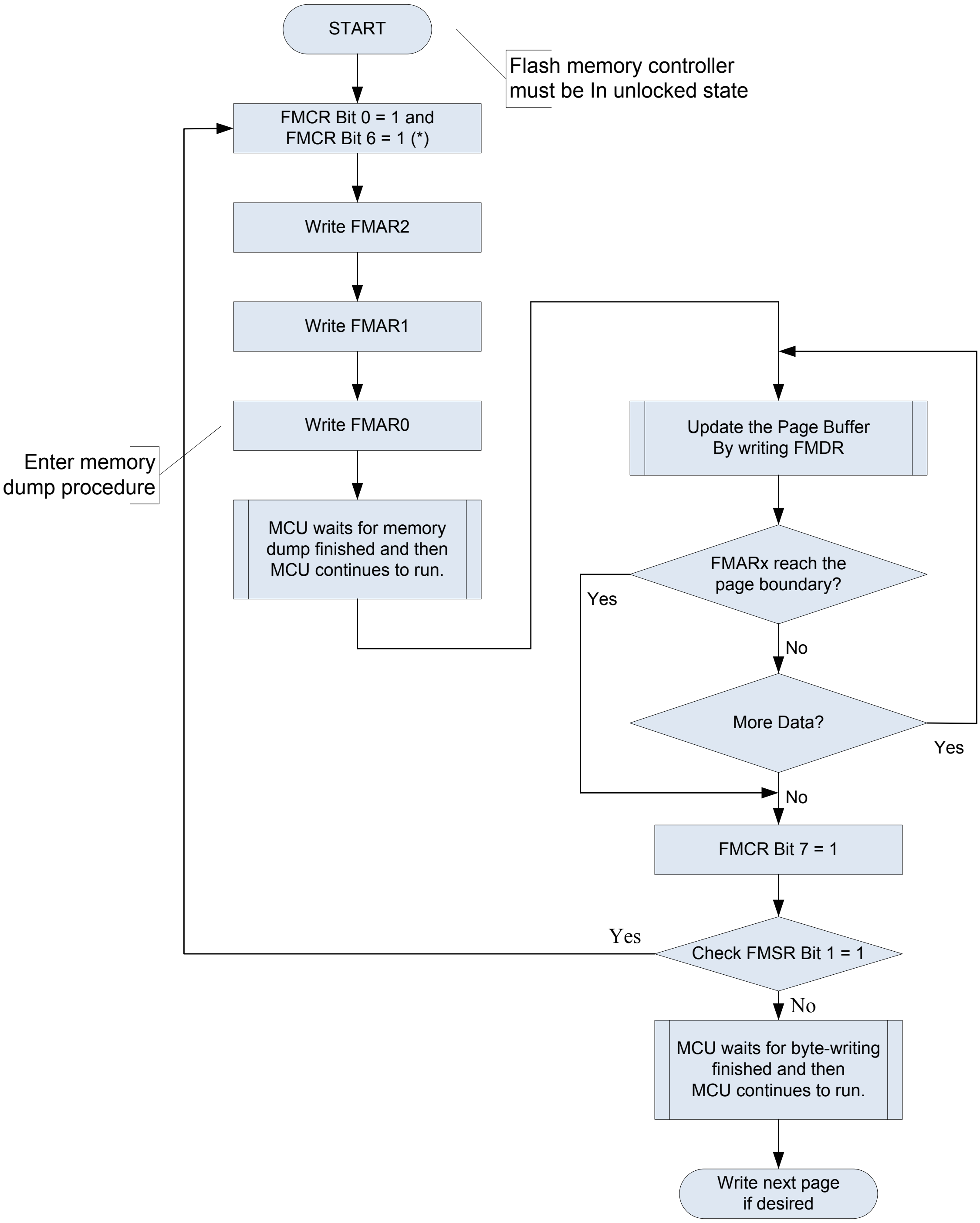
Byte Read Flowchart

Byte Write Operation

The Flash Memory must be first unlocked before implementing a Byte Write procedure. The first step is to assign the target memory page and erase it. Refer to the Page Erase Operation section for details. The Flash Memory Byte Write function is controlled by the control bits, FMCR.0 and FMCR.6, in the FMCR register. Data is first written into the FMDR register to update the Page Buffer. The Flash memory will check if the memory address has reached the page boundary. If the boundary has been reached or there is no more data, then set the FMCR.0 bit to high to enable the Byte Write function. When the FMCR.7 bit is set high the Byte Write procedure will be executed. When the procedure is ready, the MCU will continue to run automatically. The following flowchart illustrates the Byte Write procedure.


START
Flash memory controller must be In unlocked state
FMCR Bit 0 = 1 and FMCR Bit 6 = 0 (*)
Write FMAR2
Write FMAR1
Write FMAR0
Enter memory dump procedure
MCU waits for memory dump finished and then MCU continues to run.
Update the Page Buffer By writing FMDR
FMARx reach the page boundary?
Yes
No
More Data?
Yes
No
FMCR Bit 7 = 1
Yes
Check FMSR Bit 1 = 1
No
MCU waits for byte-writing finished and then MCU continues to run.
Write next page if desired

Byte Write Flowchart (FMCR.0=1, FMCR.6=0)


START
Flash memory controller must be In unlocked state
FMCR Bit 0 = 1 and FMCR Bit 6 = 1 (*)
Write FMAR2
Write FMAR1
Write FMAR0
Enter memory dump procedure
MCU waits for memory dump finished and then MCU continues to run.
Update the Page Buffer By writing FMDR
FMARx reach the page boundary?
Yes
No
More Data?
Yes
No
FMCR Bit 7 = 1
Yes
Check FMSR Bit 1 = 1
No
MCU waits for byte-writing finished and then MCU continues to run.
Write next page if desired

Byte Write Flowchart (FMCR.0=1, FMCR.6=1)

Program Memory Protection

The flash program memory is partitioned into 2 memory blocks. One is the main memory block and the other is the ID block. The ID block size is 256 bytes and is used to setup the protected sectors. This memory protection function is used to protect the Program Memory from improper Program, Erase or Read operations. The flash program memory is divided into several sectors related to the memory size. Each sector has a capacity of 4K bytes. The memory protection function is implemented by register control. If a value, with the exception of 0FFH, is written into the corresponding control register, the corresponding sector program memory protection function will be enabled. This program memory sector will then be unable to be programmed, erased or read by corresponding instructions. In this way, the user can select which block of the flash memory is to be protected.

Memory Protection Control Bytes

The protection of program code memory is categorised to two types: Security Type 1 and Security Type 2.

■ Security Type 1

For the HT85F2280 device, the inhibit bytes SECURITY1[0:15] are located at the address 0x00~0x0F of the ID block page 0. If a value, with the exception of 0FFH, is written into these bytes, the sectors corresponding to SECURITY1[0:15] cannot be programmed, erased or read by the ICP. For the IAP program, when in the OCDS mode, any sector N with a security mechanism can be protected from being programmed, erased or read by the OCDSINSTR instruction. But when in the main program, all sector N with security or not, can be programmed, erased or read by the IAP. For the MOVC instructions, any sector N with security mechanism cannot be read by the ocdsinstr instruction when in the OCDS mode, but still can be read by MOVC instructions when in the main program. Since these bytes can only be written once, to release the respective sectors in the unprotected mode, the device must be erased.

The following table illustrates the protection status when in the OCDS/ICP/IAP/MOVC modes when the SECURITY1[0:15] bytes are written with a value other than 0FFH:

SECURITY1[N] N=0~15		Program	Erase	Read	Protect Sector #	Remove Protection
ICP		X	X	N/A(1)	N	Erase All
IAP	OCDS(5)	X(3)	X	X(4)	N	Erase All
	Main Program	O(3)	O	O	N	Erase All
MOVC	OCDS(5)	N/A(2)		X(4)	N	Erase All
	Main Program	N/A(2)		O	N	Erase All

Note: (1) "N/A" means no path to read ROM code.
(2) "N/A" means none of these functions.
(3) "X" stands for inhibited; "O" stands for enabled.
(4) If a read operation is inhibited, reading the Flash will return a fixed Flash code of 00H.
(5) When in the OCDS mode, only the OCDSINSTR instruction has the security protection mechanism.

- Security Type 2

For the HT85F2280 device, the inhibit bytes SECURITY2[0:15] are located at the addresses 0x10~0x1F of the ID block page 0. If a value, with the exception of 0FFH, is written into these bytes, the sectors corresponding to SECURITY2[0:15] cannot be programmed, erased or read when in any mode. Since these bytes can only be written once, to release the respective sectors in the unprotected mode, the device must be erased.

The following table illustrates the protection state in the OCDS/ICP/IAP/MOVC modes when the SECURITY2[0:15] bytes are written with a value other than 0FFH:

SECURITY2[N] N=0~15		Program	Erase	Read	Protect Sector #	Remove Protection
ICP		X	X	N/A[1]	N	Erase All
IAP	OCDS[5]	X[3]	X	X[4]	N	Erase All
	Main Program	X[3]	X	X[4]	N	Erase All
MOVC	OCDS[5]	N/A[2]		X[4]	N	Erase All
	Main Program	N/A[2]		X[4]	N	Erase All

Note: (1) "N/A" means no path to read ROM code.
(2) "N/A" means none of these functions.
(3) "X" stands for inhibited; "O" stands for enabled.
(4) If a read operation is inhibited, reading to the Flash will return a fixed Flash code of 00H.
(5) When in the OCDS mode, only the OCDSINSTR instruction has the security protection mechanism.

The following tables illustrate the corresponding address ID sectors and the inhibited bytes.

HT85F2260 Program Memory Contents

The HT85F2260 program memory is divided into 4 sectors, each with a capacity of 4k bytes.

Page	Address	Description
0	0x00~0x03	SECURITY1[0]~SECURITY1[3]
	0x04~0x0F	Not used
	0x10~0x13	SECURITY2[0]~SECURITY2[3]
	0x14~0x1F	Not used
	0x20~0x6F	Reserved
	0x70~0x7F	Reserved
1	0x80~0x83	Reserved
	0x84~0x8F	Reserved
	0x90~0x93	Reserved
	0x94~0x9F	Reserved
	0xA0~0xEF	Reserved
	0xF0~0xFF	Reserved

HT85F2270 Program Memory Contents

The HT85F2270 program memory is divided into 8 sectors, each with a capacity of 4k bytes.

Page	Address	Description
0	0x00~0x07	SECURITY1[0]~SECURITY1[7]
	0x08~0x0F	Not used
	0x10~0x17	SECURITY2[0]~SECURITY2[7]
	0x18~0x1F	Not used
	0x20~0x6F	Reserved
	0x70~0x7F	Reserved
1	0x80~0x87	Reserved
	0x88~0x8F	Reserved
	0x90~0x97	Reserved
	0x98~0x9F	Reserved
	0xA0~0xEF	Reserved
	0xF0~0xFF	Reserved

HT85F2280 Program Memory Contents

The HT85F2280 program memory is divided into 16 sectors, each with a capacity of 4k bytes.

Page	Address	Description
0	0x00~0x0F	SECURITY1[0]~SECURITY1[15]
	0x10~0x1F	SECURITY2[0]~SECURITY2[15]
	0x20~0x6F	Reserved
	0x70~0x7F	Reserved
1	0x80~0x8F	Reserved
	0x90~0x9F	Reserved
	0xA0~0xEF	Reserved
	0xF0~0xFF	Reserved

Security Bytes

Name	Description
SECURITY1[N]	Sector N Program/Erase Inhibited Bytes 0xFF: unprotected Else: protected
SECURITY2[N]	Sector N Access Inhibited Bytes 0xFF: unprotected Else: protected

Note: N=0~15

These two types of flash memory inhibited bytes, SECURITY1[N] and SECURITY2[N], are used for Program Memory protection. However, the SECURITY2[N] bytes have the higher priority. If data has be written to the SECURITY2[N] bytes, the corresponding sectors will be protected and cannot be read from or written to, no matter what data is in the SECURITY1[N] bytes. Note that the Flash Memory protect function will not affect the instruction fetched by the MCU core. The accompanying table illustrates the inhibited bytes priority.

SECURITY2[N]	SECURITY1[N]	Privilege
0FFH	0FFH	Program Sector N is not protected Can be erased and programmed. Can be read by flash control registers related to the IAP and OCDS[(note)] and the MOVC instructions.
0FFH	Other values except 0FFH	Sector N is inhibited from Programming/Erasing Can not be erased and programmed by the ICP or flash control registers related to the OCDS[(note)]. Can be erased and programmed by flash control registers related to the IAP. Can be read by flash control registers related to the IAP and the MOVC instructions.
Other values except 0FFH	X	Sector N is inhibited from Programming/Erasing/Accessing (instruction fetch is still allowed) Can not be erased and programmed by the ICP or flash control registers related to the IAP and OCDS[(note)]. Can not be read by the ICP or flash control registers related to the IAP and OCDS[(note)] and the MOVC instructions.

Note: Here “OCDS” stands for executing OCDSINSTR instruction when in the OCDS mode.

19 RAM Data Memory

The Data Memory is a volatile area of 8-bit wide RAM internal memory and is the location where temporary information is stored. Divided into several sections, the first of these is an area of RAM where special function registers are located. These registers have fixed locations and are necessary for correct operation of the devices. Many of these registers can be read and written to directly under program control, however, some remain protected from user manipulation. The second area of Data Memory is reserved for general purpose use. All locations within this area are read and write accessible under program control. The Data Memory also includes the Bit-Addressable Space and four Register Banks.

Structure

The Data Memory is subdivided into two blocks, Internal Data RAM (IDATA) and On-Chip External Data RAM (XDATA), which are implemented in 8-bit wide RAM. The IDATA is subdivided into two sections, known as the Upper section and the Lower section. The Upper section includes two blocks, the Special Function Registers, SFR, and the 128-byte General Purpose RAM. The Special Function Register can be accessed using direct addressing methods while the 128-byte General Purpose RAM must be accessed using indirect addressing methods.

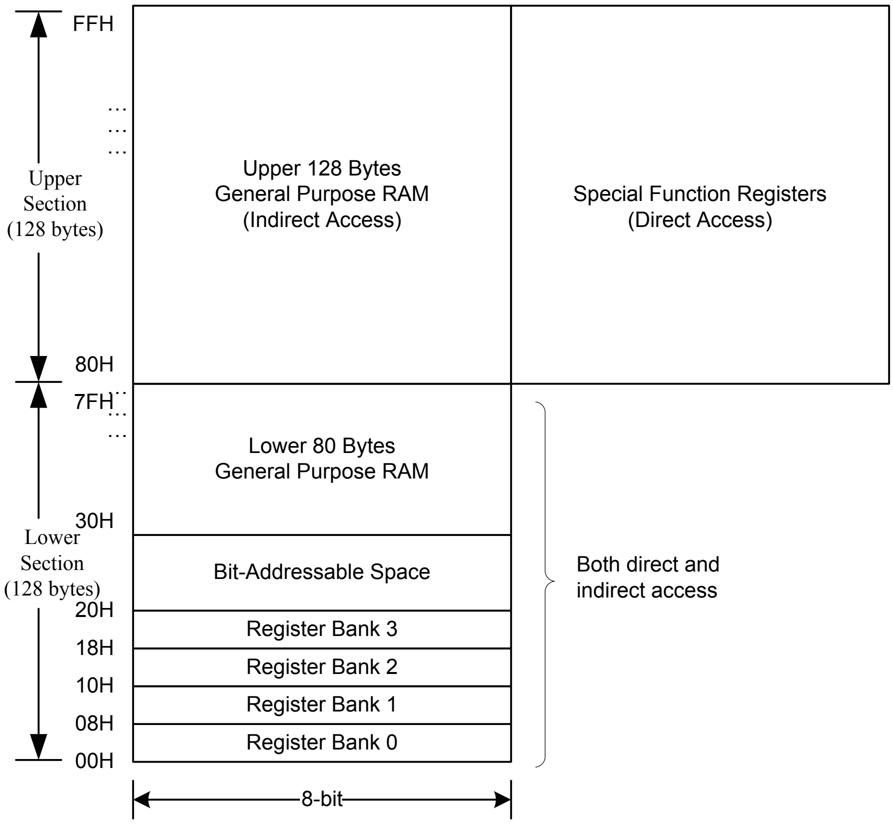
The upper section 128-byte RAM has an address range of 80H to FFH, and is assigned to both the General Purpose memory and the Special Function Registers. Although the address range is identical these two RAM sections are physically separate, they are distinguished by their different addressing methodology. Using direct addressing instructions will point to the SFR registers while indirect addressing instructions will point to the upper 128-byte General Purpose RAM.

The lower section 128-byte RAM is dedicated to the General Purpose RAM, and consists of an 80-byte General Purpose RAM section, four 8-byte register banks and 16-bytes of Bit-Addressable Space. The lower section can be accessed both by Indirect and Direct addressing methods. The 16-byte Bit-Addressable Space, which can be addressed by both byte format and 128 bit location format, is located from at the address range, 20H to 2FH. The four register banks, each of which contains eight bytes of general purpose registers, are located at the address range 00H to 1FH.

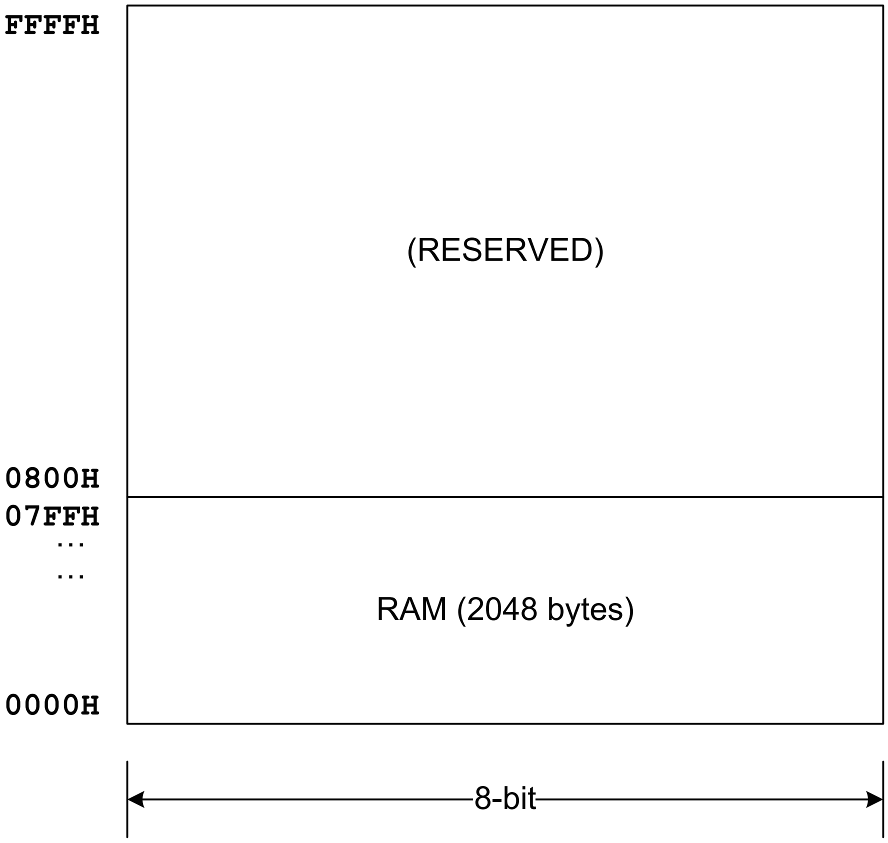
The XDATA is assigned as General Purpose Data RAM and can only be accessed using indirect addressing. The HT85F2270 and HT85F2280 have 2048-bytes of XDATA while the HT85F2260 has 1024-bytes of XDATA.

Note that the internal data memory is also used as a software stack. The designer must initiate the stack pointer register, namely SP, in the application program.

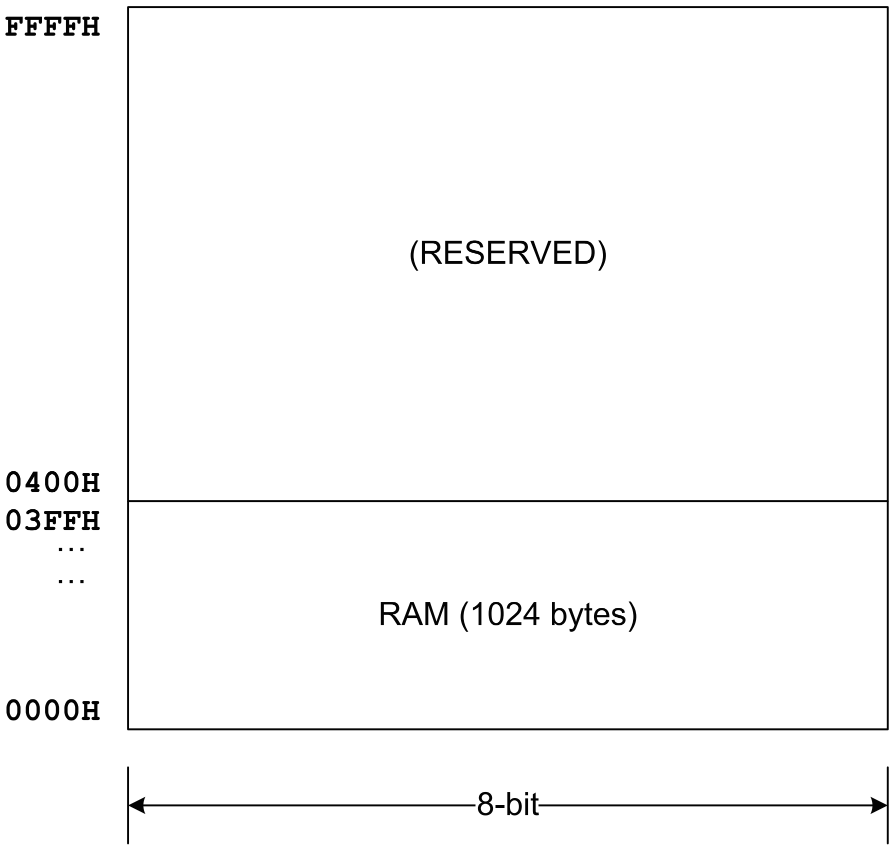
The following diagram illustrates the memory structure and their various access methods.



Internal Data Memory Structure



HT85F2270/HT85F2280 XDATA



HT85F2260 XDATA

Register Banks

There are four register banks, with addresses from 00H to 1FH, with each bank containing eight bytes. The active bank is selected by the control bits, RS1 and RS0, in the PSW register. It should be noted that only one bank can be enabled at any time. This total of 32 bytes are used as General Purpose data memory, which can be accessed by either direct or indirect instructions.

Bit Addressable Space

Some instructions in the 8051 language allow for single bit addressing. These single bit instructions can only be used in the bit addressable data memory area, located both in the General Purpose Data RAM and the Special Function Register area. Note that these bit addressable registers are both byte and bit addressable.

The 16 bytes bit addressable registers of the General Purpose Data RAM, located from 20H to 2FH, can address up to 128 individual bits. Each bit has its corresponding bit address from 00H to 7FH. For example, bit 0 of the 20H register is mapped to the bit address 00H, bit 7 of the 20H register is mapped to the bit address 07H and bit 7 of the 2FH register is mapped to the bit address 7FH. The accompanying table illustrates the Bit-Addressable register map description for General Purpose Data RAM, 20H~2FH. Using the bit operational instruction, such as SETB or CLR on the bit address can implement operations on the corresponding bit of the register. For example:

```
SETB 00H          ; Set the bit 0 of the register location 20H to "1"
SETB 07H          ; Set the bit 7 of the register location 20H to "1"
CLR 25H           ; Clear the bit 5 of the register location 24H to "0"
CLR 7FH           ; Clear the bit 7 of the register location 2FH to "0"
```

General Purpose Data RAM, 20H~2FH, Bit Address Map

High 5-bit Address	Low 3-bit Address							
	0H	1H	2H	3H	4H	5H	6H	7H
78H	0x2F.0	0x2F.1	0x2F.2	0x2F.3	0x2F.4	0x2F.5	0x2F.6	0x2F.7
70H	0x2E.0	0x2E.1	0x2E.2	0x2E.3	0x2E.4	0x2E.5	0x2E.6	0x2E.7
68H	0x2D.0	0x2D.1	0x2D.2	0x2D.3	0x2D.4	0x2D.5	0x2D.6	0x2D.7
60H	0x2C.0	0x2C.1	0x2C.2	0x2C.3	0x2C.4	0x2C.5	0x2C.6	0x2C.7
58H	0x2B.0	0x2B.1	0x2B.2	0x2B.3	0x2B.4	0x2B.5	0x2B.6	0x2B.7
50H	0x2A.0	0x2A.1	0x2A.2	0x2A.3	0x2A.4	0x2A.5	0x2A.6	0x2A.7
48H	0x29.0	0x29.1	0x29.2	0x29.3	0x29.4	0x29.5	0x29.6	0x29.7
40H	0x28.0	0x28.1	0x28.2	0x28.3	0x28.4	0x28.5	0x28.6	0x28.7
38H	0x27.0	0x27.1	0x27.2	0x27.3	0x27.4	0x27.5	0x27.6	0x27.7
30H	0x26.0	0x26.1	0x26.2	0x26.3	0x26.4	0x26.5	0x26.6	0x26.7
28H	0x25.0	0x25.1	0x25.2	0x25.3	0x25.4	0x25.5	0x25.6	0x25.7
20H	0x24.0	0x24.1	0x24.2	0x24.3	0x24.4	0x24.5	0x24.6	0x24.7
18H	0x23.0	0x23.1	0x23.2	0x23.3	0x23.4	0x23.5	0x23.6	0x23.7
10H	0x22.0	0x22.1	0x22.2	0x22.3	0x22.4	0x22.5	0x22.6	0x22.7
08H	0x21.0	0x21.1	0x21.2	0x21.3	0x21.4	0x21.5	0x21.6	0x21.7
00H	0x20.0	0x20.1	0x20.2	0x20.3	0x20.4	0x20.5	0x20.6	0x20.7

There are also 16 bytes of bit addressable registers located in the SFR which are both byte and bit addressable. These bit addressable registers in the SFR are registers whose addresses end with the low 3-bit address of “000b”, such as 80h, 88h, 90h…F8h, etc. The accompanying table illustrates the Bit-Addressable registers in the SFR. Using special instructions, such as SETB and CLR, can implement operations on the individual bit. For example:

```
SETB   ACC.3        ; Set the bit 3 of the ACC register to “1”
CLR    ACC.3        ; Clear the bit 3 of the ACC register to “0”
```

Special Function Register Bit Addresses Map

High 5-bit Address	Low 3-bit Address							
	0H	1H	2H	3H	4H	5H	6H	7H
F8h	FMCR.0	FMCR.1	FMCR.2	—	—	—	FMCR.6	FMCR.7
F0h	B.0	B.1	B.2	B.3	B.4	B.5	B.6	B.7
E8h	SPCON.0	SPCON.1	SPCON.2	SPCON.3	SPCON.4	SPCON.5	SPCON.6	SPCON.7
E0h	ACC.0	ACC.1	ACC.2	ACC.3	ACC.4	ACC.5	ACC.6	ACC.7
D8h	—	—	I2CCON.2	I2CCON.3	I2CCON.4	I2CCON.5	I2CCON.6	—
D0h	PSW.0	PSW.1	PSW.2	PSW.3	PSW.4	PSW.5	PSW.6	PSW.7
C8h	T2CON.0	T2CON.1	T2CON.2	T2CON.3	T2CON.4	T2CON.5	T2CON.6	—
C0h	—	IRCON.1	IRCON.2	IRCON.3	IRCON.4	IRCON.5	IRCON.6	IRCON.7
B8h	IP0.0	IP0.1	IP0.2	IP0.3	IP0.4	IP0.5	IP0.6	—
B0h	P3.0	P3.1	P3.2	P3.3	P3.4	P3.5	P3.6	P3.7
A8h	IEN0.0	IEN0.1	IEN0.2	IEN0.3	IEN0.4	IEN0.5	IEN0.6	IEN0.7
A0h	P2.0	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6	P2.7
98h	S0CON.0	S0CON.1	S0CON.2	S0CON.3	S0CON.4	S0CON.5	S0CON.6	S0CON.7
90h	P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7
88h	TCON.0	TCON.1	TCON.2	TCON.3	TCON.4	TCON.5	TCON.6	TCON.7
80h	P0.0	P0.1	P0.2	P0.3	P0.4	P0.5	P0.6	P0.7

Notes: 1. address in this table is “bit address”
2. “—” is stand for unimplemented

Special Function Registers

To ensure successful operation of the microcontroller, certain internal registers, known as Special Function Registers or SFRs for short, are implemented in the Data Memory area. These registers ensure correct operation of internal functions such as timers, interrupts, etc., as well as external functions such as I/O data control. The SFRs are located at the address range 80H to FFH in the upper section and are addressed directly. All can be addressed by byte but some are also bit-addressable. The following table shows the SFR register list. Note that some of the registers are defined by standard 8051 protocol while others are defined by Holtek.

Special Function Register Map

High 5-bit Address	Low 3-bit Address							
	0H	1H	2H	3H	4H	5H	6H	7H
F8h	FMCR	FMKEY	FMAR0	FMAR1	FMAR2	FMDR	T2CON1	RSTSRC
F0h	B	ADCR0	ADCR1	ADCR2	ADPGA	ADRL	ADRH	SRST
E8h	SPCON	I2CLK	LVRCR	LVDCR	SCCR	PLLCR	LSOCR	HSOCR
E0h	ACC	SPSTA	FMSR	SPDAT	IP1	IP1H	IP2	IP2H
D8h	I2CCON	P5	I2CDAT	I2CADR	SBRCON	I2CSTA	CP0CR	CP1CR
D0h	PSW	—	—	—	—	—	—	—
C8h	T2CON	IEN3	CRCL	CRCH	TL2	TH2	IP3	IP3H
C0h	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
B8h	IP0	IP0H	S0RELH	S1RELH	—	CPHCR	CPICR	IRCON2
B0h	P3	P4	TBCR	DACTRL	DAL	DAH	P3M0	P3M1
A8h	IEN0	IEN1	S0RELL	—	—	—	P2M0	P2M1
A0h	P2	T3CON	TL3	TH3	SRCR	SPPRE	P1M0	P1M1
98h	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL	P0M0	P0M1
90h	P1	P0WAKE	DPS	DPC	—	—	WDTCR	—
88h	TCON	TMOD	TL0	TL1	TH0	TH1	—	TMPRE
80h	P0	SP	DPL	DPH	DPL1	DPH1	WDTREL	PCON

Notes: "—": unimplemented

Most of the Special Function Registers will be described in detail under the function that they are related to. In this section a register description is provided for those registers which are not described elsewhere.

ACC Register – Accumulator

The Accumulator is central to the operation of any microcontroller and is closely related with operations carried out by the ALU. The Accumulator is the place where all intermediate results from the ALU are stored. Without the Accumulator it would be necessary to write the result of each calculation or logical operation such as addition, subtraction, shift, etc., to the Data Memory resulting in higher programming and timing overheads. Data transfer operations usually involve the temporary storage function of the Accumulator; for example, when transferring data between one user-defined register and another, it is necessary to do this by passing the data through the Accumulator as no direct transfer between two registers is permitted.

B Register

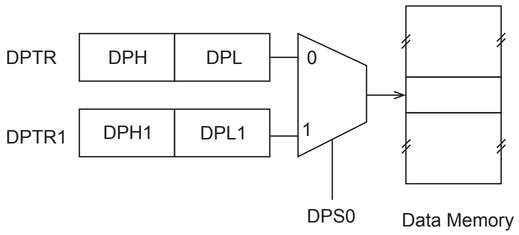
The B register is used as a general purpose register for these devices. It is used during multiplying and division instructions.

SP Register – Stack Pointer

The Stack Pointer register is 8 bits wide. It denotes the top of the Stack, which is the last used value. The user can place the Stack anywhere in the internal scratchpad Data Memory by setting the Stack Pointer to the desired location, although the lower bytes are normally used for working registers. After a reset, the Stack Pointer is initialised to 07H. This causes the stack to begin at location 08H. It is used to store the return address of the main program before executing interrupt routines or subprograms. The SP is incremented before executing a PUSH or CALL instruction and it is decremented after executing a POP, RET or RETI instruction.

DPL, DPH, DPL1, DPH1 Registers – Data Pointer Registers

The Data Pointer (DPTR) registers, DPL, DPH, DPL1 and DPH1, although having their locations in normal Data Memory register space, do not actually physically exist as normal registers. Indirect addressing instructions for Data Memory data manipulation use these Indirect Addressing Registers and Memory Pointers, in contrast to direct memory addressing, where the actual memory address is specified. Actions on the DPTR registers will result in no actual read or write operation to these registers but rather to the memory location specified by their corresponding Memory Pointer for the MOVX, MOVC or JMP instructions. The DPTR registers can be operated as two 16-bit registers or four individual 8-bit registers. There are two sets of 16-bit Data Pointer register: DPTR1 and DPTR. The DPTR register is composed of DPL and DPH, while the DPTR1 register is composed of DPL1 and DPH1. They are generally used to access external code or data space using instructions such as MOVC A,@A+DPTR or MOVX A,@DPTR respectively. The selection of DPTR or DPTR1 is controlled by the DPS0 bit. Setting the DPS0 bit high will select the DPTR1 register, otherwise the DPTR register is selected.



DPTRn Registers Control Block Diagram

Data Pointer Select Registers

The devices contain up to two data pointers, depending on configuration. Each of these registers can be used as 16-bits address source for indirect addressing. The DPS register serves to select the active data pointer register.

DPS Register – Data Pointer Select Register
SFR Address: 92h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	DPS0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 Unimplemented, read as "0"

Bit 0 **DPS0:** Data Pointer Register select
0: DPTR selected
1: DPTR1 selected
This bit is used to determine if the accessing addresses are sourced from either DPTR or DPTR1 when executing Read and Write instructions.

Data Pointer Control Register

This register is used to control whether the DPTR auto-increment/auto-decrement has a value of either 1 or 2, and auto-switching between active DPTRs functions. The auto-switching active DPTR function is controlled by the DPC3 bit in the DPC register. The content of this bit will be loaded to the DPS register after a MOVX @ DPTR instruction is executed. The auto-modification function is controlled by the DPC0 bit. When this bit is enabled, the current DPTR can be automatically increased or decreased by 1 or 2 positions selected by the DPC1 and DPC2 bits.

There are separate DPC register controls for each DPTR, to provide flexibility during data transfer operations. The actual DPC register is selected using the DPS register. If the DPS0 bit is set high, then DPTR1 is selected, and the DPC register is used as the DPTR1 control register. If the DPS0 bit is cleared to zero, the DPTR is selected, and the DPC register is used as the DPTR control register.

DPC Register – Data Pointer Control Register
SFR Address: 93h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	DPC3	DPC2	DPC1	DPC0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as “0”

Bit 3 **DPC3:** Next Data Pointer select

The content of this bit will be loaded to the “DPS” register after each MOVX @DPTR instruction is executed.

Note that this feature is always enabled, therefore for each of the “DPC” register this field has to contain a different value pointing to itself so that the auto-switching does not occur with default (reset) values.

Bit 2 **DPC2:** Auto-modification size
0: Modified size by 1
1: Modified size by 2

The current DPTR will be automatically modified by size, selected by the DPC2 bit, after each MOVX @DPTR instruction when DPC0=1.

Bit 1 **DPC1:** the current DPTR Auto-modification direction
0: Automatically incremented
1: Automatically decremented

The current DPTR will be automatically decremented or incremented, selected by the DPC1 bit, after each MOVX @DPTR instruction when DPC0=1.

Bit 0 **DPC0:** Auto-modification control bit
0: Disable
1: Enable

When this bit is set to high, enables auto-modification of the current DPTR after each MOVX @DPTR instruction.

Program Status Word

This register contains the Parity flag (P), General purpose flag 1 (F1), overflow flag (OV), Register bank select control bits (RS0, RS1), General purpose flag 0 (F0), Auxiliary Carry flag (AC) and Carry flag (CY). These arithmetic/logical operation and system management flags are used to record the status and operation of the microcontroller. Note that the Parity bit can only be modified by hardware depending upon the ACC state.

PSW Register – Program Status Word Register
SFR Address: D0h

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
POR	0	0	0	0	0	0	0	0

Bit 7 **CY:** Carry flag
0: No carry-out
1: An operation results in a carry during arithmetic operations and accumulator for Boolean operations.

Bit 6 **AC:** Auxiliary flag
0: No auxiliary carry
1: An operation results in a carry out of the low nibbles in addition, or no borrow from the high nibble into the low nibble on subtraction.

Bit 5 **F0:** General Purpose Flag 0
This bit is used as a general purpose flag by the application program.

Bit 4~3 **RS1~RS0:** Select Data Memory Banks
00: Bank 0
01: Bank 1
10: Bank 2
11: Bank 3

RS1	RS0	Selected Register Bank	Locations (within Internal Data Area)
0	0	Bank 0	00H – 07H
0	1	Bank 1	08H – 0FH
1	0	Bank 2	10H – 17H
1	1	Bank 3	18H – 1FH

Bit 2 **OV:** Overflow flag
0: No overflow
1: An operation results in a carry into the highest-order bit but not a carry out of the highest-order bit or vice versa.

Bit 1 **F1:** General Purpose Flag 1
This bit is used as a general purpose flag by the application program.

Bit 0 **P:** Parity flag
0: Accumulator contains an even number of ‘1’s
1: Accumulator contains an odd number of ‘1’s
This bit is used to indicate the number of ‘1’s in the Accumulator.

20 Oscillators

Various oscillator options offer the user a wide range of functions according to their various application requirements. The flexible features of the oscillator functions ensure that the best optimisation can be achieved in terms of speed and power saving. Oscillator selections and operation are selected using internal registers.

System Oscillator Overview

In addition to being the source of the main system clock the oscillators also provide clock sources for the Watchdog Timer and Time Base functions. External oscillators requiring some external components as well as two fully integrated internal oscillators, requiring no external components, are provided to form a wide range of both fast and slow system oscillators. After a reset occurs the HIRC oscillator is selected as the initial system clock but can be later switched by the application program using the clock control register.

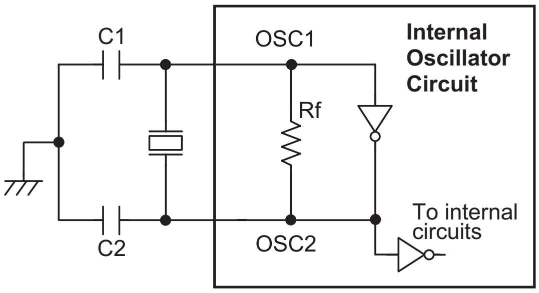
Type	Name	Function	Freq.	Pins
External High Speed Crystal	HXT	Precision High Speed System Clock	400kHz~24MHz	OSC1/OSC2
Internal High Speed RC	HIRC	High Speed System Clock	3.6864MHz	—
External Low Speed Crystal	LXT	Precision WDT and Time Base Clock	32768Hz	XT1/XT2
Internal Low Speed RC	LIRC	WDT and Time Base Clock	32kHz	—

System Clock Configuration

There are four oscillators, two high speed oscillators and two low speed oscillators. The high speed oscillators are the external crystal, HXT, and the internal RC oscillator, HIRC, which are used as the system oscillators. The two low speed oscillators are the external 32768Hz oscillator, LXT, and the internal 32kHz RC oscillator, LIRC, which are used as peripheral clocks for the Watchdog Timer and Time Base functions.

External High Speed Crystal Oscillator – HXT

The simple connection of a crystal across OSC1 and OSC2 will create the necessary phase shift and feedback for oscillation. However, for some crystals and most resonator types, to ensure oscillation and accurate frequency generation, it is necessary to add two small value external capacitors, C1 and C2. The exact values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer's specification. The external crystal frequency can be multiplied from 1 to 8 times using the internal PLL. For example, if a 4MHz crystal is used for oscillator and if the PLL is selected as 8 times, the system clock can be increased to 32MHz. Note that if the internal PLL is enabled, the external crystal frequency should be fixed at 4MHz; otherwise, an unexpected frequency will be generated. When the internal PLL function is not to be used, the external crystal frequency can be within the range, from 400kHz to 24MHz.



Crystal/Resonator Oscillator – HXT

Crystal Recommended Capacitor Values

Crystal Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
24MHz	10pF	10pF
12MHz	10pF	10pF
8 MHz	10pF	10pF
4 MHz	20pF	20pF
400kHz	300pF	300pF

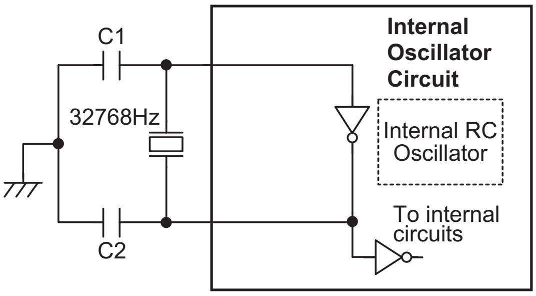
Note: C1 and C2 values are for guidance only.

Internal High Speed RC Oscillator – HIRC

The internal RC oscillator is a fully integrated system oscillator requiring no external components. The internal RC oscillator has a single frequency of 3.6864MHz. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. The internal RC oscillator frequency can be multiplied from 1 to 8 times using the internal PLL. If the HIRC oscillator is used as the system oscillator, then the OSC1 and OSC2 pins should be left unconnected.

External Low Speed Crystal Oscillator – LXT

The external low speed crystal oscillator, LXT, is used as the clock source for the Watchdog Timer and the Time Base functions. When the microcontroller enters the IDLE Mode, the CPU clock is switched off to stop microcontroller activity and to conserve power, however the LXT oscillator will continue to run and can maintain WDT and Time Base operation if it is selected as their clock source. The LXT oscillator is implemented using a 32768Hz crystal connected to pins XT1/XT2. However, for some crystals, to ensure oscillation and accurate frequency generation, it is necessary to add two small value external capacitors, C1 and C2. The exact values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer's specification.



External LXT Oscillator – LXT

32768Hz Crystal Recommended Capacitor Values

LXT Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
32768Hz	10pF	10pF

Note: C1 and C2 values are for guidance only.

Internal Low Speed RC Oscillator – LIRC

The internal low speed oscillator, LIRC, is a fully self-contained free running on-chip RC oscillator, used as a clock source for the Watchdog Timer and the Time Base functions. When the microcontroller enters the IDLE Mode, the CPU clock is switched off to stop microcontroller activity and to conserve power, however the LIRC oscillator will continue to run and can maintain WDT and Time Base operation if it is selected as their clock source. The LIRC oscillator has a typical frequency of 32kHz at 5V and requires no external components, however its actual frequency may vary with temperature and supply voltage. For precise low speed oscillator functions the LXT oscillator should be used.

21 Operating Modes and System Clocks

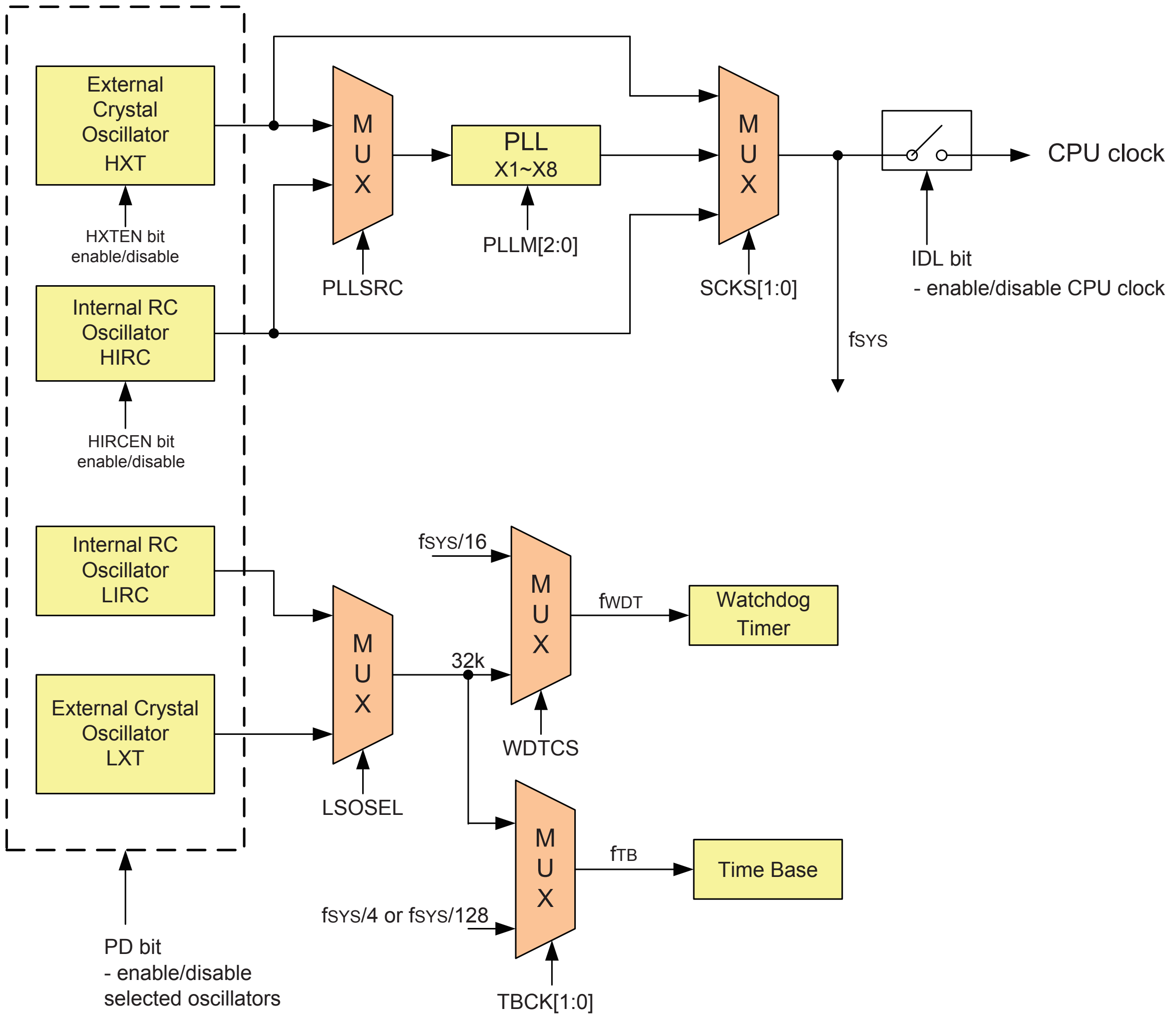
Present day applications require that their microcontrollers have high performance but often still demand that they consume as little power as possible, conflicting requirements that are especially important in battery powered portable applications. This usually requires the microcontroller can provide a range of clock sources which can be dynamically selected.

System Clocks Description

The fast clocks required for high performance will inherently have a higher power consumption and of course vice-versa, lower speed clocks will have a lower power consumption. As Holtek has provided these devices with a range of oscillators and a PLL function the user can optimise the system clock frequency to achieve the best performance/power ratio. In addition to the two high frequency system oscillators, two low frequency 32kHz oscillators are also provided as clock sources for the WDT and Time Base.

The MCU system clock is sourced from the high speed external crystal, HXT oscillator, or internal, HIRC oscillator. These oscillators can be used directly as the system clock and can be routed via an internal PLL to give a wide range of operating frequencies. The PLL frequency can be dynamically changed to suit varying operating conditions and to achieve maximum performance.

The system clock, namely f_{SYS}, can also be used as a clock source for the peripheral functions, such as WDT, Time Base, Timers, UART, I^2C, SPI, ADC and DAC. Refer to the related sections for the clock source selections.



System Clock Configurations

The main system clock source, known as f_{SYS}, and which is used by the CPU and the peripheral functions, can come from one of three sources. These are the internal HIRC oscillator, the external crystal HXT oscillator or a frequency multiplied version of these oscillators using the internal PLL. The selection is implemented using the SCKS0 and SCKS1 bits in the SCCR register. The HXT and HIRC oscillators also have independent enable control bits, which are the HXTEN and HIRCEN bits in the HSOCR register. There are also two oscillator status bits, HIRCRDY and HXTRDY, in the HSOCR register to indicate whether the oscillators are ready for operation. After power on, these bits should be monitored by the program to indicate the "ready or not" status of the respective oscillator, before they are used with instruction execution. After power on, the device will automatically select the HIRC oscillator as its default system clock, which can be changed later by the application program.

There are two additional internal 32kHz low frequency clocks for the peripheral circuits. These are the external crystal LXT oscillator and the internal LIRC oscillators. The selection is implemented using the LSOSEL bit in the LSOCR register. There is a low frequency oscillator status bit, LSORDY, to indicate the "ready or not" status of the low frequency oscillator. This bit is common to both low frequency oscillators, and should be monitored by the program to indicate the "ready or not" status of the oscillator before it is used for instruction execution. This bit will be automatically cleared to zero during low speed oscillator switching and set high once the chosen oscillator is stable.

System Clock Control Register – SCCR
SFR Address: ECh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	SCKS1	SCKS0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **SCKS1, SCKS0:** High Frequency System clock select
00: HIRC oscillator clock source
01: HIRC oscillator clock source
10: HXT oscillator clock source
11: PLL clock source
The HIRC will be the default system clock source after a power on reset.
When switching between different clock sources an oscillator stabilisation time delay must be provided before continuing with program execution.

High Speed Oscillator Control Register – HSOCR
SFR Address: EFh

Bit	7	6	5	4	3	2	1	0
Name	—	—	HXTRDY	HIRCRDY	—	—	HXTEN	HIRCEN
R/W	—	—	R	R	—	—	R/W	R/W
POR	—	—	0	1	—	—	0	1

Bit 7~6 Unimplemented, read as "0"

Bit 5 **HXTRDY:** HXT oscillator ready indication bit
0: Not ready
1: Ready
This is the external high frequency oscillator, HXT, ready indication bit which indicates if the HXT oscillator is stable or not. This bit will be cleared to zero by hardware when the device is powered on. After power on, if the HXT oscillator is selected, the bit will change to a high level when the external high frequency oscillator is stable.

Bit 4 **HIRCRDY:** HIRC oscillator ready indication bit
0: Not ready
1: Ready
This is the internal high frequency oscillator, HIRC, ready indication bit which indicates if the HIRC oscillator is stable or not. This bit will be cleared to zero by hardware when the HIRC function is disabled. After power on, if the HIRC oscillator is enabled, the bit will change to a high level when the internal high frequency oscillator is stable.

Bit 3~2 Unimplemented, read as "0"

Bit 1 **HXTEN:** HXT control bit
0: Disable
1: Enable

Bit 0 **HIRCEN:** HIRC control bit
0: Disable
1: Enable
After power on, this bit will be set high thus selecting the HIRC as the initial system oscillator.

Low Speed Oscillator Control Register – LSOCR
SFR Address: EEh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	LSORDY	—	—	LSOSEL	—
R/W	—	—	—	R	—	—	R/W	—
POR	—	—	—	1	—	—	0	—

Bit 7~5 Unimplemented, read as "0"

Bit 4 **LSORDY:** Low speed oscillator ready indication flag
0: Not ready
1: Ready
This is the common ready flag for the two low speed oscillators, LIRC and LXT, which indicates if the low speed oscillator is stable or not. During low speed oscillator switching this bit will be automatically cleared to zero by the hardware.

Bit 3~2 Unimplemented, read as "0"

Bit 1 **LSOSEL:** Low frequency oscillator select bit
0: LIRC
1: LXT

Bit 0 Unimplemented, read as "0"

Phase Locked Loop – PLL

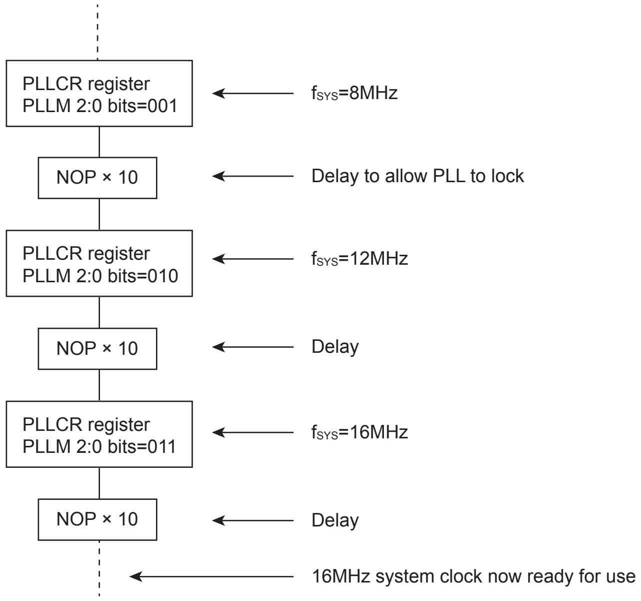
All devices contain a fully internal PLL function which is used to multiply the frequency of the selected high speed oscillator, either HIRC or HXT. As all PLL functions are internal, no external components, including those for the loop filter, are required.

The PLL is enabled by the PLLEN bit in the PLLCR register. After being enabled the PLL must be given a certain amount of time to lock and stabilise. After the PLL is enabled the PLLRDY bit should be monitored to indicate when the PLL has locked and is ready for use. If the PLL function is disabled, then the high frequency oscillators can be used directly as the system clock. The PLL input clock source, from either the HIRC or HXT oscillators, is determined by the PLLSRC bit in the PLLCR register. The frequency multiplier range has a range of one to eight times, selected by the PLLM0~PLLM2 bits in the PLLCR register.

Changing the PLL Frequency

After the PLL is enabled and is being used as the system clock, its frequency can be changed dynamically by the application program, by programming the PLLM0~PLLM2 bits in the PLLCR register. However the program must execute this operation in a specific way to ensure stable frequency switching. There are a total of eight different PLL frequency multiplier selections, however during dynamic PLL frequency changing, the multiplier value should only be changed one stage at a time. In addition a recommended delay of at least 10 instruction cycles, which can be implemented by 10 NOP instructions, should be inserted after each frequency multiplier stage change to allow the PLL to re-lock and stabilise. Note that the PLLRDY bit will remain at a high level during any dynamic PLL frequency change and cannot be used to indicate PLL stability after the PLL changes frequency. The accompanying flowchart illustrates this point.

Example: Change the system clock from 8 MHz to 16 MHz



Note: 4MHz HXT external crystal oscillator

PLL Frequency Changing

PLL Control Register – PLLCR
SFR Address: EDh

Bit	7	6	5	4	3	2	1	0
Name	PLLEN	PLLRDY	—	PLLSRC	—	PLLM2	PLLM1	PLLM0
R/W	R/W	R	—	R/W	—	R/W	R/W	R/W
POR	0	0	—	0	—	0	0	0

Bit 7 **PLLEN:** PLL enable/disable control
0: PLL disable
1: PLL enable

Bit 6 **PLLRDY:** PLL output ready indication flag
0: Not ready
1: Ready

After the PLL is enabled this bit is used to indicate when the PLL is locked and ready for use. This bit will be initially cleared to zero by hardware when the device is powered on. The bit will be cleared to zero if the PLL is in use and is then disabled but will not be cleared if the PLL changes frequency.

Bit 5 Unimplemented, read as “0”

Bit 4 **PLLSRC:** PLL Clock Source Select
0: HIRC clock source
1: HXT clock source

Note that if the PLL clock source is selected to be the external oscillator, HXT, the crystal frequency should be 4MHz.

Bit 3 Unimplemented, read as “0”

Bit 2~0 **PLLM2, PLLM1, PLLM0:** PLL Frequency Multiplier select
000: ×1
001: ×2
010: ×3
011: ×4
100: ×5
101: ×6
110: ×7
111: ×8

Operation Modes

There are three different modes of operation for the microcontroller, each one with its own special characteristics and which can be chosen according to the specific performance and power requirements of the application. There is one mode allowing normal operation of the microcontroller, the NORMAL Mode, in which all oscillators and function remain active. There are also two low power modes, the IDLE mode and the Power-Down Mode. In the IDLE mode, the microcontroller CPU will stop and instruction execution will cease, however, the high speed oscillators will continue to run and can continue to provide a clock source for the peripheral functions such as WDT, Time Base, Timers, UARTs, I^{2}C, SPI, ADC and DAC. The slow speed oscillators will also continue to run and keep the WDT and Time Base functions active, if their clock sources are not the system clock. In the Power-Down mode all oscillators are stopped and therefore all functions cease operation.

Operating Mode	NORMAL Mode	IDLE Mode	Power-Down Mode
CPU Clock	On	Off	Off
Peripheral Clock(Note)	On	On	Off
Low Frequency XTAL Oscillator (LXT)	On (LSOSEL=1)/Off	On (LSOSEL=1)/Off	Off
Low Frequency Internal RC Oscillator (LIRC)	On (LSOSEL=0)/Off	On (LSOSEL=0)/Off	Off
High Frequency XTAL Oscillator (HXT)	On (HXTEN=1)/Off	On (HXTEN=1)/Off	Off
High Frequency Internal RC Oscillator (HIRC)	On (HIRCEN=1)/Off	On (HIRCEN=1)/Off	Off

Note: Peripheral Clock is the clock for Timer 0, Timer 1, Timer 2, Timer 3, PCA, UART0, UART1, I^{2}C, SPI, ADC, and DAC.

NORMAL Mode

As the name suggests this is the main operating mode where all of the selected oscillators and clocks are active and the microcontroller has all of its functions operational and where the system clock is provided directly by one of the high speed oscillators, HXT, HIRC or the PLL.

IDLE Mode

The IDLE Mode is entered when the IDL bit in the PCON register is set high. When the instruction that sets the IDL bit high is executed the CPU operation will be inhibited, however, the high frequency clock source will continue to run and can continue to provide a clock source for the peripheral functions if selected. The low frequency clock sources will also remain operational and can also provide a clock source for the WDT and Time Base functions, if they are enabled and if their clock source is not selected to come from the system clock.

Power-Down Mode

The Power-Down Mode is entered when the PD bit in the PCON register is set high. When the instruction that sets the PD bit high is executed the all oscillators will stop thus inhibiting both CPU and peripheral functions such as the WDT and Time Base if they are enabled.

Power Control Register

Two bits, PD and IDL, in the PCON register control overall mode selection.

PCON Register – Power Control Register
SFR Address: 87h

Bit	7	6	5	4	3	2	1	0
Name	SMOD	—	—	—	—	GF0	PD	IDL
R/W	R/W	—	—	—	R	R/W	R/W	R/W
POR	0	—	—	—	1	0	0	0

Bit 7 **SMOD:** Serial Port 0 double baud rate select
Described elsewhere

Bit 6~3 Unimplemented

Bit 2 **GF0:** General Purpose bit

Bit 1 **PD:** Power-Down Mode control bit
0: No Power-Down – selected oscillators running
1: Power-Down – all oscillators stopped
Setting the PD bit to high will enable the Power-Down mode function. This bit will be cleared by hardware before entering the Power-Down mode and always read as “0”.

Bit 0 **IDL:** IDLE Mode control bit
0: No Idle Mode – CPU clock running
1: Idle Mode – CPU clock stopped
Setting the IDL bit to high will enable IDLE mode function. This bit will be cleared by hardware before entering the IDLE mode and always read as “0”. Note that if the PD bit is set high, to enable the Power-Down Mode, then the condition of the IDL bit will be overridden.

Standby Current Considerations

As the main reason to stop the oscillators is to keep the current consumption of the MCU to as low a value as possible, perhaps only in the order of several micro-amps, there are other considerations which must also be taken into account by the circuit designer if the power consumption is to be minimised.

Special attention must be made to the I/O pins on the device. All high-impedance input pins must be connected to either a fixed high or low level as any floating input pins could create internal oscillations and result in increased current consumption. Care must also be taken with the loads, which are connected to I/O pins, which are setup as outputs. These should be placed in a condition in which minimum current is drawn or connected only to external circuits that do not draw current, such as other CMOS inputs. And for power saving purpose, all the analog modules have to be disabled using the application program before MCU enter the IDLE or Power-Down mode.

The high speed and low speed oscillators will continue to run when in the IDLE Mode and will thus consume some power.

Wake-up

After the system enters the IDLE or Power-Down Mode, it can be woken up from one of various sources listed as follows:

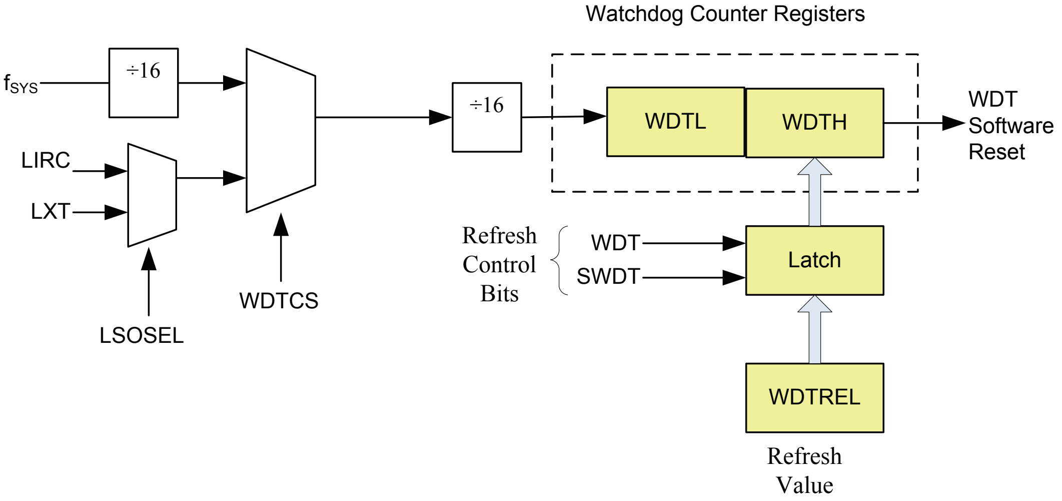
- An external reset
- An external low level on any P0 I/O pin
- A system interrupt
- A WDT overflow

If the system is woken up by an external reset, the device will experience a full system reset, however, if the device is woken up by a WDT overflow, a Watchdog Timer reset will be initiated. Pins P0 [0:7] can be setup via the P0WAKE register to permit a low level on the pin to wake-up the system. When an I/O pin wake-up occurs, the program will resume execution at the instruction following the point where the PD or IDL control bits were set high.

If the system is woken up by an interrupt, then two possible situations may occur. The first is where the related interrupt is disabled or the interrupt is enabled but the stack is full, in which case the program will resume execution at the instruction following the control bits settings. In this situation, the interrupt which woke-up the device will not be immediately serviced, but will rather be serviced later when the related interrupt is finally enabled or when a stack level becomes free. The other situation is where the related interrupt is enabled and the stack is not full, in which case the regular interrupt response takes place. If an interrupt request flag is set to 1 before entering the IDLE or Power-Down modes, then any interrupt requests will not generate a wake-up function and the related interrupt will be ignored. No matter what the source of the wake-up event is, once a wake-up event occurs, the program can check if the system clock is stable or not by examining the oscillator status bits. It is recommended that these bits are examined before proceeding with instruction execution after a wake up.

22 Watchdog Timer

The Watchdog Timer, also known as the WDT, is provided to inhibit program malfunctions caused by the program jumping to unknown locations or entering endless program loops, due to certain uncontrollable external events such as electrical noise. Its basic structure is a 16-bit timer which when it overflows will execute an MCU reset operation. The accompanying diagram illustrates the basic operational block diagram.



Watchdog Timer

Watchdog Registers

There are several registers for overall watchdog timer operation. The WDTREL register is used to setup the reload value of the Watchdog Timer. The remaining four registers are control registers which setup the operating and control function of the WDT function. The WDTCR register controls the WDT enable/disable operation, software reset and clock source select functions. The WDT and SWDT bits, located in the IEN0 and IEN1 registers respectively, are used to refresh the WDT counter to prevent the WDT overflow and reset the device. The WDTS bit in the IP0 register is used to indicate that a WDT software reset has been generated. For details regarding the WDT software reset function, refer to the datasheet Reset section for details.

WDT Register Contents

Name	Bit							
	7	6	5	4	3	2	1	0
IEN0	(EAL)	WDT	(ET2)	(ES0)	(ET1)	(EX1)	(ET0)	(EX0)
IEN1	(EXEN2)	SWDT	(ET3)	(ECMP)	(EX6)	(EX5)	(EX4)	(EX3)
WDTREL	D7	D6	D5	D4	D3	D2	D1	D0
WDTCR	WE4	WE3	WE2	WE1	WE0	—	—	WDTCS
IP0	—	WDTS	(PT2)	(PS0)	(PT1)	(PX1)	(PT0)	(PX0)

Note: The bit and flag names in brackets are used to manage other functions and not related to the WDT control.

IEN0 Register
SFR Address: A8h

Bit	7	6	5	4	3	2	1	0
Name	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **EAL:** Master interrupt global enable
Described elsewhere

Bit 6 **WDT:** Watchdog timer refresh flag
Setting this bit to “1” is the first step in initiating a Watchdog Timer refresh action. This WDT bit must be set immediately before setting the SWDT bit in the IEN1 register. The two instructions should be executed consecutively and not have any other instruction in between to prevent an unintentional watchdog timer refresh. This bit will be cleared by hardware automatically. This bit is always read as 0.

Bit 5 **ET2:** Timer2 interrupt enable
Described elsewhere

Bit 4 **ES0:** Serial Port 0 interrupt enable
Described elsewhere

Bit 3 **ET1:** Timer1 overflow interrupt enable
Described elsewhere

Bit 2 **EX1:** External interrupt 1 enable
Described elsewhere

Bit 1 **ET0:** Timer0 overflow interrupt enable
Described elsewhere

Bit 0 **EX0:** External interrupt 0 enable
Described elsewhere

IEN1 Register
SFR Address: A9h

Bit	7	6	5	4	3	2	1	0
Name	EXEN2	SWDT	ET3	ECMP	EX6	EX5	EX4	EX3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **EXEN2:** Timer2 external reload interrupt enable
Described elsewhere

Bit 6 **SWDT:** Watchdog timer start/refresh flag
This bit is used to activate and refresh the watchdog timer.
When this bit is set to "1" directly after the WDT bit is set, a watchdog timer refresh will be enabled. This bit will be cleared by hardware automatically. This bit is always read as 0.

Bit 5 **ET3:** Timer 3 overflow interrupt enable
Described elsewhere

Bit 4 **ECMP:** Comparator overall interrupt enable
Described elsewhere

Bit 3 **EX6:** External interrupt 6 enable
Described elsewhere

Bit 2 **EX5:** External interrupt 5 enable
Described elsewhere

Bit 1 **EX4:** External interrupt 4 enable
Described elsewhere

Bit 0 **EX3:** External interrupt 3 enable
Described elsewhere

WDTREL Register
SFR Address: 86h

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 Watchdog reload value
Reload value for the highest 8 bits of the watchdog timer.
This value is loaded to the Watchdog Timer when a refresh is triggered by the consecutive setting of bits, WDT and SWDT.

WDTCR Register
SFR Address: 96h

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	—	—	WDTCS
R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W
POR	0	1	0	1	0	—	—	0

Bit 7~3 **WE4~WE0:** WDT function software control
10101: Disable
01010: Enable - default
Other values: Reset MCU

Bit 2~1 Unimplemented, read as “0”

Bit 0 **WDTCS:** Watchdog clock (f_{WDT}) select
0: LIRC or LXT
1: $f_{SYS}/16$

Note that the WDTCR value will default to 01010000B after any reset resource which means that the WDT will be enabled after any reset takes place. For more details regarding the reset operation, refer to the Reset section.

IP0 Register
SFR Address: B8h

Bit	7	6	5	4	3	2	1	0
Name	—	WDTS	PT2	PS0	PT1	PX1	PT0	PX0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 Unimplemented, read as “0”

Bit 6 **WDTS:** Watchdog timer reset indication flag
0: No Watchdog timer reset
1: Watchdog timer reset

Bit 5 **PT2:** Timer 2 Interrupt priority low
Described elsewhere

Bit 4 **PS0:** UART 0 Interrupt priority low
Described elsewhere

Bit 3 **PT1:** Timer 1 Interrupt priority low
Described elsewhere

Bit 2 **PX1:** External interrupt 1 priority low
Described elsewhere

Bit 1 **PT0:** Timer 0 Interrupt priority low
Described elsewhere

Bit 0 **PX0:** External interrupt 0 priority low
Described elsewhere

Watchdog Timer Clock Source

The Watchdog Timer clock source is provided by an internal clock which is in turn supplied by one of three sources selected by the WDTCS bit in the WDTCR register: a 32kHz clock or f_{SYS}/16. The 32kHz clock can be sourced from either the LXT or LIRC oscillators, selected by the LSOSEL bit in the LSOCR register. The Watchdog Timer source clock is then subdivided by a ratio of 16 to give a longer timeout. The LIRC internal oscillator has an approximate period of 32kHz at a supply voltage of 5V. However, it should be noted that this specified internal clock period can vary with V_{DD}, temperature and process variations. The LXT oscillator is supplied by an external 32.768kHz crystal. The other Watchdog Timer clock source option is the f_{SYS}/16 clock.

Watchdog Timer Operation

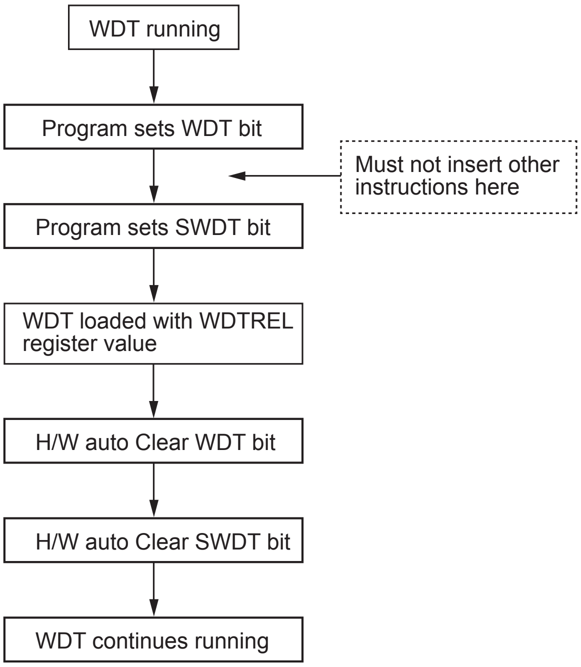
The Watchdog Timer operates by providing a device reset when its 16-bit timer overflows. The WDT is formed of two 8-bit registers, WDTL and WDTH, both of which are inaccessible to the application program. The WDTH register of the Watchdog Timer is reloaded with the contents of the WDTREL register. In the application program and during normal operation the user has to strategically clear the Watchdog Timer before it overflows to prevent the Watchdog Timer from executing a reset. This is done by setting the WDT and SWDT bits. If the program malfunctions for whatever reason, jumps to an unknown location, or enters an endless loop, these clear-bit instructions will not be executed in the correct manner as setup up by the user, in which case the Watchdog Timer will overflow and reset the device. There are five bits, WE4~WE0, in the WDTCR register to enable/disable the Watchdog Timer. The WE4~WE0 bits must be set to a specific value of "10101" to disable the WDT. A value of "01010" will enable the WDT while any other value will execute an MCU reset. Using this methodology, enhanced device protection is provided. After power on, these bits will have a value of "01010" which is the WDT enable setup value, and the WDT function will be enabled and began counting. The application program can disable the WDT at the beginning of the program if it is not required.

Watchdog Timer Enable/Disable Control

WE4~WE0 Bits	WDT Function
01010B	Enable
10101B	Disable
Other values	Reset MCU

The watchdog timer must be refreshed regularly to prevent the reset request signal, WDTS, from becoming active. This requirement imposes an obligation on the programmer to issue two consecutive instructions. The first instruction is to set the WDT bit of the IEN0 register and the second one is to set the SWDT bit in the IEN1 register. The maximum allowed delay time between setting the WDT and SWDT bits is one instruction cycle, which means the instructions which set the both bits should not be separated by any other instruction. If these instructions are not executed consecutively then the WDT refresh procedure is incomplete and an unexpected WDT reset will take place.

After the application program has set both the WDT and SWDT bits and the WDT refreshed, the WDT bit as well the SWDT bit will be automatically cleared by hardware. The 8 high-order bits of the Watchdog Timer are re-loaded with the contents of the WDTREL register. The larger the WDTREL value, the shorter the WDT time out will be. For the maximum WDT time out value, the WDTREL register should be cleared to zero.



Watchdog Timer Refresh Operation

23 Low Voltage Detector – LVD

Each device has a Low Voltage Detector function, also known as LVD. This enables the device to monitor the power supply voltage, V_{DD}, and provide an interrupt should it fall below a certain level. This function may be especially useful in battery applications where the supply voltage will gradually reduce as the battery ages, as it allows a battery low early warning signal to be generated. The LVD function can also generate an interrupt signal if required.

LVD Register

The Low Voltage Detector function is controlled using a single register with the name LVDCR. Three bits in this register, LVDS2~LVDS0, are used to select one of eight fixed voltages below which a low voltage condition will be determined. The LVDEN bit is used to control the overall on/off function of the low voltage detector. Setting the bit high will enable the low voltage detector. Clearing the bit to zero will switch off the internal low voltage detector circuits. As the low voltage detector will consume a certain amount of power, it may be desirable to switch off the circuit when not in use, an important consideration in power sensitive battery powered applications.

LVDCR Register
SFR Address: EBh

Bit	7	6	5	4	3	2	1	0
Name	LVDEN	—	—	—	—	LVDS2	LVDS1	LVDS0
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 **LVDEN:** LVD Function Control
0: Disable
1: Enable

Bit 6~3 Unimplemented, read as "0"

Bit 2~0 **LVDS2~LVDS0:** Select LVD Voltage
000: 2.0V
001: 2.2V
010: 2.4V
011: 2.7V
100: 3.0V
101: 3.3V
110: 3.6V
111: 4.2V

LVD Operation

The Low Voltage Detector function operates by comparing the power supply voltage, V_{DD}, with a pre-specified voltage level stored in the LVDCR register. This has a range of between 2.0V and 4.2V. When the power supply voltage, V_{DD}, falls below this pre-determined value and if the LVD interrupt function is enabled, the LVD interrupt will take place and the interrupt request flag, LVDF, in the IRCON2 register, will be set high. The LVDF bit will be cleared to low by hardware automatically. The LVD interrupt can cause the device to wake-up from the IDLE Mode. If the Low Voltage Detector wake up function is not required then the LVDF flag should be first set high and disable the LVD interrupt function before the device enters the IDLE Mode. When the device is powered down the low voltage detector will be disabled to reduce the power consumption.

24 Reset and Initialisation

A reset function is a fundamental part of any microcontroller ensuring that the device can be set to some predetermined condition irrespective of outside parameters. A hardware reset will of course be automatically implemented after the device is powered-on, however there are a number of other hardware and software reset sources that can be implemented dynamically when the device is running.

Reset Overview

The most important reset condition is after power is first applied to the microcontroller. In this case, internal circuitry will ensure that the microcontroller, after a short delay, will be in a well defined state and ready to execute the first program instruction. After this power-on reset, certain important internal registers will be set to defined states before the program instructions commence execution. One of these registers is the Program Counter, which will be reset to zero forcing the microcontroller to begin program execution from the lowest Program Memory address.

The devices provide several reset sources to generate the internal reset signal, providing extended MCU protection. The different types of resets are listed in the accompanying table.

Reset Source Summary

No.	Reset Name	Abbreviation	Indication Bit	Register	Notes
1	Power-On Reset	POR	PORF	RSTSRC	Auto generated at power on
2	Reset Pin	$\overline{\text{RESET}}$	XRSTF	RSTSRC	Hardware Reset
3	Low-Voltage Reset	LVR	LVRF	RSTSRC	Low V_{DD} voltage
4	LVRCR Register Setting Software Reset	—	LRF	RSTSRC	Write to LVRCR register
5	Watchdog Reset	WDT	WDTS	IP0	Watchdog overflow
6	WDTCR Register Setting Software Reset	—	WRF	RSTSRC	Write to WDTCR register
7	Comparator 0 Output Reset	—	CMP0F	RSTSRC	To enable – set CP0RST bit in CP0CR register
8	SRST Register Setting Software Reset	—	SRSTREQ	SRST	Write to SRST register
9	ROM Code Check Reset	—	—	—	—

Reset Operations

After the initial power on reset, there are many ways in which a microcontroller reset can occur, through events occurring both internally and externally.

Reset Source Register – RSTSRC

After a reset occurs the device will be reset to some initial condition. Several registers are used to indicate which actual reset type caused the device to reset. Seven of the possible reset sources will be indicated by the reset source register, RSTSRC. The additional reset sources are indicated by the SRSTREQ bit in the SRST register for the Software Reset and the WDTS bit in the IP0 register for the Watchdog reset. And the MCU reset can also caused by ROM Code Check.

All of the bits in the RSTSRC register are read only and can therefore not be cleared by the application program after one of the relevant reset occurs. After one of these reset occurs and the relevant bit is high to indicate its occurrence, the bit can only be cleared by hardware when another different reset type occurs.

RSTSRC Register
SFR Address: FFh

Bit	7	6	5	4	3	2	1	0
Label	—	LRF	WRF	—	CMP0F	LVRF	XRSTF	PORF
R/W	—	R	R	—	R	R	R	R
POR	—	0	0	—	0	x	0	1

Bit 7 Unimplemented, read as “0”

Bit 6 **LRF:** LVRCR Register Setting Software Reset Indication Flag
0: No LVRCR Setting Software Reset
1: LVRCR Software Reset

Bit 5 **WRF:** WDTCR Register Setting Software Reset Indication Flag
0: No WDTCR Setting Software Reset
1: WDTCR Setting Software Reset

Bit 4 Unimplemented, read as “0”

Bit 3 **CMP0F:** Comparator 0 Reset Indication Flag
0: No Comparator 0 Reset
1: Comparator 0 Reset

Bit 2 **LVRF:** Low-Voltage Reset Indication Flag
0: No Low-Voltage Reset
1: Low-Voltage Reset

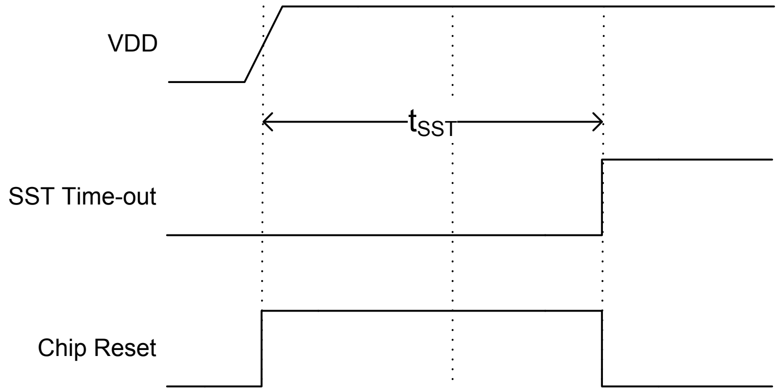
Bit 1 **XRSTF:** External Pin Reset Indication Flag
0: No External Reset
1: External Reset

Bit 0 **PORF:** Power-on Reset Indication Flag
0: No Power-on Reset
1: Power-on Reset

Power-on Reset

The most fundamental and unavoidable reset is the one that occurs after power is first applied to the microcontroller. As well as ensuring that the Program Memory begins execution from the first memory address, a power-on reset also ensures that certain other registers are preset to known conditions. The entire I/O data and port mode registers will power up to ensure that all pins will be first set to the quasi-bidirection structure.

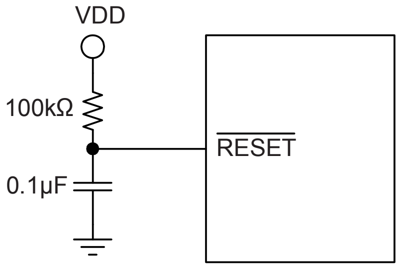
Although the microcontroller has an internal RC reset function, if the VDD power supply rise time is not fast enough or does not stabilise quickly at power-on, the internal reset function may be incapable of providing proper reset operation. For this reason it is recommended that an external RC network is connected to the $\overline{\text{RESET}}$ pin, whose additional time delay will ensure that the $\overline{\text{RESET}}$ pin remains low for an extended period to allow the power supply to stabilise. During this time delay, normal operation of the microcontroller will be inhibited. After the $\overline{\text{RESET}}$ line reaches a certain voltage value, the reset delay time of t_{SST}, which is equal to 1024 system clock pulses, is invoked to provide an extra delay time after which the microcontroller will begin normal operation. The abbreviation SST in the figures stands for System Start-up Timer. When the Power-on reset takes place, the PORF bit in the RSTSRC register will be set high to indicate this reset.



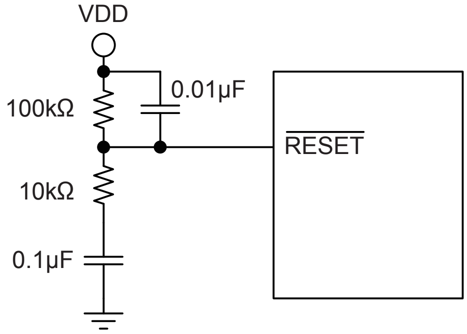
Power-On Reset Timing

$\overline{\text{RESET}}$ Pin Reset

For most applications a resistor connected between VDD and the $\overline{\text{RESET}}$ pin and a capacitor connected between VSS and the $\overline{\text{RESET}}$ pin will provide a suitable external reset circuit. Any wiring connected to the $\overline{\text{RESET}}$ pin should be kept as short as possible to minimise any stray noise interference. For applications that operate within an environment where more noise is present the Enhanced Reset Circuit shown is recommended.



Basic Reset Circuit



Enhanced Reset Circuit

This type of reset occurs when the microcontroller is already running and the $\overline{\text{RESET}}$ pin is forcefully pulled low by external hardware such as an external switch. In this case as in the case of other resets, the Program Counter will reset to zero and program execution initiated from this point. Note that, during the power-up sequence, the reset circuit should make sure that the external reset to be released after the internal power-on reset is over plus a suitable delay time. To improve the noise immunity, the low portion of external reset signal must be greater than that specified by t_{RES} in the A.C. characteristics, for the internal logic to recognise a valid reset. When a $\overline{\text{RESET}}$ pin reset takes place, the XRSTF bit in the RSTSRC register will be set high to indicate this reset.

Low Voltage Reset – LVR

The microcontroller contains a low voltage reset circuit in order to monitor the supply voltage of the device and provide an MCU reset should the value fall below a certain predefined level.

The LVR function is always enabled with a specific LVR voltage, V_{LVR}. If the supply voltage of the device drops to within a range of 0.9V~V_{LVR} such as might occur when changing the battery in battery powered applications, the LVR will automatically reset the device internally and the LVRF bit in the RSTSRC register will also be set to1. For a valid LVR signal, a low voltage, i.e., a voltage in the range between 0.9V~V_{LVR} must exist for greater than the value t_{LVR} specified in the A.C. characteristics. If the low voltage state does not exceed this value, the LVR will ignore the low supply voltage and will not perform a reset function. The actual V_{LVR} value can be selected by the LVSn bits in the LVRCR register. If the LVS7~LVS0 bits are changed to some certain values by the environmental noise, the LVR will reset the device after 2~3 LIRC clock cycles. When this happens, the LRF bit in the RSTSRC register will be set to 1. After power on the register will have the value of 01010101B. Note that the LVR function will be automatically disabled when the device enters the power-down mode.

LVRCR Register
SFR Address: EAh

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~0 **LVS7~LVS0:** LVR Voltage Select control

01010101: 2.1V
00110011: 2.55V
10011001: 3.15V
10101010: 4.0V
Any other value: Generates MCU reset – register is reset to POR value

When an actual low voltage condition occurs, as specified by the above defined LVR voltage value, an MCU reset will be generated. The reset operation will be activated after 2~3 LIRC clock cycles. In this situation this register contents will remain the same after such a reset occurs.

Any register value, other than the four defined values above, will also result in the generation of an MCU reset. The reset operation will be activated after 2~3 LIRC clock cycles. However in this situation this register contents will be reset to the POR value.

Watchdog Reset

All devices contain a Watchdog Timer which is used as a protection feature. The Watchdog Timer has to be periodically cleared by the application program and prevented from overflowing during normal MCU operation. However should the program enter an endless loop or should external environmental conditions such as noise causes the device to jump to unpredicted program locations, the Watchdog Timer will overflow from FFFFh to 0000h, and generate an MCU reset. Refer to the Watchdog Timer section for more details regarding the Watchdog Timer operation.

When a Watchdog Reset occurs the WDTS bit in the IP0 register will be set to indicate the reset source. Note that this bit must be reset by the application program.

IP0 Register
SFR Address: B8h

Bit	7	6	5	4	3	2	1	0
Name	—	WDTS	PT2	PS0	PT1	PX1	PT0	PX0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 Unimplemented, read as “0”

Bit 6 **WDTS:** Watchdog timer reset indication flag
0: No Watchdog timer reset
1: Watchdog timer reset
This bit must be cleared by the application program as it will not be automatically cleared by hardware.

Bit 5 **PT2:** Timer 2 Interrupt priority
Described elsewhere

Bit 4 **PS0:** Serial Port 0 Interrupt priority
Described elsewhere

Bit 3 **PT1:** Timer 1 Interrupt priority
Described elsewhere

Bit 2 **PX1:** External interrupt 1 priority
Described elsewhere

Bit 1 **PT0:** Timer 0 Interrupt priority
Described elsewhere

Bit 0 **PX0:** External interrupt 0 priority
Described elsewhere

Comparator 0 Reset

Comparator 0 contains an output reset function which can provide a reset when the output of Comparator 0 changes state. The Comparator 0 reset function is enabled by setting the CP0RST bit in the CP0CR register. If the CP0RST is set high, the comparator 0 output bit, CP0OUT, will determine if a Comparator 0 reset is generated or not. The CP0RSTL bit determines which polarity of the CP0OUT bit generates the reset, The CMP0F bit in the RSTSRC register is used to indicate the Comparator 0 reset source.

CP0CR Register
SFR Address: DEh

Bit	7	6	5	4	3	2	1	0
Label	—	CP0ON	CP0POL	CP0OUT	CP0OS	CP0RSTL	CP0RST	—
R/W	—	R/W	R/W	R	R/W	R/W	R/W	—
POR	—	0	0	0	1	0	0	—

Bit 7 Unimplemented, read as “0”

Bit 6 **CP0ON:** Comparator 0 on/off bit
Described elsewhere

Bit 5 **CP0POL:** Comparator 0 output polarity
Described elsewhere

Bit 4 **CP0OUT:** Comparator 0 output bit
Described elsewhere

Bit 3 **CP0OS:** Comparator 0 output path selection
Described elsewhere

Bit 2 **CP0RSTL:** Comparator 0 output reset selection – CP0RST=1
0: CP0OUT=0 will reset MCU
1: CP0OUT=1 will reset MCU

Bit 1 **CP0RST:** Comparator 0 output reset MCU control
0: Disable
1: Enable

Bit 0 Unimplemented, read as “0”

Software Resets

There are three ways to generate Software Reset, each of which are generated by writing certain values to the SRST register, the WDTCR register or the LVRCR register.

Software Reset Summary

Software Reset Name	Register	Bit	Operation
SRST Register	SRST	SRSTREQ	Write two successive "1" values to this bit
WDTRCR Register	WDTCR	WE4~WE0	Write value other than "10101" or "01010"
LVRCR Register	LVRCR	LVS7~LVS0	Write value other than "01010101", "00110011", "011001" or "10101010"

SRST Register Software Reset

A software reset will be generated after two consecutive instructions to write a high value to the SRSTREQ bit in the SRST register. The same bit can be used to identify the reset source.

SRST Register
SFR Address: F7h

Bit	7	6	5	4	3	2	1	0
Label	—	—	—	—	—	—	—	SRSTREQ
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 Unimplemented, read as "0"

Bit 0 **SRSTREQ:** Software reset request.
Writing a '0' value to this bit will have no effect.
A single '1' value write to this bit will have no effect.
Two consecutive '1' value writes to this bit will generate a software reset.
Reading this bit can indicate the reset source:
0: No software reset
1: Software reset
This bit must be cleared by the application program as it will not be automatically cleared by hardware.

WDTCR Register Software Reset

A WDTCR software reset will be generated when a value other than "10101" or "01010", exist in the highest five bits of the WDTCR register. The WRF bit in the RSTSRC register will be set high when this occurs, thus indicating the generation of a WDTCR software reset.

WDTCR Register
SFR Address: 96h

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	—	—	WDTCS
R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W
POR	0	1	0	1	0	—	—	0

Bit 7~3 **WE4~WE0:** WDT function software control
10101: Disable
01010: Enable – default
Other values: Reset MCU
If the MCU reset is caused by WE[4:0] in WDTC software reset, the WRF flag of RSTSRC register will be set.

Bit 2~1 Unimplemented, read as "0"

Bit 0 **WDTCS:** Watchdog clock (f_{WDT}) select
Described elsewhere

LVRCR Register Software Reset

An LVRCR software reset will be generated when a value other than "01010101", "00110011", "10011001" and "10101010", exist in the LVRCR register. The LRF bit in the RSTSRC register will be set high when this occurs, thus indicating the generation of an LVRCR software reset. The LVRCR register value will be rest to a value of 01010101B after any reset other than the LVR reset, and will remain unchanged after an LVR reset or during a WDT time out in the Power-Down mode.

LVRCR Register
SFR Address: EAh

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~0 **LVS7~LVS0:** LVR Voltage Select control
01010101: 2.1V – default value
00110011: 2.55V
10011001: 3.15V
10101010: 4.0V
Any other value: Generates MCU reset – register is reset to POR value

ROM Code Check Reset

ID block addresses 0xF0~0xFF can be written into ROM codes such as the following table shows, or a value of FFH which means no ROM codes are written into these addresses. When reading the option table, the hardware will automatically compare with the ROM code pattern, if any one of the ID block addresses has a mismatch, the MCU will automatically reset and re-read the option table until all the ID block addresses are matched.

ID block address	ROM code
0xF0	01H/FFH
0xF1	23H/FFH
0xF2	45H/FFH
0xF3	67H/FFH
0xF4	89H/FFH
0xF5	ABH/FFH
0xF6	CDH/FFH
0xF7	EFH/FFH
0xF8	FEH/FFH
0xF9	DCH/FFH
0xFA	BAH/FFH
0xFB	98H/FFH
0xFC	76H/FFH
0xFD	54H/FFH
0xFE	32H/FFH
0xFF	10H/FFH

Reset Initial Conditions

The different types of reset described affect the reset flags in different ways. The following table indicates the way in which the various components of the microcontroller are affected after a power-on reset occurs.

Item	Condition After RESET
Program Counter	Reset to zero
Interrupts	All interrupts will be disabled
WDT	Clear after reset, WDT begins counting
Timer/Even Counters	Timer/Even Counters will be turned off
Input/Output Ports	I/O ports will be setup as a quasi-bidirection structure
Stack Pointer	Set to 007H value

The different kinds of resets all affect the internal registers of the microcontroller in different ways. To ensure reliable continuation of normal program execution after a reset occurs, it is important to know what condition the microcontroller is in after a particular reset occurs. The following table describes how each type of reset affects each of the microcontroller internal registers.

Register Name	Power-On Reset	External Reset	WDT Time-out Reset	Software Reset
Program Counter	0000h	0000h	0000h	0000h
P0	1111_1111b	1111_1111b	1111_1111b	1111_1111b
SP	0000_0111b	0000_0111b	0000_0111b	0000_0111b
DPL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPH	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
WDTREL	0000_0000b	0000_0000b	uuuu_uuuub	0000_0000b
PCON	0---_1000b	0---_1000b	0---_1000b	0---_1000b
TCON	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TMOD	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TL0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TMPRE	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P1	1111_1111b	1111_1111b	1111_1111b	1111_1111b
P0WAKE	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPS	----_---0b	----_---0b	----_---0b	----_---0b
DPC	----_0000b	----_0000b	----_0000b	----_0000b
WDTCR	0101_0--0b	0101_0--0b	0101_0--ub	0101_0--0b
S0CON	0000_0000b	0000_0000b	0000_0000b	0000_0000b
S0BUF	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IEN2	----_-000b	----_-000b	----_-000b	----_-000b
S1CON	0-00_0000b	0-00_0000b	0-00_0000b	0-00_0000b
S1BUF	0000_0000b	0000_0000b	0000_0000b	0000_0000b
S1RELL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P0M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P0M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P2	1111_1111b	1111_1111b	1111_1111b	1111_1111b
T3CON	0000_--00b	0000_--00b	0000_--00b	0000_--00b
TL3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SRCR	--00_0000b	--00_0000b	--00_0000b	--00_0000b
SPPRE	----_1111b	----_1111b	----_1111b	----_1111b
P1M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P1M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IEN0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IEN1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
S0RELL	1101_1001b	1101_1001b	1101_1001b	1101_1001b
P2M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P2M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P3	1111_1111b	1111_1111b	1111_1111b	1111_1111b
P4	1111_1111b	1111_1111b	1111_1111b	1111_1111b
TBCR	0-00_-111b	0-00_-111b	0-00_-111b	0-00_-111b
DACTRL	000-_--00b	000-_--00b	000-_--00b	000-_--00b
DAL	0000_----b	0000_----b	0000_----b	0000_----b
DAH	1000_0000b	1000_0000b	1000_0000b	1000_0000b

Register Name	Power-On Reset	External Reset	WDT Time-out Reset	Software Reset
P3M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P3M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP0	-000_0000b	-000_0000b	-100_0000b	-000_0000b
IP0H	--00_0000b	--00_0000b	--00_0000b	--00_0000b
S0RELH	----_--11b	----_--11b	----_--11b	----_--11b
S1RELH	----_--11b	----_--11b	----_--11b	----_--11b
CPHCR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CPICR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IRCON2	----_0000b	----_0000b	----_0000b	----_0000b
IRCON	0000_000-b	0000_000-b	0000_000-b	0000_000-b
CCEN	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCL2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCL3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
T2CON	-000_0000b	-000_0000b	-000_0000b	-000_0000b
IEN3	----_0000b	----_0000b	----_0000b	----_0000b
CRCL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CRCH	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TL2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP3	----_0000b	----_0000b	----_0000b	----_0000b
IP3H	----_0000b	----_0000b	----_0000b	----_0000b
PSW	0000_0000b	0000_0000b	0000_0000b	0000_0000b
I2CCON	-000_00--b	-000_00--b	-000_00--b	-000_00--b
P5	1111_1111b	1111_1111b	1111_1111b	1111_1111b
I2CDAT	0000_0000b	0000_0000b	0000_0000b	0000_0000b
I2CADR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SBRCON	00--_----b	00--_----b	00--_----b	00--_----b
I2CSTA	1111_1---b	1111_1---b	1111_1---b	1111_1---b
CP0CR	-000_100-b	-000_100-b	-000_100-b	-000_100-b
CP1CR	-000_1---b	-000_1---b	-000_1---b	-000_1---b
ACC	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SPSTA	0000_----b	0000_----b	0000_----b	0000_----b
FMSR	0---_0000b	0---_0000b	0---_0000b	0---_0000b
SPDAT	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP1	--00_0000b	--00_0000b	--00_0000b	--00_0000b
IP1H	--00_0000b	--00_0000b	--00_0000b	--00_0000b
IP2	----_-000b	----_-000b	----_-000b	----_-000b
IP2H	----_-000b	----_-000b	----_-000b	----_-000b
SPCON	0001_0100b	0001_0100b	0001_0100b	0001_0100b
I2CLK	0001_1001b	0001_1001b	0001_1001b	0001_1001b
LVRCR	0101_0101b	0101_0101b	0101_0101b	0101_0101b
LVDCR	0---_-000b	0---_-000b	0---_-000b	0---_-000b
SCCR	----_--00b	----_--00b	----_--00b	----_--00b
PLLCR	00-0_-000b	00-0_-000b	00-0_-000b	00-0_-000b

Register Name	Power-On Reset	External Reset	WDT Time-out Reset	Software Reset
LSOCR	---1_--0-b	---1_--0-b	---1_--0-b	---1_--0-b
HSOCR	--01_--01b	--01_--01b	--01_--01b	--01_--01b
B	0000_0000b	0000_0000b	0000_0000b	0000_0000b
ADCR0	0110_0000b	0110_0000b	0110_0000b	0110_0000b
ADCR1	00-0_0000b	00-0_0000b	00-0_0000b	00-0_0000b
ADCR2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
ADPGA	----_-000b	----_-000b	----_-000b	----_-000b
ADRL(ADRFS=0)	0000_----b	0000_----b	0000_----b	0000_----b
ADRH(ADRFS=0)	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SRST	----_---0b	----_---0b	----_---0b	----_---1b
FMCR	01--_-000b	01--_-000b	01--_-000b	01--_-000b
FMKEY	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMAR0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMAR1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMAR2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMDR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
T2CON1	---0_10--b	---u_10--b	---u_10--b	---u_10--b
RSTSRC	-000_0x01b	-000_0010b	-000_0000b	-000_0000b

Register Name	WDTCR Reset	LVR Reset	LVRCR Reset	Comparator0 Reset
Program Counter	0000h	0000h	0000h	0000h
P0	1111_1111b	1111_1111b	1111_1111b	1111_1111b
SP	0000_0111b	0000_0111b	0000_0111b	0000_0111b
DPL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPH	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
WDTREL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
PCON	0---_1000b	0---_1000b	0---_1000b	0---_1000b
TCON	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TMOD	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TL0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TMPRE	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P1	1111_1111b	1111_1111b	1111_1111b	1111_1111b
P0WAKE	0000_0000b	0000_0000b	0000_0000b	0000_0000b
DPS	----_---0b	----_---0b	----_---0b	----_---0b
DPC	----_0000b	----_0000b	----_0000b	----_0000b
WDTCR	0101_0--0b	0101_0--0b	0101_0--0b	0101_0--0b
S0CON	0000_0000b	0000_0000b	0000_0000b	0000_0000b
S0BUF	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IEN2	----_-000b	----_-000b	----_-000b	----_-000b
S1CON	0-00_0000b	0-00_0000b	0-00_0000b	0-00_0000b
S1BUF	0000_0000b	0000_0000b	0000_0000b	0000_0000b

Register Name	WDTCR Reset	LVR Reset	LVRCR Reset	Comparator0 Reset
S1RELL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P0M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P0M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P2	1111_1111b	1111_1111b	1111_1111b	1111_1111b
T3CON	0000_--00b	0000_--00b	0000_--00b	0000_--00b
TL3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SRCR	--00_0000b	--00_0000b	--00_0000b	--00_0000b
SPPRE	----_1111b	----_1111b	----_1111b	----_1111b
P1M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P1M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IEN0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IEN1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
S0RELL	1101_1001b	1101_1001b	1101_1001b	1101_1001b
P2M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P2M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P3	1111_1111b	1111_1111b	1111_1111b	1111_1111b
P4	1111_1111b	1111_1111b	1111_1111b	1111_1111b
TBCR	0-00_-111b	0-00_-111b	0-00_-111b	0-00_-111b
DACTRL	000-_--00b	000-_--00b	000-_--00b	000-_--00b
DAL	0000_----b	0000_----b	0000_----b	0000_----b
DAH	1000_0000b	1000_0000b	1000_0000b	1000_0000b
P3M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P3M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP0	-000_0000b	-000_0000b	-000_0000b	-000_0000b
IP0H	--00_0000b	--00_0000b	--00_0000b	--00_0000b
S0RELH	----_--11b	----_--11b	----_--11b	----_--11b
S1RELH	----_--11b	----_--11b	----_--11b	----_--11b
CPHCR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CPICR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IRCON2	----_0000b	----_0000b	----_0000b	----_0000b
IRCON	0000_000-b	0000_000-b	0000_000-b	0000_000-b
CCEN	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCL2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCL3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH3	0000_0000b	0000_0000b	0000_0000b	0000_0000b
T2CON	-000_0000b	-000_0000b	-000_0000b	-000_0000b
IEN3	----_0000b	----_0000b	----_0000b	----_0000b
CRCL	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CRCH	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TL2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
TH2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP3	----_0000b	----_0000b	----_0000b	----_0000b
IP3H	----_0000b	----_0000b	----_0000b	----_0000b
PSW	0000_0000b	0000_0000b	0000_0000b	0000_0000b
I2CCON	-000_00--b	-000_00--b	-000_00--b	-000_00--b

Register Name	WDTCR Reset	LVR Reset	LVRCR Reset	Comparator0 Reset
P5	1111_1111b	1111_1111b	1111_1111b	1111_1111b
I2CDAT	0000_0000b	0000_0000b	0000_0000b	0000_0000b
I2CADR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SBRCON	00--_----b	00--_----b	00--_----b	00--_----b
I2CSTA	1111_1---b	1111_1---b	1111_1---b	1111_1---b
CP0CR	-000_100-b	-000_100-b	-000_100-b	-000_100-b
CP1CR	-000_1---b	-000_1---b	-000_1---b	-000_1---b
ACC	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SPSTA	0000_----b	0000_----b	0000_----b	0000_----b
FMSR	0---_0000b	0---_0000b	0---_0000b	0---_0000b
SPDAT	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP1	--00_0000b	--00_0000b	--00_0000b	--00_0000b
IP1H	--00_0000b	--00_0000b	--00_0000b	--00_0000b
IP2	----_-000b	----_-000b	----_-000b	----_-000b
IP2H	----_-000b	----_-000b	----_-000b	----_-000b
SPCON	0001_0100b	0001_0100b	0001_0100b	0001_0100b
I2CLK	0001_1001b	0001_1001b	0001_1001b	0001_1001b
LVRCR	0101_0101b	uuuu_uuuub	0101_0101b	0101_0101b
LVDCR	0---_-000b	0---_-000b	0---_-000b	0---_-000b
SCCR	----_--00b	----_--00b	----_--00b	----_--00b
PLLCR	00-0_-000b	00-0_-000b	00-0_-000b	00-0_-000b
LSOCR	---1_--0-b	---1_--0-b	---1_--0-b	---1_--0-b
HSOCR	--01_--01b	--01_--01b	--01_--01b	--01_--01b
B	0000_0000b	0000_0000b	0000_0000b	0000_0000b
ADCR0	0110_0000b	0110_0000b	0110_0000b	0110_0000b
ADCR1	00-0_0000b	00-0_0000b	00-0_0000b	00-0_0000b
ADCR2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
ADPGA	----_-000b	----_-000b	----_-000b	----_-000b
ADRL(ADRFS=0)	0000_----b	0000_----b	0000_----b	0000_----b
ADRH(ADRFS=0)	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SRST	----_---0b	----_---0b	----_---0b	----_---0b
FMCR	01--_-000b	01--_-000b	01--_-000b	01--_-000b
FMKEY	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMAR0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMAR1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMAR2	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMDR	0000_0000b	0000_0000b	0000_0000b	0000_0000b
T2CON1	---u_10--b	---u_10--b	---u_10--b	---u_10--b
RSTSRC	-010_0000b	-000_0100b	-100_0000b	-000_1000b

Note: "-" not implement
"u" stands for "unchanged"
"x" stands for "unknown"

25 Interrupts

Interrupts are an important part of any microcontroller system. When an external event or an internal function such as a Timer/Event Counter or Time Base requires microcontroller attention, their corresponding interrupt will enforce a temporary suspension of the main program allowing the microcontroller to direct attention to their respective needs. These devices contain multiple external interrupt pins, while the internal interrupts are generated by the various functions such as Timer/Event Counters, Time Base, Comparator, LVD, I²C, SPI, UART and the A/D converter. In addition, the interrupt priority can be controlled using registers.

Interrupt Registers

Overall interrupt control, which means interrupt enabling, priority and request flag setting, is controlled using several registers. By controlling the appropriate enable bits in these registers each individual interrupt can be enabled or disabled. Also when an interrupt occurs, the corresponding request flag will be automatically set by the microcontroller. The global enable control bit if cleared to zero will disable all interrupts.

Overall interrupt control, which basically means the setting of request flags when certain microcontroller conditions occur and the setting of interrupt enable bits by the application program, is controlled by a series of registers, located in the Special Function Registers , as shown in the accompanying table. Each register contains a number of enable bits to enable or disable individual registers as well as interrupt flags to indicate the presence of an interrupt request.

Interrupt Register Bit Naming Conventions

Function	Enable Bit	Request Flag	Notes
Global	EAL	—	—
Comparator	ECMP	CMPF	Overall Comparator Interrupt
	CP0IEN	CP0IF	Comparator 0 Interrupt
	CP1IEN	CP1IF	Comparator 1 Interrupt
INTn Pin	EXn	IEn	n=0~1
		IEXn	n=2~6
A/D Converter	EADC	IADC	—
Time Base	ETB	TBF	—
I²C	EI2C	SI	—
SPI	ESPI	SPIF WCOL SSERR MODF	The same interrupt vector with INT2
LVD	ELVD	LVDF	—
UART n	ESn	RI0/TI0, RI1/TI1	n=0~1
Timer n	ETn	TFn	n=0~3
Timer 2 External Reload	EXEN2	EXF2	—

Interrupt Register Contents

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EAL	(WDT)	ET2	ES0	ET1	EX1	ET0	EX0
IEN1	EXEN2	(SWDT)	ET3	ECMP	EX6	EX5	EX4	EX3
IEN2	—	—	—	—	—	ES1	ELVD	EX2
IEN3	—	—	—	—	ETB	EADC	EI2C	ESPI
IRCON	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	—
IRCON2	—	—	—	—	LVDF	TBF	CMPF	IADC
S0CON	(SM0)	(SM1)	(SM20)	(REN0)	(TB80)	(RB80)	TI0	RI0
S1CON	(SM)	—	(SM21)	(REN1)	(TB81)	(RB81)	TI1	RI1
TCON	TF1	(TR1)	TF0	(TR0)	IE1	IT1	IE0	IT0
T2CON	—	I3FR	I2FR	(T2R1)	(T2R0)	(T2CM)	(T2I1)	(T2I0)
T3CON	(GATE3)	(C/T3)	(T3M1)	(T3M0)	—	—	TF3	(TR3)
SPSTA	SPIF	WCOL	SSERR	MODF	—	—	—	—
CPICR	CP1IF	CP1IEN	CP1P1	CP1P0	CP0IF	CP0IEN	CP0P1	CP0P0
I2CCON	—	(ENSI)	(STA)	(STO)	SI	(AA)	—	—

Note: The bits in brackets are used to manage other functions and not related to the interrupt control.

IEN0 Register
SFR Address: A8h

Bit	7	6	5	4	3	2	1	0
Name	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **EAL:** Master interrupt global enable
0: Disable
1: Enable

Bit 6 **WDT:** Watchdog timer refresh flag
Described elsewhere

Bit 5 **ET2:** Timer 2 interrupt enable
0: Disable
1: Enable

Bit 4 **ES0:** UART0 interrupt enable
0: Disable
1: Enable

Bit 3 **ET1:** Timer 1 interrupt enable
0: Disable
1: Enable

Bit 2 **EX1:** External interrupt 1 enable
0: Disable
1: Enable

Bit 1 **ET0:** Timer 0 interrupt enable
0: Disable
1: Enable

Bit 0 **EX0:** External interrupt 0 enable
0: Disable
1: Enable

IEN1 Register
SFR Address: A9h

Bit	7	6	5	4	3	2	1	0
Name	EXEN2	SWDT	ET3	ECMP	EX6	EX5	EX4	EX3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **EXEN2:** Timer 2 external reload interrupt enable
0: Disable
1: Enable

Bit 6 **SWDT:** Watchdog timer start/refresh flag
Described elsewhere

Bit 5 **ET3:** Timer3 interrupt enable
0: Disable
1: Enable

Bit 4 **ECMP:** Comparator overall interrupt enable
0: Disable
1: Enable

Bit 3 **EX6:** External interrupt 6 enable
0: Disable
1: Enable

Bit 2 **EX5:** External interrupt 5 enable
0: Disable
1: Enable

Bit 1 **EX4:** External interrupt 4 enable
0: Disable
1: Enable

Bit 0 **EX3:** External interrupt 3 enable
0: Disable
1: Enable

IEN2 Register
SFR Address: 9Ah

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	ES1	ELVD	EX2
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 Unimplemented, read as “0”

Bit 2 **ES1:** UART1 interrupt enable
0: Disable
1: Enable

Bit 1 **ELVD:** LVD interrupt enable
0: Disable
1: Enable

Bit 0 **EX2:** External interrupt 2 enable
0: Disable
1: Enable

IEN3 Register
SFR Address: C9h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	ETB	EADC	EI2C	ESPI
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as "0"

Bit 3 **ETB:** Time Base interrupt enable
0: Disable
1: Enable

Bit 2 **EADC:** ADC interrupt enable
0: Disable
1: Enable

Bit 1 **EI2C:** I^2C interrupt enable
0: Disable
1: Enable

Bit 0 **ESPI:** SPI interrupt enable
0: Disable
1: Enable

IRCON Register
SFR Address: C0h

Bit	7	6	5	4	3	2	1	0
Name	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
POR	0	0	0	0	0	0	0	—

Bit 7 **EXF2:** Timer 2 external reload interrupt request flag
0: No request
1: Interrupt request
The EXF2 bit will be set high by a negative transition on the T2EX pin. This bit must be cleared using the application program. The EXF2 bit will be invalid in the Timer 2 Timer/Counter mode.

Bit 6 **TF2:** Timer 2 overflow interrupt request flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

Bit 5 **IEX6:** External interrupt 6 interrupt request flag
0: No request
1: Interrupt request
This bit is triggered by rising edge of external interrupt INT6. The IEX6 flag also will be set high when Timer 2 compare mode is enabled and counter value (TH2, TL2) is equal to Compare/Capture register 3 (CCH3, CCL3). Once the program into the interrupt subroutine, the IEX6 flag will be cleared by hardware automatically.

Bit 4 **IEX5:** External interrupt 5 interrupt request flag
0: No request
1: Interrupt request
This bit is triggered by rising edge of external interrupt INT5. The IEX5 flag also will be set high when Timer 2 compare mode is enabled and counter value (TH2, TL2) is equal to Compare/Capture register 2 (CCH2, CCL2). Once the program into the interrupt subroutine, the IEX5 flag will be cleared by hardware automatically.

Bit 3 **IEX4:** External interrupt 4 interrupt request flag
0: No request
1: Interrupt request
This bit is triggered by rising edge of external interrupt INT4. The IEX4 flag also will be set high when Timer 2 compare mode is enabled and counter value (TH2, TL2) is equal to Compare/Capture register 1 (CCH1, CCL1). Once the program into the interrupt subroutine, the IEX4 flag will be cleared by hardware automatically.

Bit 2 **IEX3:** External interrupt 3 interrupt request flag
0: No request
1: Interrupt request
This bit is triggered by falling or rising edge of external interrupt INT3. The IEX3 flag also will be set high when Timer 2 compare mode is enabled and counter value (TH2, TL2) is equal to Compare/Reload/Capture register (CRCH, CRCL). Once the program into the interrupt subroutine, the IEX3 flag will be cleared by hardware automatically.

Bit 1 **IEX2:** External interrupt 2 interrupt request flag
0: No request
1: Interrupt request
This bit is triggered by falling or rising edge of external interrupt INT2. This bit will be cleared by hardware automatically.

Bit 0 Unimplemented, read as "0"

IRCON2 Register
SFR Address: BFh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	LVDF	TBF	CMPF	IDAC
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as "0"

Bit 3 **LVDF:** LVD interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 2 **TBF:** Time Base interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 1 **CMPF:** Comparator overall interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 0 **IADC:** ADC interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

S0CON Register
SFR Address: 98h

Bit	7	6	5	4	3	2	1	0
Name	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **SM0~SM1:** UART 0 mode select bits
Described elsewhere

Bit 5 **SM20:** Multiprocessor communication enable control
Described elsewhere

Bit 4 **REN0:** UART 0 serial data reception enable
Described elsewhere

Bit 3 **TB80:** UART 0 Ninth Transmit bit assignment
Described elsewhere

Bit 2 **RB80:** UART 0 Ninth Receive bit assignment
Described elsewhere

Bit 1 **TI0:** UART 0 transmit interrupt flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

Bit 0 **RI0:** UART 0 receive interrupt flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

S1CON Register
SFR Address: 9Bh

Bit	7	6	5	4	3	2	1	0
Name	SM	—	SM21	REN1	TB81	RB81	TI1	RI1
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	—	0	0	0	0	0	0

Bit 7 **SM:** UART 1 operating mode select bit
Described elsewhere

Bit 6 Unimplemented, read as "0"

Bit 5 **SM21:** Multiprocessor communication enable control
Described elsewhere

Bit 4 **REN1:** UART 1 serial data reception enable
Described elsewhere

Bit 3 **TB81:** UART 1 Ninth Transmit bit assignment
Described elsewhere

Bit 2 **RB81:** UART 1 Ninth Receive bit assignment
Described elsewhere

Bit 1 **TI1:** UART 1 transmit interrupt flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

Bit 0 **RI1:** UART 1 receive interrupt flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

TCON Register
SFR Address: 88h

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **TF1:** Timer 1 interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 6 **TR1:** Timer 1 Run control
Described elsewhere

Bit 5 **TF0:** Timer 0 interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 4 **TR0:** Timer 0 Run control
Described elsewhere

Bit 3 **IE1:** External interrupt 1 request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 2 **IT1:** External interrupt 1 type control
0: Falling Edge
1: Low Level

Bit 1 **IE0:** External interrupt 0 request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 0 **IT0:** External interrupt 0 type control
0: Falling Edge
1: Low Level

T2CON Register
SFR Address: C8h

Bit	7	6	5	4	3	2	1	0
Name	—	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 Unimplemented, read as "0"

Bit 6 **I3FR:** Active edge selection for external interrupt “INT3” and PCA module 0 Compare and Capture functions
0: Falling edge
1: Rising edge
This bit is used to select the external interrupt triggered edge for INT3, the PCA Module 0 Compare mode output interrupt triggered edge and the PCA Module 0 Capture mode input triggered edge. Once the compare mode is enabled, the PCA interrupt will replace the external interrupt. When Timer 2 is selected as compare mode 0, the I3FR bit is recommended to be set high by firmware.

Bit 5 **I2FR:** Active edge selection for external interrupt “INT2”
0: Falling edge
1: Rising edge

Bit 4~3 **T2R1, T2R0:** Timer 2 reload mode selection
Described elsewhere

Bit 2 **T2CM:** Timer 2 Compare mode selection
Described elsewhere

Bit 1~0 **T2I1, T2I0:** Timer 2 input selection
Described elsewhere

T3CON Register
SFR Address: A1h

Bit	7	6	5	4	3	2	1	0
Name	GATE3	C/T3	T3M1	T3M0	—	—	TF3	TR3
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

Bit 7 **GATE3:** Timer 3 Gate Control
Described elsewhere

Bit 6 **C/T3:** Timer 3 Counter/Timer selection
Described elsewhere

Bit 5~4 **T3M1, T3M0:** Timer 3 mode selection
Described elsewhere

Bit 3~2 Unimplemented, read as "0"

Bit 1 **TF3:** Timer 3 interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 0 **TR3:** Timer 3 run flag
Described elsewhere

SPSTA Register
SFR Address: E1h

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	SSERR	MODF	—	—	—	—
R/W	R/W	R/W	R/W	R/W	—	—	—	—
POR	0	0	0	0	—	—	—	—

Bit 7 **SPIF:** SPI Transmit/Receive Complete flag

0: Data is being transferred

1: SPI data transmission completed

The SPIF bit is the Transmit/Receive Complete flag and is set high automatically when an SPI data transmission is completed, it must be cleared using the application program. The SPIF bit can be also cleared by hardware when the data transfer is in progress. It can also be used to generate an interrupt.

Bit 6 **WCOL:** SPI Write Collision flag

0: No collision

1: Collision

The WCOL flag is used to detect if a data collision has occurred. If this bit is high it means that data has been attempted to be written to the SPDAT register during a data transfer operation. An SPI interrupt will occur if the SPI interrupt function is enabled. This write operation will be ignored if data is being transferred. It must be cleared using the application program.

Bit 5 **SSERR:** Synchronous Serial Slave Error Flag

0: No error

1: Error

This bit is set by hardware when the SSN pin input is selected to disable the Slave device status while the receive sequence is incomplete. A SPI interrupt will occur if the SPI interrupt function is enabled. This bit will be cleared by disabling the SPI module, clearing the SPEN bit in the SPCON register.

Bit 4 **MODF:** SPI Master/Slave Mode Mismatch Flag

0: No Mismatch

1: Mismatch

This bit is set by hardware when the Slave Select SSN pin level conflicts with actual Master/Slave mode of the SPI Master controller which is configured as a master while externally selected as a slave. A SPI interrupt will occur if the SPI interrupt function is enabled. It must be cleared using the application program.

Bit 3~0 Unimplemented, read as "0"

CPICR Register
SFR Address: BEh

Bit	7	6	5	4	3	2	1	0
Name	CP1IF	CP1IEN	CP1P1	CP1P0	CP0IF	CP0IEN	CP0P1	CP0P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CP1IF:** Comparator 1 Output Transition Interrupt Request Flag
0: No request
1: Interrupt request
This bit should be cleared using the application program.

Bit 6 **CP1IEN:** Comparator 1 Output Transition Interrupt Enable
0: Disable
1: Enable

Bit 5~4 **CP1P1, CP1P0:** Comparator 1 Output Transition Setting for interrupt request
00: Interrupt disabled
01: High to low
10: Low to high
11: High to low or low to high

Bit 3 **CP0IF:** Comparator 0 Output Transition Interrupt Request Flag
0: No request
1: Interrupt request
This bit should be cleared using the application program.

Bit 2 **CP0IEN:** Comparator 0 Output Transition Interrupt Enable
0: Disable
1: Enable

Bit 1~0 **CP0P1, CP0P0:** Comparator 0 Output Transition Setting for interrupt request
00: Interrupt disabled
01: High to low
10: Low to high
11: High to low or low to high

I2CCON Register
SFR Address: D8h

Bit	7	6	5	4	3	2	1	0
Name	—	ENS1	STA	STO	SI	AA	—	—
R/W	—	R/W	R/W	R/W	R	R	—	—
POR	—	0	0	0	0	0	—	—

Bit 7 Unimplemented, read as "0"

Bit 6 **ENS1:** I^{2}C Enable Control
Described elsewhere

Bit 5 **STA:** I^{2}C Start flag
Described elsewhere

Bit 4 **STO:** I^{2}C Stop flag
Described elsewhere

Bit 3 **SI:** I^{2}C Interrupt Request flag
0: No request
1: Interrupt request
The SI bit is set by hardware when one of the 25 possible I^{2}C states takes place. This bit must be cleared using the application program.

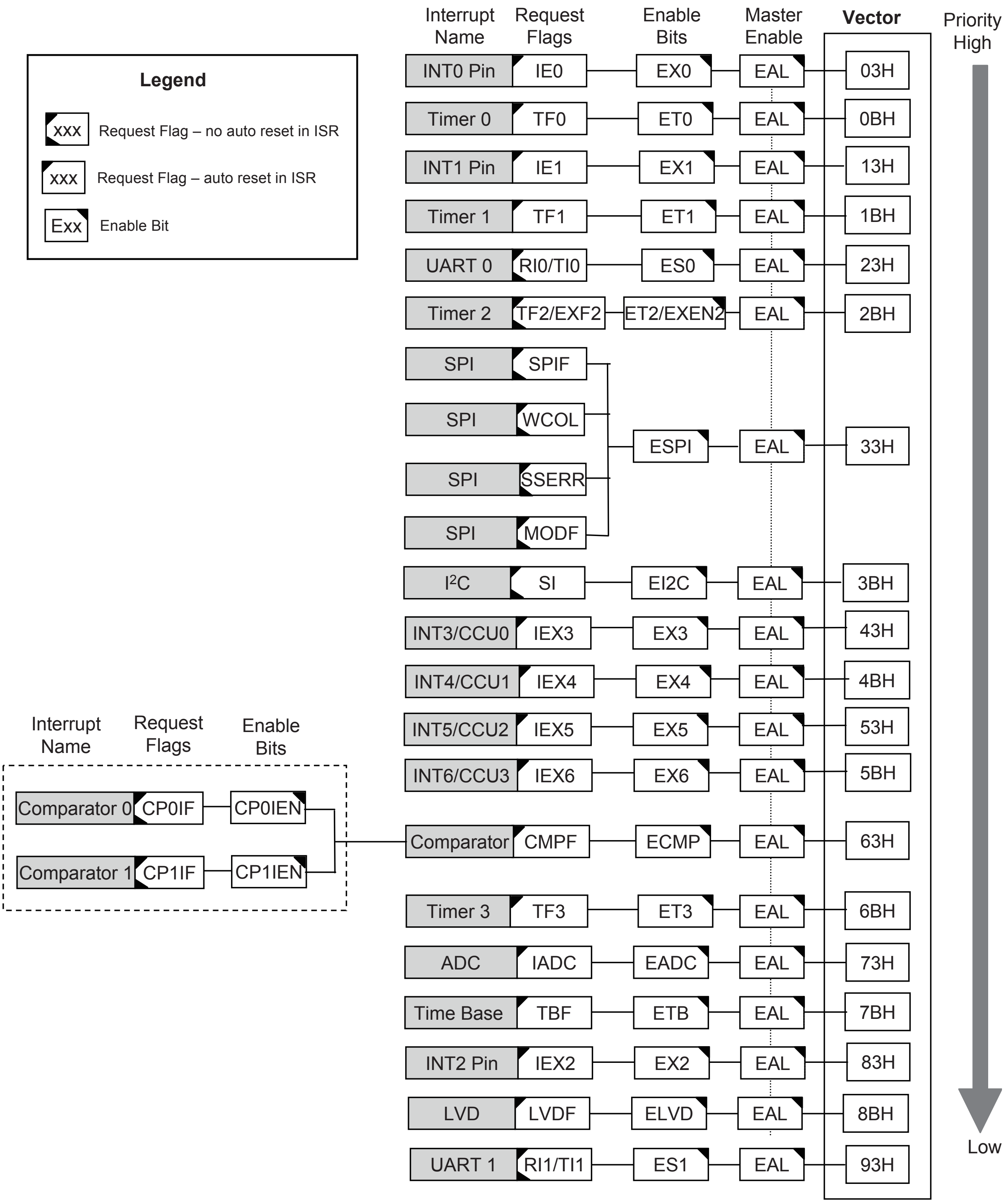
Bit 2 **AA:** I^{2}C Acknowledge Indication flag
Described elsewhere

Bit 1~0 Unimplemented, read as "0"

Interrupt Operation

A Timer Counter overflow, an active edge or level on the external interrupt pin, a comparator output changes state or A/D conversion completion etc, will all generate an interrupt request by setting their corresponding request flag. When this happens, if the interrupt enable bit is set, then the Program Counter, which stores the address of the next instruction to be executed, will be transferred onto the stack. The Program Counter will then be loaded with a new address which will be the value of the corresponding interrupt vector. The microcontroller will begin then fetch its next instruction from this interrupt vector. The instruction at this vector will jump to another section of program which is known as the interrupt service routine. Here is located the code to control the appropriate interrupt. The interrupt service routine must be terminated with a RETI instruction, which retrieves the original Program Counter address from the stack and allows the microcontroller to continue with normal execution at the point where the interrupt occurred.

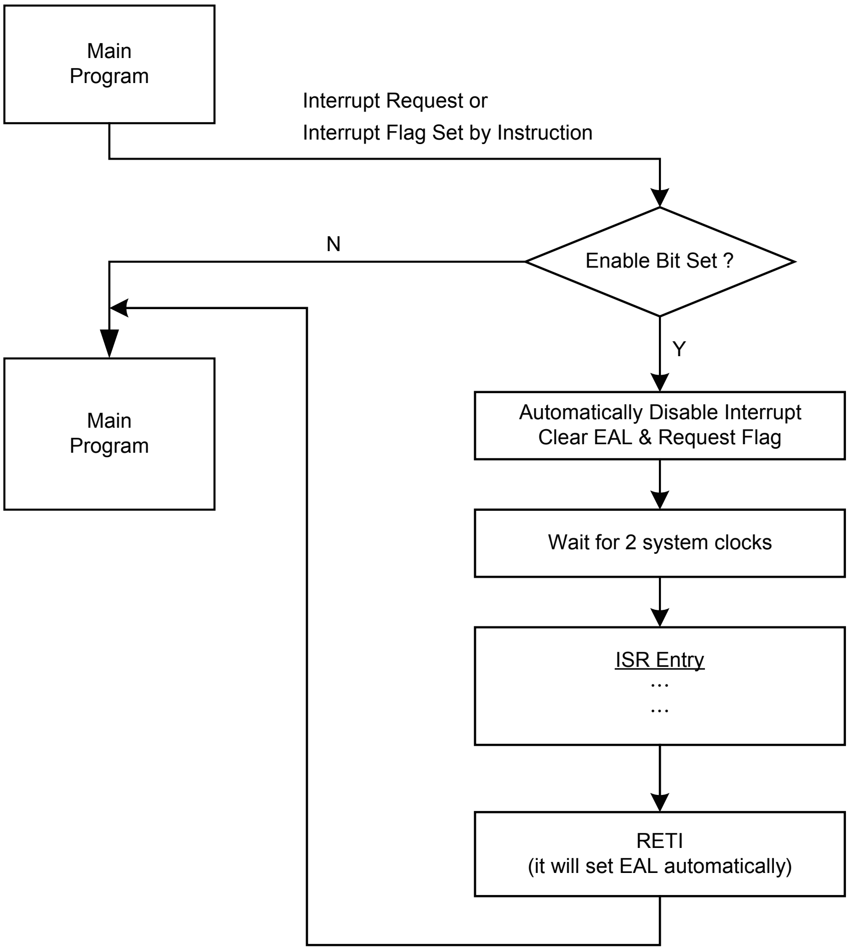
The various interrupt enable bits, together with their associated request flags, are shown in the following diagram with their order of priority. The interrupts are assigned into groups. Interrupts with higher priority can stop lower priority ones. All interrupts are categorised into 19 groups and 4 priority levels, setup using the IP0 and IP1 registers.



Interrupt Structure

Once an interrupt subroutine is serviced, all the other interrupts must be blocked by clearing the EAL bit using the application program. This will prevent any further interrupt nesting from occurring. However, if other interrupt requests occur during this interval, although the interrupt will not be immediately serviced, the request flag will still be recorded. If an interrupt requires immediate servicing while the program is already in another interrupt service routine, the EAL bit should be set after entering the routine, to allow interrupt nesting. If the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the Stack Pointer is decremented. If immediate service is desired, the stack must be prevented from becoming full.

When an interrupt request is generated, it takes several instruction cycles before the program jumps to the interrupt vector.



Interrupt Flowchart

Interrupt Priority

In case of simultaneous requests, the following table shows the priority that is applied. The interrupts can be assigned into groups. Higher priority interrupts can stop the lower priority interrupts. All interrupts are categorised into 19 groups with 4 priority levels. In cases where both higher priority and lower priority interrupts are enabled and where a higher priority and lower priority interrupt occurs simultaneously, the higher priority interrupt will always have priority and will therefore be serviced first. Suitable masking of the individual interrupts using the interrupt registers can prevent simultaneous occurrences. Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the interrupt priority registers: IP0, IP1, IP2, IP3, IP0H, IP1H, IP2H and IP3H. IP0, IP1, IP2 and IP3 hold the low order priority bits and IP0H, IP1H, IP2H and IP3H hold the high priority bits for each interrupt.

If the priority level is the same for different groups, then an internal polling sequence determines which interrupt request is serviced. The polling sequence is based on the vector address; an interrupt with a lower vector address has higher priority than an interrupt with a higher vector address. Note that the polling sequence is only used to resolve interrupt requests of the same priority level.

Priority Levels

The accompanying table illustrates the interrupt priority level assigned by the corresponding IPnH.x and IPn.x bits (n=0~3).

IPnH.x	IPn.x	Priority Level	Note
1	1	Level 3	Highest Priority
1	0	Level 2	↓
0	1	Level 1	↓
0	0	Level 0	Lowest Priority

If the interrupt levels are assigned the same priority level by the IPnH.x and IPn.x bits, the interrupt priority is followed by the accompanying table.

Interrupt Source	Interrupt Vector	Service Priority	Priority Control	Group Priority
Reset	0000H	Top	Always Highest	Highest Priority
External Interrupt 0 (INT0)	0003H	0	PX0H(IP0H.0)/PX0(IP0.0)	↓
Timer 0 Overflow Interrupt	000BH	1	PT0H(IP0H.1)/PT0(IP0.1)	↓
External Interrupt 1 (INT1)	0013H	2	PX1H(IP0H.2)/PX1(IP0.2)	↓
Timer 1 Overflow Interrupt	001BH	3	PT1H(IP0H.3)/PT1(IP0.3)	↓
Serial Port 0 Interrupt	0023H	4	PS0H(IP0H.4)/PS0(IP0.4)	↓
Timer 2 Overflow Interrupt or Timer 2 External Reload Interrupt	002BH	5	PT2H(IP0H.5)/PT2(IP0.5)	↓
Serial Peripheral Interface Interrupt	0033H	6	PSPIH(IP3H.0)/PSPI(IP3.0)	↓
I^{2}C Interrupt	003BH	7	PI2CH(IP3H.1)/PI2C(IP3.1)	↓
External Interrupt 3 (INT3) or CCU0 Interrupt	0043H	8	PX3H(IP1H.0)/PX3(IP1.0)	↓
External Interrupt 4 (INT4) or CCU1 Interrupt	004BH	9	PX4H(IP1H.1)/PX4(IP1.1)	↓
External Interrupt 5 (INT5) or CCU2 Interrupt	0053H	10	PX5H(IP1H.2)/PX5(IP1.2)	↓
External Interrupt 6 (INT6) or CCU3 Interrupt	005BH	11	PX6H(IP1H.3)/PX6(IP1.3)	↓
Comparator Interrupt (CMP0 & CMP1)	0063H	12	PCMPH(IP1H.4)/PCMP(IP1.4)	↓
Timer 3 Overflow Interrupt	006BH	13	PT3H(IP1H.5)/PT3(IP1.5)	↓
ADC End of Conversion Interrupt	0073H	14	PADCH(IP3H.2)/PADC(IP3.2)	↓
Time Base Overflow Interrupt	007BH	15	PTBH(IP3H.3)/PTB(IP3.3)	↓
External Interrupt 2 (INT2)	0083H	16	PX2H(IP2H.0)/PX2(IP2.0)	↓
LVD Interrupt	008BH	17	PLVDH(IP2H.1)/PLVD(IP2.1)	↓
Serial Port 1 Interrupt	0093H	18	PS1H(IP2H.2)/PS1(IP2.2)	Lowest Priority

Priority Control Registers

Low byte of Interrupt Priority Register 0: IP0
SFR Address: B8h

Bit	7	6	5	4	3	2	1	0
Name	—	WDTS	PT2	PS0	PT1	PX1	PT0	PX0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 Unimplemented, read as “0”

Bit 6 **WDTS:** Watchdog timer reset indication flag
Described elsewhere

Bit 5 **PT2:** Timer 2 Interrupt priority low
Low order bit for Timer 2 interrupt priority level.

Bit 4 **PS0:** UART 0 Interrupt priority low
Low order bit for UART 0 interrupt priority level.

Bit 3 **PT1:** Timer 1 Interrupt priority low
Low order bit for Timer 1 interrupt priority level.

Bit 2 **PX1:** External interrupt 1 priority low
Low order bit for External Interupt 1 interrupt priority level.

Bit 1 **PT0:** Timer 0 Interrupt priority low
Low order bit for Timer 0 interrupt priority level.

Bit 0 **PX0:** External interrupt 0 priority low
Low order bit for External Interupt 0 interrupt priority level.

High byte of Interrupt Priority Register 0: IP0H
SFR Address: B9h

Bit	7	6	5	4	3	2	1	0
Name	—	—	PT2H	PS0H	PT1H	PX1H	PT0H	PX0H
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5 **PT2H:** Timer 2 Interrupt priority high
High order bit for Timer 2 interrupt priority level.

Bit 4 **PS0H:** UART 0 Interrupt priority high
High order bit for UART 0 interrupt priority level.

Bit 3 **PT1H:** Timer 1 Interrupt priority high
High order bit for Timer 1 interrupt priority level.

Bit 2 **PX1H:** External interrupt 1 priority high
High order bit for External Interupt 1 interrupt priority level.

Bit 1 **PT0H:** Timer 0 Interrupt priority high
High order bit for Timer 0 interrupt priority level.

Bit 0 **PX0H:** External interrupt 0 priority high
High order bit for External Interupt 0 interrupt priority level.

Low byte of Interrupt Priority Register 1: IP1
SFR Address: E4h

Bit	7	6	5	4	3	2	1	0
Name	—	—	PT3	PCMP	PX6	PX5	PX4	PX3
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5 **PT3:** Timer 3 Interrupt priority low
Low order bit for Timer 3 interrupt priority level.

Bit 4 **PCMP:** Comparator Interrupt priority low
Low order bit for Comparator interrupt priority level.

Bit 3 **PX6:** External interrupt 6 Interrupt priority low
Low order bit for External interrupt 6 interrupt priority level.

Bit 2 **PX5:** External interrupt 5 priority low
Low order bit for External Interupt 5 interrupt priority level.

Bit 1 **PX4:** External interrupt 4 Interrupt priority low
Low order bit for External interrupt 4 interrupt priority level.

Bit 0 **PX3:** External interrupt 3 priority low
Low order bit for External Interupt 3 interrupt priority level.

High byte of Interrupt Priority Register 1: IP1H
SFR Address: E5h

Bit	7	6	5	4	3	2	1	0
Name	—	—	PT3H	PCMPH	PX6H	PX5H	PX4H	PX3H
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5 **PT3H:** Timer 3 Interrupt priority high
High order bit for Timer 3 interrupt priority level.

Bit 4 **PCMPH:** Comparator Interrupt priority high
High order bit for Comparator interrupt priority level.

Bit 3 **PX6H:** External interrupt 6 Interrupt priority high
High order bit for External interrupt 6 interrupt priority level.

Bit 2 **PX5H:** External interrupt 5 priority high
High order bit for External Interupt 5 interrupt priority level.

Bit 1 **PX4H:** External interrupt 4 Interrupt priority high
High order bit for External interrupt 4 interrupt priority level.

Bit 0 **PX3H:** External interrupt 3 priority high
High order bit for External Interupt 3 interrupt priority level.

Low byte of Interrupt Priority Register 2: IP2
SFR Address: E6h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PS1	PLVD	PX2
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 Unimplemented, read as “0”

Bit 2 **PS1:** UART 1 priority low
Low order bit for UART 1 interrupt priority level.

Bit 1 **PLVD:** LVD Interrupt priority low
Low order bit for LVD interrupt priority level.

Bit 0 **PX2:** External interrupt 2 priority low
Low order bit for External Interupt 2 interrupt priority level.

High byte of Interrupt Priority Register 2: IP2H
SFR Address: E7h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PS1H	PLVDH	PX2H
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 Unimplemented, read as “0”

Bit 2 **PS1H:** UART 1 priority high
High order bit for UART 1 interrupt priority level.

Bit 1 **PLVDH:** LVD Interrupt priority high
High order bit for LVD interrupt priority level.

Bit 0 **PX2H:** External interrupt 2 priority high
High order bit for External Interupt 2 interrupt priority level.

Low byte of Interrupt Priority Register 3: IP3
SFR Address: CEh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PTB	PADC	PI2C	PSPI
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as “0”

Bit 3 **PTB:** Time Base Interrupt Priority low
Low order bit for Time Base Interrupt Priority level.

Bit 2 **PADC:** ADC Interrupt priority low
Low order bit for ADC interrupt priority level.

Bit 1 **PI2C:** I^{2}C Interrupt priority low
Low order bit for I^{2}C interrupt priority level.

Bit 0 **PSPI:** SPI Interrupt priority low
Low order bit for SPI interrupt priority level.

How byte of Interrupt Priority Register 3: IP3H
SFR Address: CFh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PTBH	PADCH	PI2CH	PSPIH
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as "0"

Bit 3 **PTBH:** Time Base Interrupt Priority high
High order bit for Time Base Interrupt Priority level.

Bit 2 **PADCH:** ADC Interrupt priority high
High order bit for ADC interrupt priority level.

Bit 1 **PI2CH:** I^{2}C Interrupt priority high
High order bit for I^{2}C interrupt priority level.

Bit 0 **PSPIH:** SPI Interrupt priority high
High order bit for SPI interrupt priority level.

External Interrupt

The external interrupt pins are pin-shared with the I/O pins and can be configured as an external interrupt pin if the corresponding external interrupt enable bits in the interrupt control registers have been set. The pin must also be setup as an input by setting the corresponding bits in the port mode register. Any pull-high resistor settings will also remain valid when the pin is used as an external interrupt pin. When the interrupt is enabled, the stack is not full and a falling edge, a rising edge or a high to low level transition appears on the external interrupt pin, a subroutine call to the external interrupt vector, will take place. When the interrupt is serviced, the external interrupt request flag will be automatically reset and the EAL bit must be cleared by the application program to disable other interrupts.

The IT0, IT1, I2FR and I3FR bits are used to select the type of active edge that will trigger the external interrupt for INT0, INT1, INT2 and INT3 respectively. The other external interrupts, INT4, INT5 and INT6, are triggered with a rising edge signal.

External Interrupt Trigger Type

Ext Int	Trigger Type	Register	Bit
INT0	Falling Edge or Low Level	TCON	IT0
INT1	Falling Edge or Low Level	TCON	IT1
INT2	Falling Edge or Rising Edge	T2CON	I2FR
INT3	Falling Edge or Rising Edge	T2CON	I3FR
INT4	Rising Edge	—	—
INT5	Rising Edge	—	—
INT6	Rising Edge	—	—

Comparator Interrupt

The comparator interrupts are controlled by the two internal comparators. A comparator interrupt request will take place when the comparator interrupt request flag, CPnIF, is set, a situation that will occur when one of the comparator output bits changes state. This will in turn cause the comparator overall request flag, CMPF, to go high. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL, individual comparator enable bit, CPnIEN, and overall comparator interrupt enable bit, ECMP, must first be set. When the interrupt is enabled, the stack is not full and the comparator inputs generate a comparator output transition, a subroutine call to the comparator interrupt vector, will take place. When the interrupt is serviced, the CP0IF and CP1IF bits can be examined to determine whether the interrupt was generated by Comparator 0 or Comparator 1. In addition, the comparator output transition interrupt can be set up by the CPICR control register. Note that the comparator overall request flag, CMPF, will be automatically cleared, however the individual comparator interrupt request flags, CPnIF, must be cleared by the application program. The EAL bit must be cleared by the application program to disable other interrupts when in the interrupt routine.

CPICR Register
SFR Address: BEh

Bit	7	6	5	4	3	2	1	0
Name	CP1IF	CP1IEN	CP1P1	CP1P0	CP0IF	CP0IEN	CP0P1	CP0P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CP1IF:** Comparator 1 Output Transition Interrupt Request Flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

Bit 6 **CP1IEN:** Comparator 1 Output Transition Interrupt Enable
0: Disable
1: Enable

Bit 5~4 **CP1P1, CP1P0:** Comparator 1 Output Transition Setting for interrupt request
00: Interrupt disabled
01: High to low
10: Low to high
11: High to low or low to high

Bit 3 **CP0IF:** Comparator 0 Output Transition Interrupt Request Flag
0: No request
1: Interrupt request
This bit must be cleared using the application program.

Bit 2 **CP0IEN:** Comparator 0 Output Transition Interrupt Enable
0: Disable
1: Enable

Bit 1~0 **CP0P1, CP0P0:** Comparator 0 Output Transition Setting for interrupt request
00: Interrupt disabled
01: High to low
10: Low to high
11: High to low or low to high

A/D Converter Interrupt

The A/D Converter Interrupt is controlled by the termination of an A/D conversion process. An A/D Converter Interrupt request will take place when the A/D Converter Interrupt request flag, IADC, is set, which occurs when the A/D conversion process finishes. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL, and A/D Interrupt enable bit, EADC, must first be set. When the interrupt is enabled, the stack is not full and the A/D conversion process has ended, a subroutine call to the A/D Converter Interrupt vector, will take place. When the interrupt is serviced, the A/D Converter Interrupt flag, IADC, will be automatically cleared. The EAL bit must be cleared by the application program to disable other interrupts.

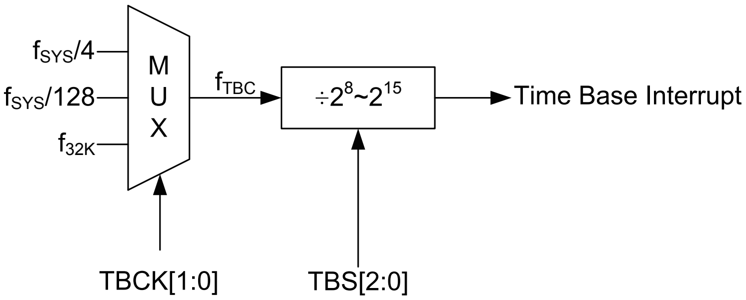
Timer/Counter Interrupt

For a Timer Counter interrupt to occur, the global interrupt enable bit, EAL, and the corresponding timer interrupt enable bit, ETn, must first be set. An actual Timer Counter interrupt will take place when the Timer Counter request flag, TFn, is set, a situation that will occur when the relevant Timer Counter overflows. When the interrupt is enabled, the stack is not full and a Timer Counter n overflow occurs, a subroutine call to the relevant timer interrupt vector, will take place. When the interrupt is serviced, the timer interrupt request flag, TFn (n=0, 1, 3), will be automatically reset, while the TF2 bit must be cleared by the application program, and the EAL bit must also be cleared using the application program to disable other interrupts.

Time Base Interrupts

The function of the Time Base Interrupt is to provide a regular time signal in the form of an internal interrupt. It is basically a simple timer whose interrupt is generated when it overflows. When this happen its respective interrupt request flag, TBF will be set. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL and Time Base enable bit, ETB, must first be set. When the interrupt is enabled, the stack is not full and the Time Base overflows, a subroutine call to its respective vector locations will take place. When the interrupt is serviced, the respective interrupt request flag, TBF, will be automatically reset but the EAL bit must be cleared by the application program to disable other interrupts.

The purpose of the Time Base Interrupt is to provide an interrupt signal at fixed time periods. Their clock sources originate from the internal clock source f_{TBC}. This f_{TBC} input clock passes through a divider, the division ratio of which is selected by programming the appropriate bits in the TBCR register to obtain longer interrupt periods. The clock source that generates f_{TBC}, which in turn controls the Time Base interrupt period, can originate from the system clock, LIRC or LXT oscillator.



Time Base Clock Source Select

TBCR Register
SFR Address: B2h

Bit	7	6	5	4	3	2	1	0
Name	TBEN	—	TBCK1	TBCK0	—	TBS2	TBS1	TBS0
R/W	R/W	—	R/W	R/W	—	R/W	R/W	R/W
POR	0	—	0	0	—	1	1	1

Bit 7 **TBEN:** TB Control bit
0: Disable
1: Enable

Bit 6 Unimplemented, read as “0”

Bit 5~4 **TBCK1~TBCK0:** Select Time Base clock source, f_{TBC}
00: $f_{SYS}/4$
01: $f_{SYS}/128$
1x: f_{32K} (f_{32K} is sourced from f_{LIRC} or f_{LXT})

Bit 3 Unimplemented, read as “0”

Bit 2~0 **TBS2~TBS0:** Select Time Base Time-out Period
000: $256/f_{TBC}$
001: $512/f_{TBC}$
010: $1024/f_{TBC}$
011: $2048/f_{TBC}$
100: $4096/f_{TBC}$
101: $8192/f_{TBC}$
110: $16384/f_{TBC}$
111: $32768/f_{TBC}$ (default setting)

I²C Interface Interrupt

An I²C Interrupt request will take place when the I²C Interrupt request flag, SI, is set, which occurs when one of the 25 possible I²C states takes place. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL, and the I²C Interface Interrupt enable bit, EI2C, must first be set. When the interrupt is enabled, the stack is not full and a byte of data has been compared match with the I²C states, a subroutine call to the respective Interrupt vector, will take place. When the I²C Interface Interrupt is serviced, the EAL bit must be cleared by the application program to disable other interrupts, and the SI flag also must be cleared using the application program.

SPI Interface Interrupt

A SPI Interrupt request will take place when one of the SPI Interrupt request flags, SPIF, WCOL, SSERR or MODF, is set, which occurs when a byte of data has been received or when there is a write collision or when there is a Serial Slave error or transmitted by the SPI interface or the Master mode or Slave mode is mismatched with the mode selected input pin level. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL, and the Serial Interface Interrupt enable bit, ESPI, must first be set. When the interrupt is enabled, the stack is not full and a byte of data has been transmitted or received by the SPI interface, or the Mode mismatch, a subroutine call to the respective Interrupt vector, will take place. When the SPI Interface Interrupt is serviced, the EAL bit must be cleared by the application program to disable other interrupts, and the SPIF, WCOL, SSERR and MODF flags also must be cleared using the application program.

UART Interface Interrupt

A UARTn Interrupt request will take place when the UARTn Interrupt request flags, RIn or TIn, is set, which occurs when a byte of data has been received or transmitted by the UARTn interface. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL, and the UART Interrupt enable bit, ESn, must first be set. When the interrupt is enabled, the stack is not full and a byte of data has been transmitted or received by the UARTn interface, will take place. When the UARTn Interface Interrupt is serviced, the EAL bit must be cleared by the application program to disable other interrupts, and the RIn or TIn flag also must be cleared using the application program.

LVD Interrupt

A LVD Interrupt request will take place when the LVD Interrupt request flag, LVDF, is set, which occurs when the Low Voltage Detector function detects a low power supply voltage. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EAL, Low Voltage Interrupt enable bit, ELVD, must first be set. When the interrupt is enabled, the stack is not full and a low voltage condition occurs, a subroutine call to the Interrupt vector, will take place. When the Low Voltage Interrupt is serviced, the EAL bit will be automatically cleared to disable other interrupts and the LVDF flag will be automatically cleared.

Interrupt Wake-up Function

Each of the interrupt functions has the capability of waking up the microcontroller when in the IDLE mode, and only INT0 and INT1 interrupts can wake up the microcontroller when in the Power-down mode. A wake-up is generated when an interrupt request flag changes from low to high and is independent of whether the interrupt is enabled or not. Therefore, even though the device is in the Power-Down or IDLE Mode and the CPU clock stopped, situations such as external edge transitions on the external interrupt pins, a low power supply voltage or comparator input change may cause their respective interrupt flag to be set high and consequently generate an interrupt. Care must therefore be taken if spurious wake-up situations are to be avoided. If an interrupt wake-up function is to be disabled then the corresponding interrupt request flag should be set high before the device enters the Power-Down or IDLE Mode. The interrupt enable bits have no effect on the interrupt wake-up function.

Programming Considerations

By disabling the interrupt enable bits, a requested interrupt can be prevented from being serviced, however, once an interrupt request flag is set, it will remain in this condition in the interrupt register until the corresponding interrupt is serviced or until the request flag is cleared by the application program.

It is recommended that programs do not use the “CALL subroutine” instruction within the interrupt subroutine. Interrupts often occur in an unpredictable manner or need to be serviced immediately in some applications. If only one stack is left and the interrupt is not well controlled, the original control sequence will be damaged once a “CALL” subroutine is executed in the interrupt subroutine.

All these interrupt functions have the capability of waking up the microcontroller when in the IDLE mode, only INT0 and INT1 interrupts can wake up the microcontroller when in the Power-down mode.

Only the Program Counter is pushed onto the stack. If the contents of the register or status register are altered by the interrupt service program, which may corrupt the desired control sequence, then the contents should be saved in advance.

26 Input/Output Ports

The devices offer a range of flexible options on their I/O ports. Many pins can be setup with a choice of different register controlled modes as well as having pull-high wake up and slew rate functions.

Input/Output Port Overview

The devices are provided with a series bidirectional input/output ports labeled with port names P0~P5. These I/O ports are mapped to the Special Function Registers with specific addresses as shown in the Special Function Registers table. All of these I/O ports can be used for both input and output operations, the data for which is stored in Port Data Registers. Ports P0~P3 can be setup using Port Mode Registers to operate in a series of different modes. Ports 4 and 5 can only operate in the traditional 8051 type quasi-bidirectional mode. The Port P0 provides register controlled wake up function as well. Bit manipulation instructions can be used to control Ports P0~P3, while Ports 4 and 5 must be controlled using byte wide instructions.

I/O Port Function Summary

Function.	Port Number						
	P0	P1	P2	P3	P4	P5	Notes
Push-Pull	√	√	√	√	—	—	CMOS Output
Open Drain	√	√	√	√	—	—	NMOS
Quasi Bi-direct	√	√	√	√	√	√	Traditional 8051 Port type
Input Only	√	√	√	√	—	—	High impedance
Bit Addressable	√	√	√	√	—	—	—
Slew Rate Control	√	√	√	√	√	√	Fast or Slow select

Register Description

This section provides a description of all the registers associated with I/O setup and control. The following table gives a summary of all associated I/O registers, which will be described in detail later.

I/O Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
P0WAKE	P07WU	P06WU	P05WU	P04WU	P03WU	P02WU	P01WU	P00WU
P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P0M0	P0M0.7	P0M0.6	P0M0.5	P0M0.4	P0M0.3	P0M0.2	P0M0.1	P0M0.0
P0M1	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P1M0	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0
P1M1	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P2M0	P2M0.7	P2M0.6	P2M0.5	P2M0.4	P2M0.3	P2M0.2	P2M0.1	P2M0.0
P2M1	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0
P3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P3M0	P3M0.7	P3M0.6	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0
P3M1	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0
P4	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
P5	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
SRCR	—	—	SRCR.5	SRCR.4	SRCR.3	SRCR.2	SRCR.1	SRCR.0

Each Port has its own data register, known as P0, P1, P2, P3, P4 and P5 which are used to control the input and output I/O pin data. These registers read input pin data or write output pin data on the selected I/O pin. For I/O pins setup as outputs a read operation to these registers will setup either a high or low level on the corresponding pin. For I/O pins setup as inputs a read operation to these registers will read the actual logic level on the corresponding pin.

P0 Register
SFR Address: 80h

Bit	7	6	5	4	3	2	1	0
Name	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

P1 Register
SFR Address: 90h

Bit	7	6	5	4	3	2	1	0
Name	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

P2 Register
SFR Address: A0h

Bit	7	6	5	4	3	2	1	0
Name	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

P3 Register
SFR Address: B0h

Bit	7	6	5	4	3	2	1	0
Name	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

P4 Register
SFR Address: B1h

Bit	7	6	5	4	3	2	1	0
Name	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

P5 Register
SFR Address: D9h

Bit	7	6	5	4	3	2	1	0
Name	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 I/O Port bit 7~bit 0 Input/Output Data Control

During reading and writing of data to these registers, what actually happens is dependent upon whether the corresponding pin is setup as an output or input.

Register Write Operations

A write operation is only effective when the corresponding pin is setup as an output. In such cases a write operation will setup the logic level on the pin as follows:

0: Output low
1: Output high

Register Read Operations

A read operation will read the current logic level on the corresponding pin.

0: Read low level
1: Read high level

PnM0/PnM1 Registers – Port Mode Registers

These registers only exist for Ports 0~3. They are used to setup the I/O operating mode of each pin. As there are four different operating modes for the Port 0~3 pins, each pin has two bits to select the mode, known as the PnM0 and PnM1 bits. As Ports 4 and 5 only have a single operating mode, they do not have port mode registers.

Port 0 Mode Control

P0M0 Register
SFR Address: 9Eh

Bit	7	6	5	4	3	2	1	0
Name	P0M0.7	P0M0.6	P0M0.5	P0M0.4	P0M0.3	P0M0.2	P0M0.1	P0M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P0M1 Register
SFR Address: 9Fh

Bit	7	6	5	4	3	2	1	0
Name	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Port 1 Mode Control

P1M0 Register
SFR Address: A6h

Bit	7	6	5	4	3	2	1	0
Name	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P1M1 Register
SFR Address: A7h

Bit	7	6	5	4	3	2	1	0
Name	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Port 2 Mode Control

P2M0 Register
SFR Address: AEh

Bit	7	6	5	4	3	2	1	0
Name	P2M0.7	P2M0.6	P2M0.5	P2M0.4	P2M0.3	P2M0.2	P2M0.1	P2M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2M1 Register
SFR Address: AFh

Bit	7	6	5	4	3	2	1	0
Name	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Port 3 Mode Control

P3M0 Register
SFR Address: B6h

Bit	7	6	5	4	3	2	1	0
Name	P3M0.7	P3M0.6	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P3M1 Register
SFR Address: B7h

Bit	7	6	5	4	3	2	1	0
Name	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

These registers operate as pairs, for example P0M0 and P0M1, to select the operating mode for each I/O pin. The following table shows how the PnM0 and PnM1 bits are used to select the I/O operating mode.

PnM0.m	PnM1.m	Configuration of Port n.m
0	0	Quasi-bidirectional
0	1	Push-Pull Output
1	0	Input-Only – High Impedance Input
1	1	Open-Drain Output

Legend: n=0~3 which selects Port 0 to Port 3
m=0~7 which selects the port pin

P0WAKE Register – Port 0 Wake-up

P0WAKE Register
SFR Address: 91h

Bit	7	6	5	4	3	2	1	0
Name	P07WU	P06WU	P05WU	P04WU	P03WU	P02WU	P01WU	P00WU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **P0WAKE:** Port 0 bit 7~bit 0 Wake-up Control
0: Disable
1: Enable

When the device enters the IDLE or Power-Down Mode, the system clock will stop resulting in power being conserved, a feature that is important for battery and other low-power applications. Various methods exist to wake-up the microcontroller, one of which is to change the logic condition on one of the P0.0~P0.7 pins to a low level. Note that the Port 0 wake-up functions are triggered by a low logic level and not by a falling edge. This Port 0 wake-up function is especially suitable for applications that can be woken up via external switches. The P0 wake up pins can be selected individually to have this wake-up feature using the P0WAKE, register.

SRCR Register – Slew Rate Control

SRCR Register
SFR Address: A4h

Bit	7	6	5	4	3	2	1	0
Label	—	—	SRCR.5	SRCR.4	SRCR.3	SRCR.2	SRCR.1	SRCR.0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5 **SRCR.5:** P5 [7:0] Slew Control Enable
0: Fast
1: Slow

Bit 4 **SRCR.4:** P4 [7:0] Slew Control Enable
0: Fast
1: Slow

Bit 3 **SRCR.3:** P3 [7:0] Slew Control Enable
0: Fast
1: Slow

Bit 2 **SRCR.2:** P2 [7:0] Slew Control Enable
0: Fast
1: Slow

Bit 1 **SRCR.1:** P1 [7:0] Slew Control Enable
0: Fast
1: Slow

Bit 0 **SRCR.0:** P0 [7:0] Slew Control Enable
0: Fast
1: Slow

The port pins, when setup as outputs, can be selected to have either a fast or slow slew rate. To minimise noise generation due to fast switching of the output drivers, it may be advisable to select the slower slew rate. The slew rates are selected port wide, individual pins cannot be selected to have either fast or slow slew rates.

I/O Pin Structures

The accompanying diagrams illustrate the internal structures of some generic I/O pin types. As the exact logical construction of the I/O pin will differ from these drawings, they are supplied as a guide only to assist with the functional understanding of the I/O pins.

Quasi-bidirectional I/O – All Ports

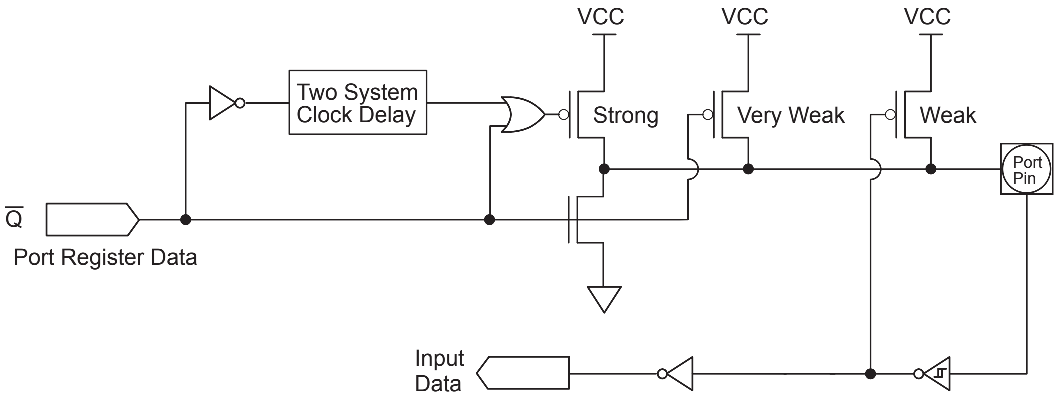
This is the traditional 8051 type I/O port type, constructed from an NMOS FET transistor and three pull high resistors, so called Strong, Weak, Very Weak pull high resistors. This structure can be used to reduce the power consumption and the output switching state respond time.

A Very Weak pull high resistor will be turned on whenever the I/O port registers, associated with the I/O pins, contain a high level.

When the I/O port registers has a high level and the corresponding I/O pins stay at high level as well, the Weak pull high resistor will be turned on. However, if the I/O port registers are high and the corresponding I/O pins are pulled low by the external devices, then the Weak pull high resistor will be disabled by hardware. These weak pull-high resistor enable/disable function are dependant on the voltage level after the I/O pin is connected to the external circuit.

The Strong pull high resistor is used to enhance the output response time. When the output state changes from low to high, the Strong resistor will be turned on after two system clock delay times.

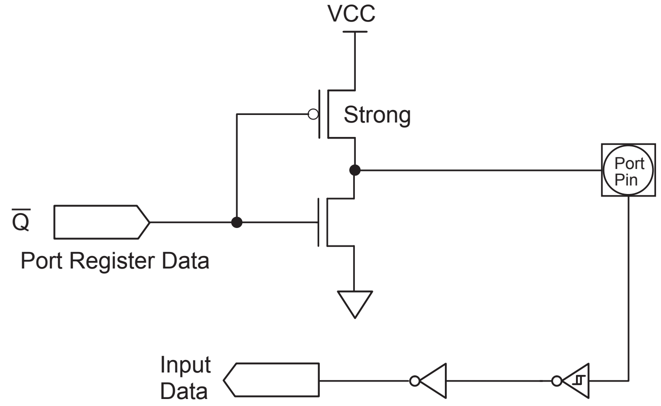
A Quasi-bidirectional pin also provides a Schmitt Trigger input.



Quasi-bidirectional I/O Structure

Push-pull Output – Ports 0~3 Only

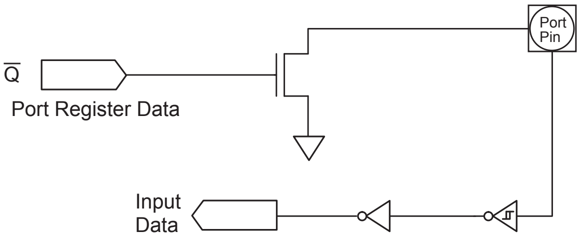
This I/O structure is a standard CMOS type structure with a single NMOS and PMOS complimentary transistor pair. The input is a Schmitt Trigger type input.



Push-pull Output Structure

Open-drain Output – Ports 0~3 Only

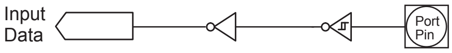
This I/O structure is an open drain type structure with a Schmitt Trigger input. Usually, an external pull high resistor is needed for such applications.



Open-drain Output Structure

Input Only – Ports 0~3 Only

This Input Only structure is a Schmitt Trigger type input without any pull high resistors.



Input Only Structure

Programming Considerations

Within the user program, one of the first things to consider is port initialisation. After a reset, the I/O data register will be set high and I/O port mode registers will be cleared to low. This means that all I/O pins will default to a Quasi-bidirectional structure. The I/O pins can be re-assigned to some other mode for each I/O using the control registers, PnM0 and PnM1. Ports P0~P3 provide four I/O structure modes option while the P4 and P5 only provide a Quasi-bidirectional I/O structure mode. Care should be taken to setup the correct I/O structure for each I/O pin, otherwise unexpected data will be input or output on the I/O pins.

The data registers, P0~P5, reflect the value of the corresponding I/O port, however, they do not necessarily reflect the I/O pin logic state. During reading and writing of data to these registers, what actually happens is dependent upon whether the corresponding pin is setup as an output or input. A write operation is only effective when the corresponding pin is setup as an output. In such cases a write operation will setup the logic level, low or high, on the pin. A read operation will read the current logic level, low or high, on the corresponding pin.

If any pins are setup to be used as A/D input pins then it is important to ensure that the I/O Port Mode registers setup the pins as inputs, which are essentially high impedance inputs. In this way the I/O logic circuits will have a minimal influence on the A/D input impedance.

When using these bit control instructions, a read-modify-write operation takes place. The microcontroller must first read in the data on the entire port, modify it to the required new bit values and then rewrite this data back to the output ports, such as using CLR or SET bit write instructions. Care should be taken that some instructions, the Read-Modify-Write instructions, operate on the Pn register, such as “INC P0” or “ANL P2, A”, while others can operate directly onto the external port input, such as “MOV A, P1”. Note that P4 and P5 cannot be modified by bit manipulation instructions as their registers are not located in bit addressable space. In case of reading, the state of P4 and P5 registers reflects the value of the corresponding I/O port.

The accompanying table illustrates the Read-Modify-Write related instructions.

Mnemonic	Instruction	Example	Bit Manipulation
ANL	Logical AND	ANL P3, A	—
ORL	Logical OR	OR P3, A	—
XRL	Logical XOR	XRL P3, A	—
JBC	Jump if bit set and then clear bit	JBC P3.0, (LABEL)	—
CPL	Complement bit	CPL P3.0	—
INC	Increment	INC P3	—
DEC	Decrement	DEC P3	—
DJNZ	Decrement and jump if not zero	DJNZ P3, (LABEL)	—
MOV Px.y, C	Move carry flag to Bit y of Port x	MOV P3.0, C	V
CLR Px.y	Clear Bit y of Port x	CLR P3.0	V
SET Px.y	Set Bit y of Port x	SET P3.0	V

27 Timer/Event Counters

One of the most fundamental functions in any microcontroller device is the ability to control and measure time. To implement time related functions each device includes several Timer/Counters. The Timers are multi-purpose timing units and serve to provide operations such as Timer/Counter, Input Capture, Compare Match Output and Programmable Clock Output. Each of the Timers has one individual interrupt. The addition of input and output pins for each Timer ensures that users are provided with timing units with a wide and flexible range of features.

Timer/Event Counter Summary

The devices contain four Timers, namely Timer 0, Timer 1, Timer 2 and Timer 3. Each individual Timer is 16-bit wide which are composed of two 8-bit registers, TLn and THn. Timers 0, 1 and 3 have similar structures and similar operating modes. Timer 2 has a different structure and is also known as a Programmable Counter Array, or PCA for short and has functions such as Compare, Reload and Capture functions, so called CRC, as well a programmable clock output function. All timers have a clock divider which provides additional range to the timers.

Various Timer control registers determine how each Timer is operated. The clock sources for the Timers can come from an internal clock source or from an external timer pin. Note that if the external timer input function is selected, the respective pin-shared I/O pins should be configured as input pins.

As Timer 0, 1 and 3 have similar structures they will be described together in their own single chapter, however as Timer 2 has a very different structure it will be described in a separate chapter. The main features and differences among the Timers are summarised in the accompanying table.

Timer Function Summary

Function	Timer 0	Timer 1	Timer 2	Timer 3
13-bit Timer/Counter	√	√	—	√
16-bit Timer/Counter	√	√	—	√
8-bit timer with auto-reload	√	√	—	√
Two 8-bit Timer/Counters	√	—	—	—
16-bit Timer/Counter with auto-reload	—	—	√	—
16-bit Timer/Counter with capture	—	—	√	—
Compare Match Output	—	—	√	—
Programmable Clock Output	—	—	√	—

28 Timer/Event Counters 0, 1, 3

These three timers provide have a similar type and structure and operate with a choice of three modes for Timers 1 and 3 and four modes for Timer 0. They provide basic timing and event counting operations.

Introduction

The different operating modes of the timers are selected using the TnM1 and TnM0 bits in the TMOD or T3CON register.

Timer Mode	TnM1, TnM0 bits	Mode Name	Application Timer
0	00	13-bit Timer-Counter	Timer 0, 1, 3
1	01	16-bit Counter	Timer 0, 1, 3
2	10	8-bit Counter Auto Reload	Timer 0, 1, 3
3	11	Two 8-bit Counters	Timer 0 only

The registers, THn and TLn, are special function registers located in the Special Function Registers and is the place where the actual timer value is stored. This register pair, are each 8-bit wide, and can be cascaded into 13-bit or 16-bit wide using mode options. The value in the timer registers increases by one each time an internal clock pulse is received or an external transition occurs on the external timer pin. The timer will count from the initial value loaded by the preload register to their full count at which point the timer overflows and an internal interrupt signal is generated. If the timer auto-reload mode is selected, the timer value will then be reset with the initial preload register value and continue counting, otherwise the timer value will be reset to zero. Note that to achieve a maximum full range count, the preload register must first be cleared to all zeros.

Timer 0/Timer 1/Timer 3 Register Description

Overall operation of the Timer 0, Timer 1 and Timer 3 are controlled using the registers listed in the accompanying table. A register pair, TLn and THn, exist to store the internal counter 13-bit or 16-bit value. The TCON, IRCON, IEN0, IEN1 registers include the TIMERn interrupt control and interrupt request flags, which are described in the Interrupt section. The remaining registers are control registers which setup the different operating and control modes as well as the clock source control bits.

Timer0/Timer1/Timer3 Register List

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0
IEN1	EXEN2	SWDT	ET3	ECMP	EX6	EX5	EX4	EX3
IRCON	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	—
TMOD	GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
T3CON	GATE3	C/T3	T3M1	T3M0	—	—	TF3	TR3
TLn	D7	D6	D5	D4	D3	D2	D1	D0
THn	D15	D14	D13	D12	D11	D10	D9	D8
TMPRE	T3PRE1	T3PRE0	T2PRE1	T2PRE0	T1PRE1	T1PRE0	T0PRE1	T0PRE0

Note: n=0, 1, 3

TL0 Register
SFR Address: 8Ah

- **16-bit**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TL0:** TIMER0 Counter Low Byte Register bit 7~bit 0

- **13-bit**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	D4	D3	D2	D1	D0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 Unimplemented, read as “0”

Bit 4~0 **TL0:** TIMER0 Counter Low Byte Register bit 4~bit 0

TH0 Register
SFR Address: 8Ch

- **16-bit**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TH0:** TIMER0 Counter High Byte Register bit 15~bit 8

- **13-bit**

Bit	7	6	5	4	3	2	1	0
Name	D12	D11	D10	D9	D8	D7	D6	D5
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TH0:** TIMER0 Counter High Byte Register bit 12~bit 5

TL1 Register
SFR Address: 8Bh

- **16-bit**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TL1:** TIMER1 Counter Low Byte Register bit 7~bit 0

- **13-bit**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	D4	D3	D2	D1	D0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 Unimplemented, read as “0”

Bit 4~0 **TL1:** TIMER1 Counter Low Byte Register bit 4~bit 0

TH1 Register
SFR Address: 8Dh

- **16-bit**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TH1:** TIMER1 Counter High Byte Register bit 15~bit 8

- **13-bit**

Bit	7	6	5	4	3	2	1	0
Name	D12	D11	D10	D9	D8	D7	D6	D5
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TH1:** TIMER1 Counter High Byte Register bit 12~bit 5

TL3 Register
SFR Address: A2h

- **16-bit**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TL3:** TIMER3 Counter Low Byte Register bit 7~bit 0

- **13-bit**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	D4	D3	D2	D1	D0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 Unimplemented, read as “0”

Bit 4~0 **TL3:** TIMER3 Counter Low Byte Register bit 4~bit 0

TH3 Register
SFR Address: A3h

- **16-bit**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TH3:** TIMER3 Counter High Byte Register bit 15~bit 8

- **13-bit**

Bit	7	6	5	4	3	2	1	0
Name	D12	D11	D10	D9	D8	D7	D6	D5
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TH3:** TIMER3 Counter High Byte Register bit 12~bit 5

TMOD Register
SFR Address: 89h

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **GATE1:** Timer 1 Gate Control
0: Disable
1: Enable
This bit is used to enable the Timer 1 Gate function. When the GATE1 bit is set high and Timer 1 is enabled to run using the TR1 bit and when the INT1 pin is input high, then the Timer 1 Counter will increment one on every falling edge on the T1 input pin.

Bit 6 **C/T1:** Timer 1 Counter/Timer selection
0: Timer
1: Counter

Bit 5~4 **T1M1, T1M0:** Timer 1 mode selection
00: Mode 0 – 13-bit Timer/Counter
01: Mode 1 – 16-bit Timer/Counter
10: Mode 2 – 8-bit Auto Reload Timer/Counter
11: Mode 3 – Timer Stopped

Bit 3 **GATE0:** Timer 0 Gate Control
0: Disable
1: Enable
This bit is used to enable the Timer 0 Gate function. When the GATE0 bit is set high and Timer 0 is enabled to run using the TR0 bit and when the INT0 pin is input high, then the Timer 0 Counter will increment one on every falling edge on the T0 input pin.

Bit 2 **C/T0:** Timer 0 Counter/Timer selection
0: Timer
1: Counter

Bit 1~0 **T0M1, T0M0:** Timer 0 mode selection
00: Mode 0 – 13-bit Timer/Counter
01: Mode 1 – 16-bit Timer/Counter
10: Mode 2 – 8-bit Auto Reload Timer/Counter
11: Mode 3 – Two independent 8-bit Timer/Counters

TCON Register
SFR Address: 88h

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **TF1:** Timer 1 interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 6 **TR1:** Timer 1 Run control
0: Stop
1: Run

Bit 5 **TF0:** Timer 0 interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 4 **TR0:** Timer 0 Run control
0: Stop
1: Run

Bit 3 **IE1:** External interrupt 1 request flag
Described elsewhere

Bit 2 **IT1:** External interrupt 1 type control
Described elsewhere

Bit 1 **IE0:** External interrupt 0 request flag
Described elsewhere

Bit 0 **IT0:** External interrupt 0 type control
Described elsewhere

T3CON Register
SFR Address: A1h

Bit	7	6	5	4	3	2	1	0
Name	GATE3	C/T3	T3M1	T3M0	—	—	TF3	TR3
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

Bit 7 **GATE3:** Timer 3 Gate Control
0: Disable
1: Enable
This bit is used to enable the Timer 3 Gate function. When the GATE3 bit is set high and Timer 3 is enabled to run using the TR3 bit and when the INT3 pin is input high, then the Timer 3 Counter will increment one on every falling edge on the T3 input pin.

Bit 6 **C/T3:** Timer 3 Counter/Timer selection
0: Timer
1: Counter

Bit 5~4 **T3M1, T3M0:** Timer 3 mode selection
00: Mode 0 – 13-bit Timer/Counter
01: Mode 1 – 16-bit Timer/Counter
10: Mode 2 – 8-bit Auto Reload Timer/Counter
11: Mode 3 – Timer Stopped

Bit 3~2 Unimplemented, read as "0"

Bit 1 **TF3:** Timer 3 interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

Bit 0 **TR3:** Timer 3 run flag
0: Stop
1: Run

TMPRE Register
SFR Address: 8Fh

Bit	7	6	5	4	3	2	1	0
Name	T3PRE1	T3PRE0	T2PRE1	T2PRE0	T1PRE1	T1PRE0	T0PRE1	T0PRE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T3PRE1, T3PRE0:** Timer 3 Clock Frequency selection
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

Bit 5~4 **T2PRE1, T2PRE0:** Timer 2 Clock Frequency selection
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

Bit 3~2 **T1PRE1, T1PRE0:** Timer 1 Clock Frequency selection
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

Bit 1~0 **T0PRE1, T0PRE0:** Timer 0 Clock Frequency selection
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

Mode 0 – 13-bit Counter/Timer Mode Operation

To select this mode, bits TnM1 and TnM0, should be set to "00". The 13 bits of data are comprised of 5 low bits in the TLn register and 8 high bits in the THn register. The C/Tn bit is used to select the timer or counter function. The Counter/Timer Run or Stop operation is controlled using the TRn bit. If the Counter function is selected, the TRn and GATEn bits can be used to manage the external INTn input to count edge transitions or measure pulse widths. The timer/counter clock source is decided by the TnPRE0 and TnPRE1 bits in the TMPRE register. Note that the TRn bit is used to control the Timer/Counter run or stop function. Clearing this bit will not clear the TLn and THn registers, the registers should be initialised by the application program. When an overflow occurs, the TFn interrupt request flag will be set and an interrupt will take place if the interrupt is enabled.

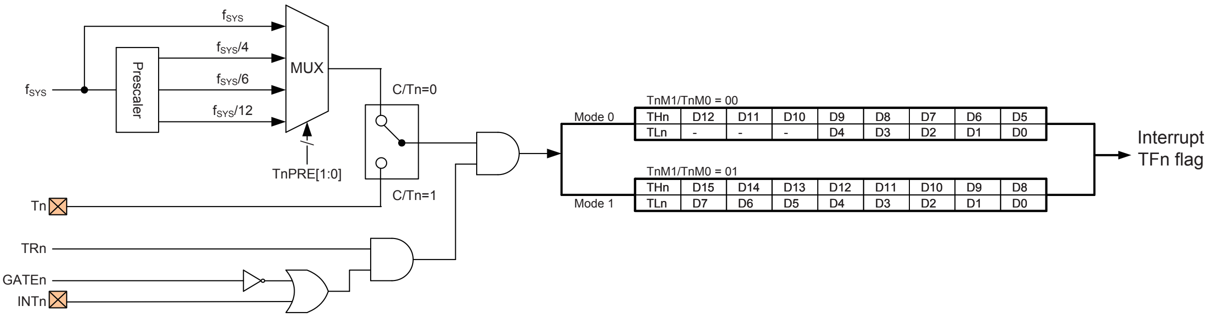
13-bit Counter Data

Register	Bit							
	7	6	5	4	3	2	1	0
THn	D12	D11	D10	D9	D8	D7	D6	D5
TLn	—	—	—	D4	D3	D2	D1	D0

Note: n=1, 2, 3

Mode 1 – 16-bit Counter/Timer Mode Operation

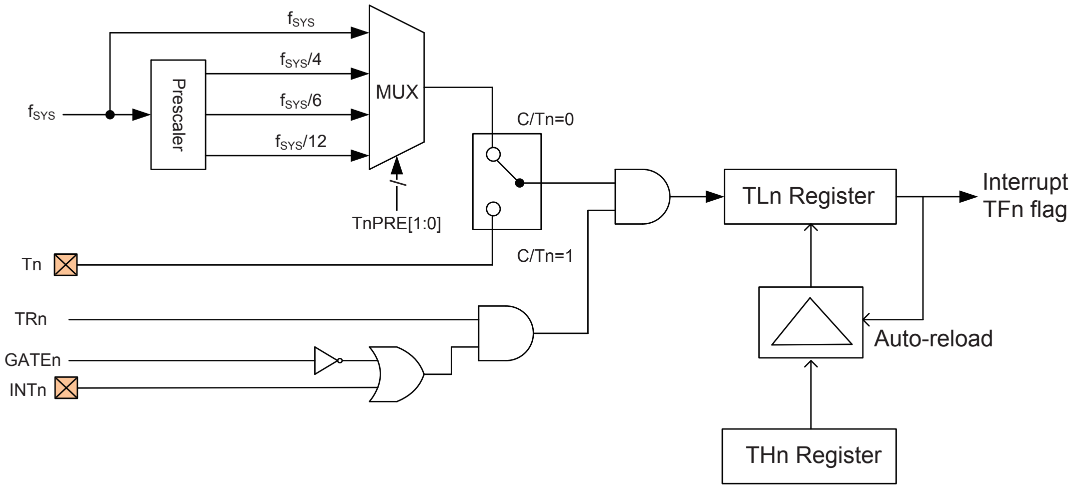
To select this mode, bits TnM1 and TnM0, should be set to "01" respectively. The 16 bits of data are stored in the TLn and THn registers. The C/Tn bit is used to select the timer or counter function. The Counter/Timer Run or Stop is controlled by TRn bit. If the Counter function is selected, the TRn and GATEn bits can be used to manage the external INTn input to count edge transitions or measure pulse widths. The timer/counter clock source is decided by the TnPRE0 and TnPRE1 bits in the TMPRE register. Note that the TRn bit is used to control the Timer/Counter run or stop function. Clearing this bit will not clear the TLn and THn registers, the registers should be initialised by the application program. When an overflow occurs, the TFn interrupt request flags will be set and an interrupt will take place if the interrupt is enabled. The following block illustrates the 13-bit and 16-bit Timer/Counter basic operational blocks.



Mode 0 and Mode 1 Block Diagram – Timer 0, 1, 3

Mode 2 – 8-bit Auto-reload Counter/Timer Mode Operation

To select this mode, bits TnM1 and TnM0, should be set to "10" respectively. This function is implemented by the 8-bit TLn and THn registers. The C/Tn bit is used to select the timer or counter function. The Counter/Timer Run or Stop is controlled by the TRn bit. If the Counter function is selected, the TRn and GATEn bits can be used to manage the external INTn input to count edge transitions or measure pulse widths. The timer/counter clock source is decided by the TnPRE0 and TnPRE1 registers in the TMPRE register. When the values in the TLn register overflows, the TLn value will be auto-reloaded with the data in the THn register and an interrupt will take place if the interrupt is enabled. Note that the value of THn register should be initialised by the application program. The accompanying block diagram illustrates the 8-bit Auto-Reload Timer/Counter basic operational blocks.

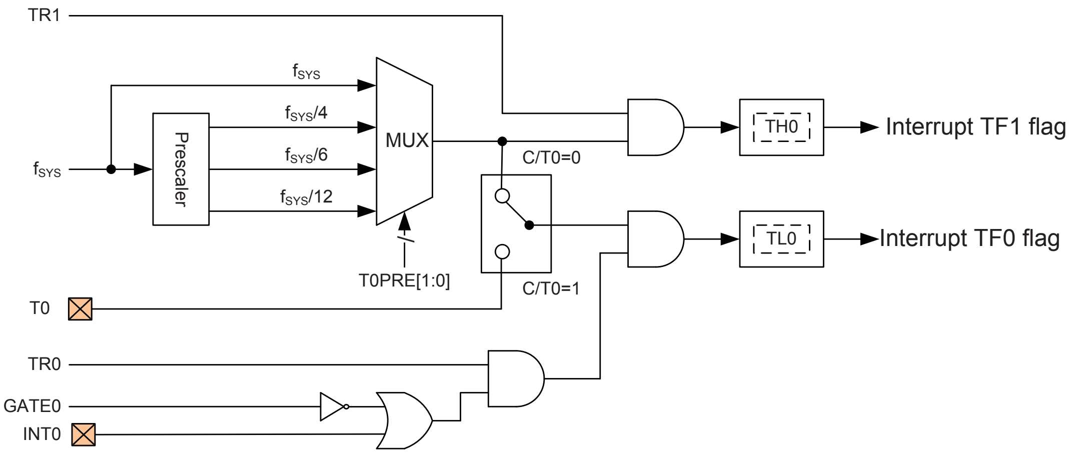


Mode 2 Block Diagram – Timer 0, 1, 3

Mode 3 – Two 8-Bit Timers/Counters Mode Operation – Timer 0 Only

To select this mode, bits T0M1 and T0M0, should be set to "11" respectively. This mode is only available for Timer 0. For Timer 1 and Timer 3, this mode is not available and if selected will stop the timer function. The two 8-bit Timer/Counter function is implemented by the two individual 8-bit TL0 and TH0 registers. TL0 can have both Timer and Counter functions while TH0 can only have a Timer function. The C/T0 bit is used to select the timer or counter function for TL0. The TL0 Run or Stop is controlled by the TR0 bit. If the Counter function is selected, the TR0 and GATE0 bits can be used to manage the external INT0 input to count external edge transitions or to measure input pulse widths. If the TL0 counter overflows, an interrupt will be generated and the interrupt request flag, TF0, will be set high. The timer/counter clock source is decided by the T0PRE0 and T0PRE1 bits in the TMPRE register.

In addition to TL0, the other 8-bit timer, TH0, can use the TR1 bit to enable the Timer. If the TH0 counter overflows, an interrupt will be generated and the interrupt request flag, TF1, will be set high. The timer clock source is decided by the T0PRE0 and T0PRE1 bits in the TMPRE register. The following block illustrates the two 8-bit Timer/Counters basic operational blocks.



Mode 3 Block Diagram – Timer 0

29 Timer 2 with Additional 4-channel PCA

The structure of Timer 2 is very different from that of Timers 0, 1 and 3 and is therefore described in its own chapter.

Introduction

The Timer 2 provides the Timer, Event Counter, Gated timer functions and also cooperates with a 4-channel Programmable Counter Array, known as PCA, to implement the Compare, Reload, Capture and Programmable Clock Output functions. Each channel has a module, so there are four modules, named Module 0~Module 3. Each module can be operated as a Compare and Capture function while Module 0 can also be operated as a Compare, Reload, Capture, known as CRC, and Programmable Clock Output functions. The accompanying tables and diagram illustrate the PCA modules functional compare table, timer I/O pin list and basic operational block diagram.

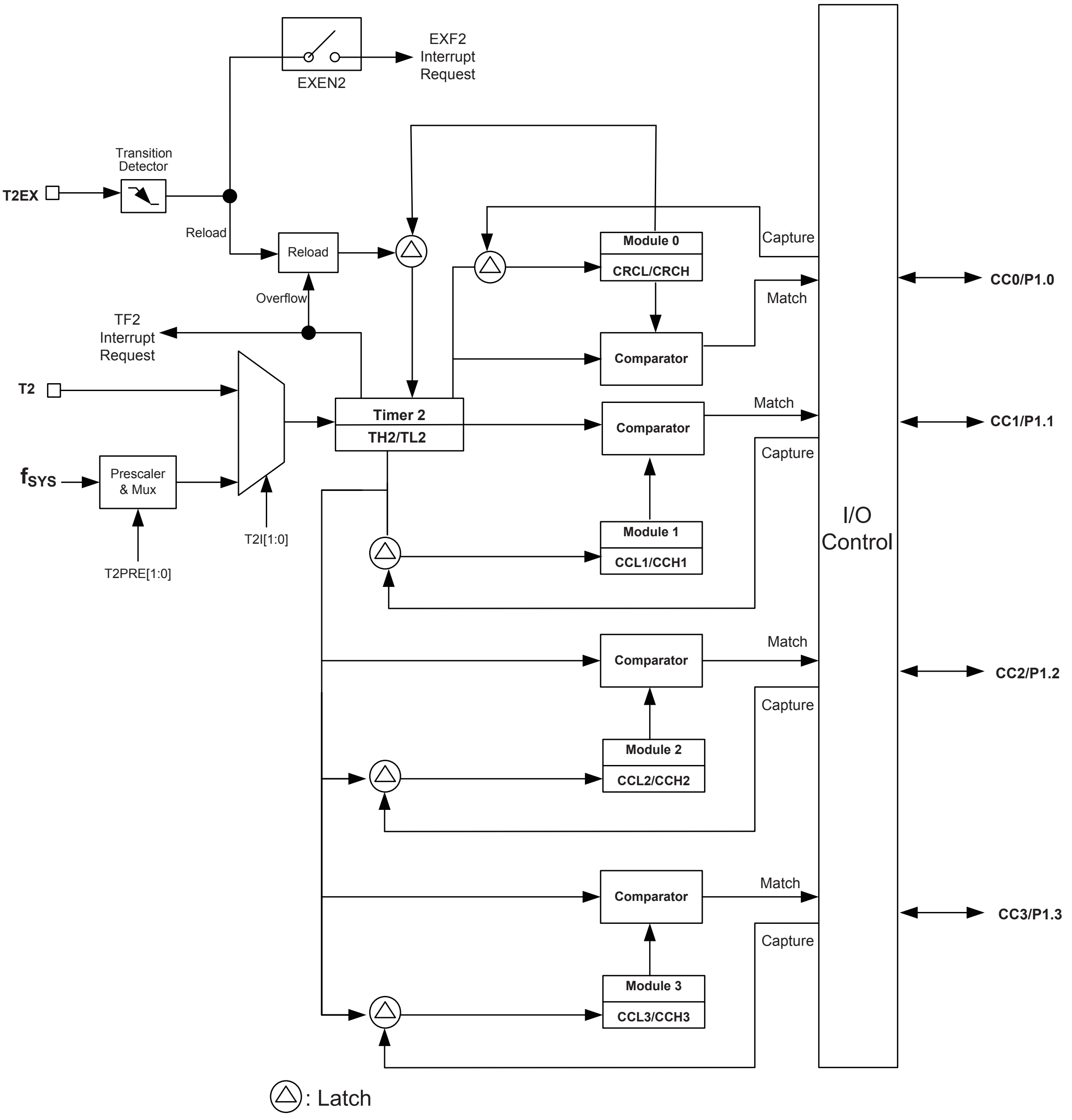
Timer 2 with PCA Modules Operating Modes Summary

Module	Compare	Capture	Reload	Clock Output
0	√	√	√	√
1	√	√	—	—
2	√	√	—	—
3	√	√	—	—

Note: Module 0 only provides the reload value from the Timer 2 capture registers CRCH and CRCL for the Clock Output Mode. It is important to note that the actual Clock Output pin is T2 and not CC0.

Timer 2 with PCA Modules I/O Pins

Function	Input Pins	Output Pins
Compare	—	CC0, CC1, CC2, CC3
Reload trigger	T2EX	—
Capture	CC0, CC1, CC2, CC3	—
Event Counter or Gated input	T2	—
Clock Output	—	T2



Timer 2 with PCA Modules Block Diagram

Timer 2

Timer 2 is a 16-bit wide count-up counter which is driven by a user selectable internal or external clock source. The counter is composed of two registers, TL2 and TH2, to implement the Timer, event counter and gated timer functions. The clock source is decided by the bits T2I1 and T2I0 in the T2CON register.

Timer function

To select this function, bits T2I1 and T2I0 in the T2CON register, should be set to "01" respectively. The value in the Timer 2 registers, TL2 and TH2, increases by one each time an internal clock pulse is received. The count rate is derived from the "f_{SYS}". The prescaler can be managed by the T2PRE1 and T2PRE0 bits in the TMPRE register. When the timer counter is overflowed, an interrupt will take place and the interrupt request flag, TF2, will be set to high.

Event Counter function

To select this function, bits T2I1 and T2I0 in the T2CON register, should be set to "10" respectively. The value in the Timer 2 registers, TL2 and TH2, increases by one each time a falling edge occurs on the external timer pin, T2. When the timer counter is overflowed, an interrupt will take place and the interrupt request flag, TF2, will be set to high. The maximum count rate is 1/4 of the system clock frequency.

Gated Timer function

To select this function, bits T2I1 and T2I0 in the T2CON register, should be set to "11" respectively. The value in the Timer 2 registers, TL2 and TH2, increases by one each time an internal clock pulse is received. The count rate is derived from the "f_{SYS}" and the prescaler can be managed by the T2PRE1 and T2PRE0 bits in the TMPRE register. The external timer pin, T2, can be a gate to the Timer 2 input. When the T2 pin is set high, the Timer 2 keeps counting and when the Timer 2 is cleared to low, the Timer 2 will be stopped. The T2 input signal will be sampled once by every internal system clock. When the timer counter is overflowed, an interrupt will take place and the interrupt request flag, TF2, will be set to high.

Timer 2 with PCA

Time 2 and 4-channel PCA modules provide the Compare, Reload, Capture and programmable clock output functions. Each of the four Timer 2 Modules contains a pair of registers, CRCL/CRCH for Module 0 and CCLn/CCHn for Modules 1, 2 and 3. These registers are compared with the Timer 2 TL2/TH2 register pair and when a compare match occurs, an interrupt signal can be generated. The value in the Timer 2 registers increases by one each time an internal clock pulse is received or an external transition occurs on the external timer pin.

The Compare function provides two modes, Mode 0 and Mode 1. When a compare match takes place, the compare results will output to the respective output pins, according to the selected mode. Refer Compare mode section for details.

There are two modes for the Capture function, Mode 0 and Mode 1, which are used to select different trigger methods. In Mode 0, the Capture function is triggered by the external I/O pins, CCn. In Mode 1, the Capture function is triggered by writing data to the CCLn or CRCL registers. Once the Capture function is enabled and triggered, the Timer 2 data in the TL2 and TH2 registers will be captured into the respective CCLn/CCHn or CRCL/CRCH registers. Refer Capture modes for details.

In the Reload mode, the timer counter registers, TH2 and TL2, are located in the Special Function Registers and is the place where the actual timer value is stored. The value in the timer registers increases by one each time an internal clock pulse is received or an external transition occurs on the external timer pin. The timer will count from the initial value loaded by the preload register to the full count of FFFFH for the 16-bit Timer/Event Counters, at which point the timer overflows and an internal interrupt signal is generated. There are two modes to reload the CRC register data, one is the counter overflow and the other is triggered by the falling edge on the T2EX pin. Refer Reload mode for details.

In the Programmable Clock Output mode, the clock output frequency depends on the system clock and the reload value of the Timer 2 capture registers, CRCH and CRCL. The output clock is generated by programming the T2CON control bit, and output via T2 pin. Refer Programmable Clock Output mode for details.

Timer 2 Register Description

The Timer 2 value is stored in a register pair, TL2/TH2. Each of the internal PCA modules has a register pair, known as CRCL/CRCH for Module 0 and CCLn/CCHn for modules 1, 2 and 3. The T2CON register is related to the interrupt control register which is described in the Interrupt section. The remaining two registers, CCEN and T2CON1, are control registers which setup the different operating and control modes. The following table provides a register summary list for Timer 2.

Timer 2 Register List

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL2	D7	D6	D5	D4	D3	D2	D1	D0
TH2	D15	D14	D13	D12	D11	D10	D9	D8
CRCL	D7	D6	D5	D4	D3	D2	D1	D0
CRCH	D15	D14	D13	D12	D11	D10	D9	D8
CCLn	D7	D6	D5	D4	D3	D2	D1	D0
CCHn	D15	D14	D13	D12	D11	D10	D9	D8
T2CON	—	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0
T2CON1	—	—	—	—	T2OI	T2OE	—	—
CCEN	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0

CCEN Register
SFR Address: C1h

Bit	7	6	5	4	3	2	1	0
Name	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **COCAH3, COCAL3:** Compare/Capture mode select for Module 3 CC3 register
00: Disable
01: Capture on rising edge at the CC3 pin
10: Compare mode
11: Capture on write data into register CCL3

Bit 5~4 **COCAH2, COCAL2:** Compare/Capture mode select for Module 2 CC2 register
00: Disable
01: Capture on rising edge at the CC2 pin
10: Compare mode
11: Capture on write data into register CCL2

Bit 3~2 **COCAH1, COCAL1:** Compare/Capture mode select for Module 1 CC1 register
00: Disable
01: Capture on rising edge at the CC1 pin
10: Compare mode
11: Capture on write data into register CCL1

Bit 1~0 **COCAH0, COCAL0:** Compare/Capture mode select for Module 0 CRC register
00: Disable
01: Capture on rising or falling edge at the CC0 pin
10: Compare mode
11: Capture on write data into register CRCL

T2CON Register
SFR Address: C8h

Bit	7	6	5	4	3	2	1	0
Name	—	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 Unimplemented, read as "0"

Bit 6 **I3FR:** Active edge selection for external interrupt "INT3" and PCA module 0 Compare and Capture functions
0: Falling edge
1: Rising edge
This bit is used to select the external interrupt triggered edge for INT3, the PCA Module 0 Compare mode output interrupt triggered edge and the PCA Module 0 Capture mode input triggered edge. Once the compare mode is enabled, the PCA interrupt will replace the external interrupt. When Timer 2 is selected as compare mode 0, the I3FR bit is recommended to be set high by firmware.

Bit 5 **I2FR:** Active edge selection for external interrupt "INT2"
Described elsewhere

Bit 4~3 **T2R1, T2R0:** Timer 2 reload mode selection
00: Reload function disabled
01: Reload function disabled
10: Mode 0
11: Mode 1

Bit 2 **T2CM:** Timer 2 Compare mode selection
0: Mode 0
1: Mode 1

Bit 1~0 **T2I1, T2I0:** Timer 2 clock source select
00: Timer 2 stopped
01: Internal clock source, decided by the T2PRE1 and T2PRE0 bits in the TMPRE register
10: External T2 pin falling edge clock source
11: Internal clock source, decided by the T2PRE1 and T2PRE0 bits, gated by the external T2 pin

T2CON1 Register
SFR Address: FEh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	T2OI	T2OE	—	—
R/W	—	—	—	—	R/W	R/W	—	—
POR	—	—	—	—	1	0	—	—

Bit 7~4 Unimplemented, read as “0”

Bit 3 **T2OI:** Timer 2 output initial state control
0: T2 pin initial output Low
1: T2 pin initial output High

The Timer 2 output initial state can be selected by the T2OI bit before enable the Timer 2 programmable clock output function.

Bit 2 **T2OE:** Timer 2 output enable bit
0: Disable
1: Enable

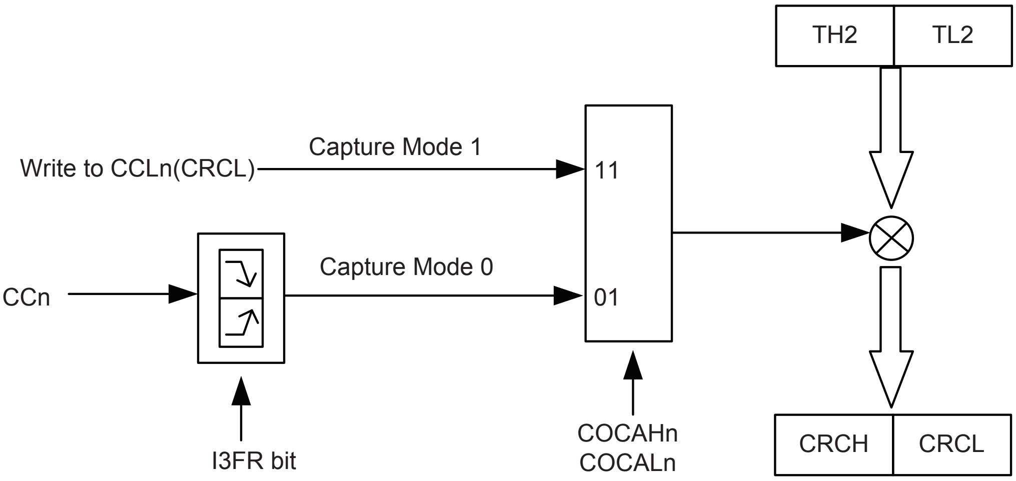
The Timer 2 output is enabled by setting the T2OE bit high.

When the Timer 2 output is disabled, this pin can be used as the other pin shared functions.

Bit 1~0 Unimplemented, read as “0”

Capture Modes

Timer 2 has two capture modes, the Capture on Edge Mode, known as Capture Mode 0, and the Capture on Write Mode, known as Capture Mode 1. The required mode is selected using the COCAHn and COCALn bits in the CCEN register. The accompanying diagram illustrates the basic operational blocks.



Note: 1. n=0~3
2. CC1~CC3 capture input by rising edge
3. CC0 capture input by rising or falling edge selected by the I3FR bit
4. Write to CCLn is for CC1~CC3 and Write to CRCL is for CC0

Capture Modes Block Diagram

Capture On Edge Mode

To select this mode, bits COCAHn and COCALn in the CCEN register, should be set to “01” respectively. In this mode, Modules 1~3 will capture the Timer 2 counter on the rising edge of an external signal applied on the CC1~CC3 pins. Module 0 will capture the Timer 2 counter contents on a rising or falling edge applied on the CC0 pin. The rising or falling edge trigger is controlled by the I3FR bit in the T2CON register.

Capture On Write Mode

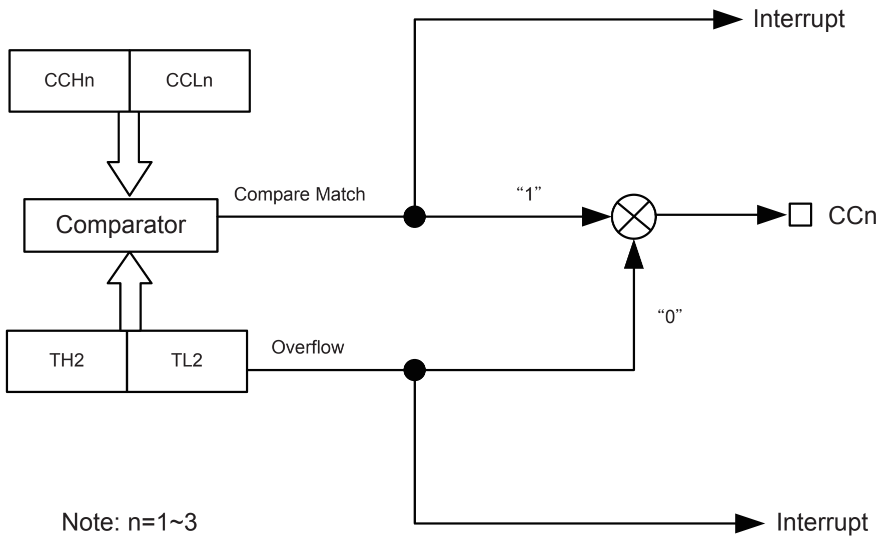
To select this mode, bits COCAHn and COCALn in the CCEN register, should be set to “11” respectively. In this mode a Timer 2 Capture is generated by any write operation into the capture register low byte. The value written to capture register is irrelevant for this function. The Timer 2 contents will be latched into the appropriate capture registers.

Compare Modes

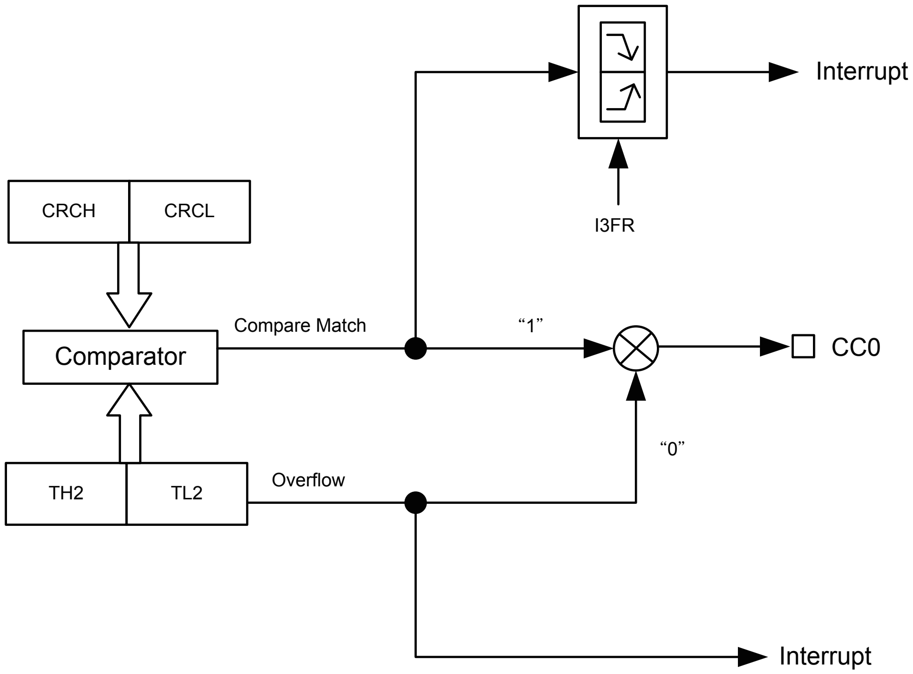
Timer 2 has two compare modes, known as Mode 0 and Mode 1. The required mode is selected using the T2CM bit in the T2CON register. Setting counter data in the Compare modes can implement the PWM function for various control applications.

Compare Mode 0

In Mode 0, if the Timer 2 counter data is the same as the Compare registers, the compare output will be set from low to high and the Timer 2 counter overflow will clear the respective output pins, CCn, to low. In addition, the Module 0 can select the output rising or falling edge interrupt trigger by the I3FR bit in the T2CON register. The accompanying diagrams illustrate the Basic application blocks.

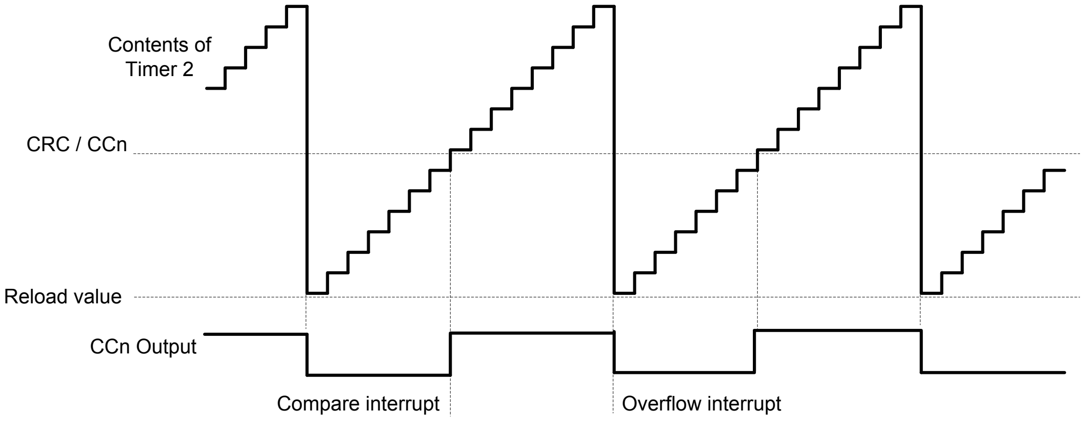


Compare Mode 0 – Module 1, Module 2, Module 3



Compare Mode 0 – Module 0

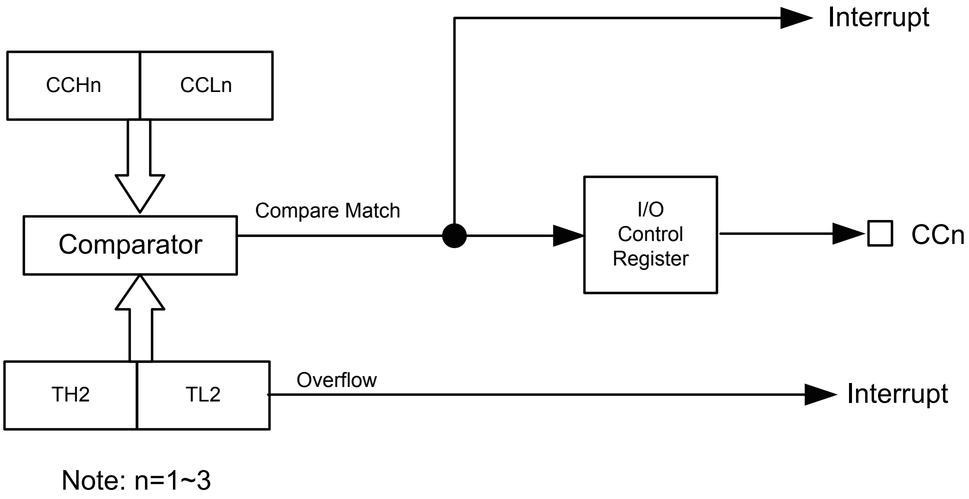
Figure below illustrates the operation of compare mode 0.



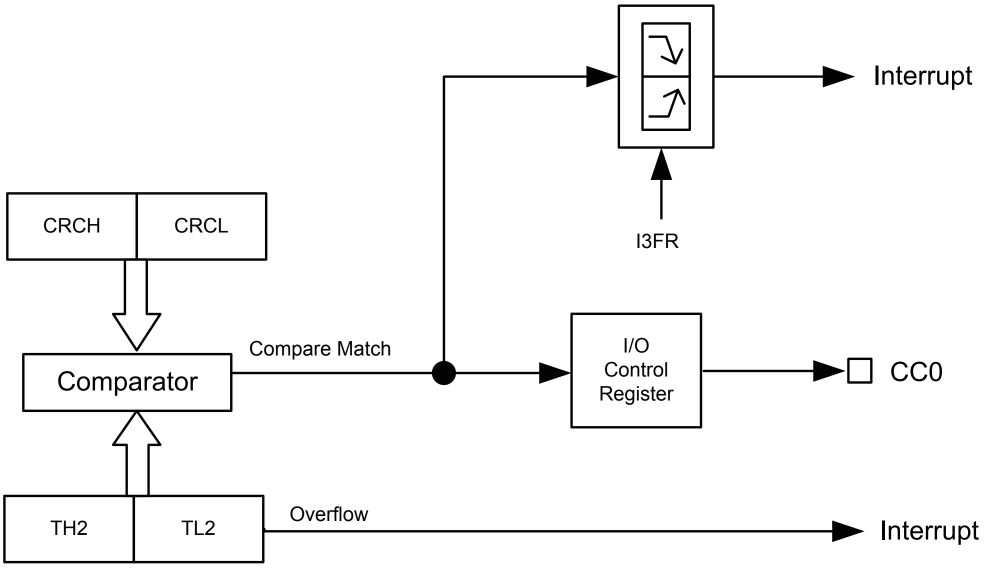
Compare Mode 0 Timing Diagram

Compare Mode 1

In Mode 1, the compare output can be decided by the software setting of the I/O pins control register, P1. When the compare match takes place, the control register value will be outputted to I/O pins, CCn, and the Timer 2 counter overflow will not affect the Compare output. In addition, the Module 0 can select the output rising or falling edge interrupt trigger by the I3FR bit in the T2CON register. The accompanying diagrams illustrate the Basic application blocks.

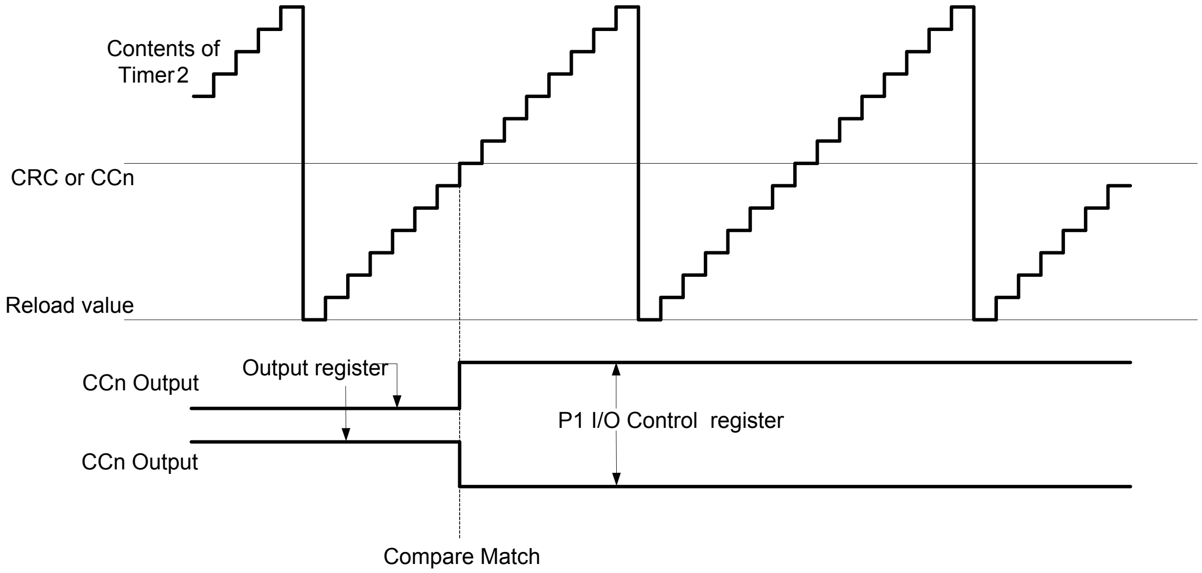


Compare Mode 1 – Module1, Module2, Module 3



Compare Mode 1 – Module 0

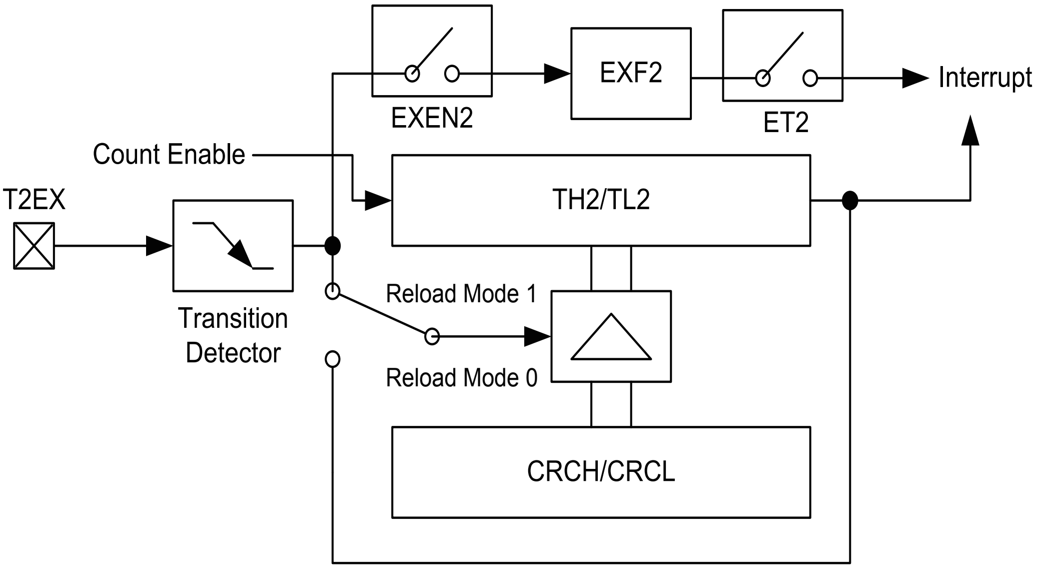
Figure below illustrates the operation of compare mode 1.



Compare Mode 1 Timing Diagram

Reload Mode

Module 0 provides a Reload Mode function. In the reload function, preset values in the CRCL and CRCH registers are loaded into the TL2 and TH2 registers. There are two kinds of Reload modes, Mode 0 and Mode 1, which are selected by the T2R1 and T2R0 bits in the T2CON register. In Reload Mode 0, the Reload enable is controlled by the Timer 2 overflow which is an auto reload and a Timer 2 interrupt will take place. In Reload Mode 1, a falling edge at the T2EX input pin will reload the data from CRCH/CRCL registers to TH2/TL2 registers. When the external reload interrupt control bit, EXEN2, and the Timer 2 interrupt control bit, ET2, are both set high, a Timer 2 interrupt will be generated .The following diagram illustrates the basic operation.



Reload Mode – Module 0

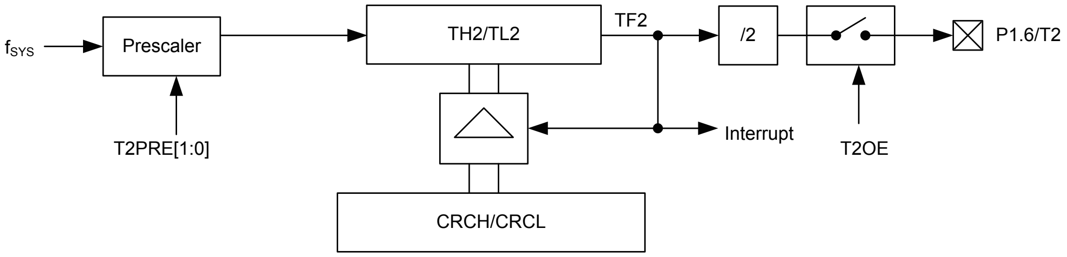
Programmable Clock Output Mode

The Programmable Clock Output mode is related to Module 0. With this function, Timer 2 can generate various clock outputs. This function is enabled by the T2OE bit in the T2CON1 register. The output initial state is decided by the T2OI bit in the T2CON1 register. The Timer 2 enable control or clock source is selected by the T2I1 and T2I0 bits in the T2CON register. The clock source is further decided by the T2PRE1 and T2PRE0 bits in the TMPRE register. The data in the TL2 and TH2 registers decides the clock duty cycle. If the counter overflows, then the CRCL and CRCH registers will be auto-reloaded to the TL2 and TH2 registers. There are two ways to implement a 50% duty cycle clock output on the T2 pin. One method is to input the external clock for Timer/Counter 2 and the other is to output a 50% duty cycle clock ranging from 61HZ to 4MHz when the system clock is selected as 16MHz. To configure the Timer/Counter 2 as a clock generator, the T2I1 and T2I0 bits in the T2CON register must be set as 0 and 1 respectively to start the timer and the T2OE bit in the T2CON1 register must be set as well.

The clock-out frequency depends on the system frequency and the reload value of Timer 2 capture registers (CRCH, CRCL) as shown in this equation:

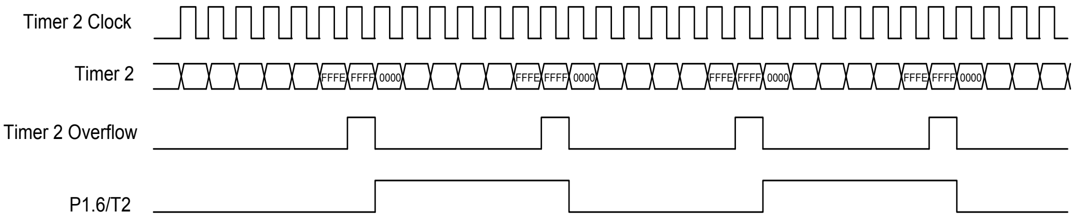
$$\textit{Clock Out Frequency} = \frac{\textit{Timer 2 Clock Frequency}}{2 * (65536 - [CRCH, CRCL])}$$

The Timer 2 Clock frequency is dependent on the T2PRE0 and T2PRE1 bits. The accompanying diagram illustrates the Timer2 Clock output basic operation block diagram.



Timer2 Clock Output Block Diagram

If the Timer 2 Programmable Clock Output Mode is selected, it is essential for the Port 1 control registers, P1M1 and P1M0, to setup the P1.6 pin as an output. The accompanying diagram illustrates the Timer2 programmable clock output timing diagram.



Programmable Clock Output Timing Diagram – Module 0

30 Analog to Digital Converter – ADC

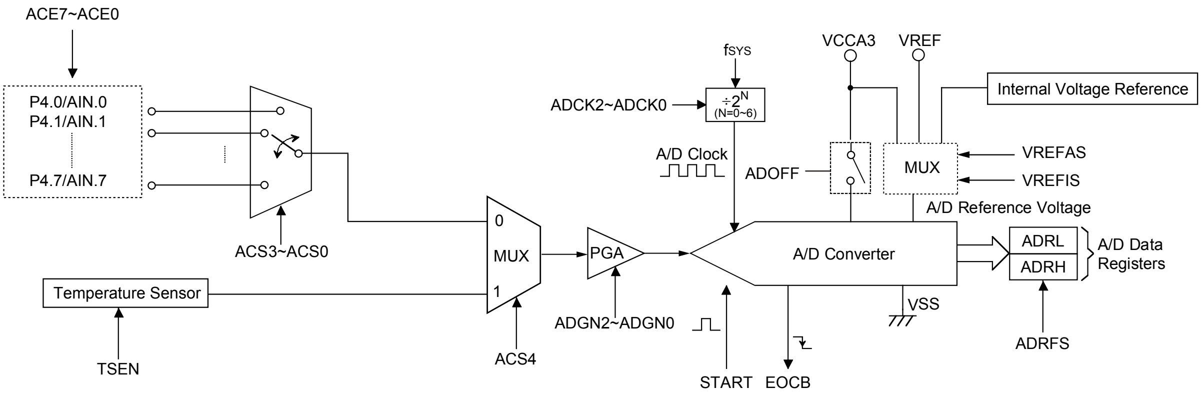
The devices include a multi-channel 12-bit fully integrated Analog to Digital Converter or ADC. A range of programmable features allow flexible and fast analog to digital conversion for a wide range of input signals.

A/D Overview

The Analog to Digital Converter contains a range of features which include:

- Multiplexed Multi-channel Inputs
- Programmable Gain Amplifier
- Temperature Sensor Input
- Internal Voltage Reference Source
- External Reference Voltage Input
- Programmable Clock Speed
- A/D Converter Interrupt

All functions are controlled using dedicated ADC control registers for setup and dynamic control. The following block diagram shows the overall structure of the converter together with its relative control bits.



A/D Converter Structure

Analog to Digital Converter – ADC

A/D Converter Register Description

A read only register pair exists to store the ADC data 12-bit value. The remaining registers are control registers which setup the operating and control function of the A/D converter.

A/D Converter Register List

Name	Bit							
	7	6	5	4	3	2	1	0
ADRL(ADRFS=0)	D3	D2	D1	D0	—	—	—	—
ADRL(ADRFS=1)	D7	D6	D5	D4	D3	D2	D1	D0
ADRH(ADRFS=0)	D11	D10	D9	D8	D7	D6	D5	D4
ADRH(ADRFS=1)	—	—	—	—	D11	D10	D9	D8
ADCR0	START	EOCB	ADOFF	ADRFS	ACS3	ACS2	ACS1	ACS0
ADCR1	ACS4	TSEN	—	VREFAS	VREFIS	ADCK2	ADCK1	ADCK0
ADCR2	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0
ADPGA	—	—	—	—	—	ADGN2	ADGN1	ADGN0

A/D Converter Data Registers – ADRL, ADRH

As the devices contain an internal 12-bit A/D converter, they require two data registers to store the converted value. These are a high byte register, known as ADRH, and a low byte register, known as ADRL. After the conversion process takes place, these registers can be directly read by the microcontroller to obtain the digitised conversion value. As only 12 bits of the 16-bit register space is utilised, the format in which the data is stored is controlled by the ADRFS bit in the ADCR0 register as shown in the accompanying table. D0~D11 are the A/D conversion result data bits. Any unused bits will be read as zero.

A/D Data Registers

ADRFS	ADRH								ADRL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
1	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

A/D Converter Control Registers – ADCR0, ADCR1, ADCR2, ADPGA

To control the function and operation of the A/D converter, four control registers known as ADCR0, ADCR1, ADCR2 and ADPGA are provided. These 8-bit registers define functions such as analog channel selection, converted data format, PGA gain, clock source as well as the start bit and end of conversion flag. As the device contains only one actual analog to digital converter hardware circuit, each of the individual 8 analog inputs must be routed to the converter. It is the function of the ACS4~ACS0 bits to determine which analog channel input pin, reference voltage or internal temperature sensor is actually connected to the internal A/D converter.

The ADCR2 control register bits determine whether the pins on Port 4 are to be used as A/D converter analog inputs or used as logic I/O pins. Setting the corresponding bit high will select the A/D input function, clearing the bit to zero will select the I/O function. When the pin is selected to be an A/D input, its logic I/O function will be removed and any internal pull-high resistors connected to these pins will be automatically removed.

The ADPGA register determines the gain of the Programmable Gain Amplifier which is used to amplify the analog input signal before conversion by the A/D Converter.

ADCR0 Register
SFR Address: F1h

Bit	7	6	5	4	3	2	1	0
Name	START	EOCB	ADOFF	ADRFS	ACS3	ACS2	ACS1	ACS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	1	0	0	0	0	0

Bit 7 **START:** Starts the A/D conversion
0→1→0: Start
0→1: Reset the A/D converter and set EOCB to "1"
This bit is used to initiate an A/D conversion process. The bit is normally low but if set high and then cleared low again, the A/D converter will initiate a conversion process. When the bit is set high the A/D converter will be reset.

Bit 6 **EOCB:** End of A/D conversion flag
0: A/D conversion ended
1: A/D conversion in progress
This read only flag is used to indicate when an A/D conversion process has completed. When the conversion process is running the bit will be high.

Bit 5 **ADOFF** : ADC power on/off control bit
0: ADC power on
1: ADC power off
This bit controls the power to the A/D internal function. This bit should be cleared to zero to enable the A/D converter. If the bit is set high then the A/D converter will be switched off reducing the device power consumption. As the A/D converter will consume a limited amount of power, even when not executing a conversion, this may be an important consideration in power sensitive battery powered applications.

Bit 4 **ADRFS:** ADC Data Format Control
0: ADC Data MSB is ADRH bit 7, LSB is ADRL bit 4
1: ADC Data MSB is ADRH bit 3, LSB is ADRL bit 0
This bit controls the format of the 12-bit converted A/D value in the two A/D data registers.

Bit 3~0 **ACS3~ACS0:** Select A/D channel (when ACS4 is "0")
0000: AIN.0
0001: AIN.1
0010: AIN.2
0011: AIN.3
0100: AIN.4
0101: AIN.5
0110: AIN.6
0111: AIN.7
1xxx: Undefined, must not be used
These are the A/D channel select control bits. As there is only one internal hardware A/D converter each of the eight A/D inputs must be routed to the internal converter using these bits. If bit ACS4 in the ADCR1 register is set high then the internal temperature sensor will be routed to the A/D Converter and these ADC input channels disconnected.

ADCR1 Register
SFR Address: F2h

Bit	7	6	5	4	3	2	1	0
Name	ACS4	TSEN	—	VREFAS	VREFIS	ADCK2	ADCK1	ADCK0
R/W	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W
POR	0	0	—	0	0	0	0	0

Bit 7 **ACS4:** Internal temperature sensor ADC input control
0: Disable
1: Enable

This bit enables the temperature sensor to the A/D converter. The TSEN bit must first have been set to enable the temperature sensor circuit. When the ACS4 bit is set high, the temperature sensor will be routed to the A/D converter and the other A/D input channels disconnected.

Bit 6 **TSEN:** Internal temperature sensor control
0: Disable
1: Enable

This bit controls the internal temperature sensor function to the A/D converter. When the bit is set high the temperature sensor can be used by the A/D converter.

Bit 5 Unimplemented, read as "0"

Bit 4 **VREFAS:** ADC reference voltage select
0: VCCA3 pin
1: Externally supplied on VREF pin or internal voltage reference generator

This bit is used to select the reference voltage for the A/D converter. If the bit is high then the A/D converter reference voltage is supplied on the external VREF pin or the internal reference voltage, the choice being made using the VREFIS bit. If the pin is low then the internal reference is used which is sourced from the power supply pin VCCA3.

Bit 3 **VREFIS:** A/D and DAC reference voltage select
0: Externally supplied on VREF pin
1: Internal Voltage Reference

Bit 2~0 **ADCK2~ADCK0:** Select ADC clock source
000: f_{SYS}
001: $f_{SYS}/2$
010: $f_{SYS}/4$
011: $f_{SYS}/8$
100: $f_{SYS}/16$
101: $f_{SYS}/32$
110: $f_{SYS}/64$
111: f_{SYS}

ADCR2 Register
SFR Address: F3h

Bit	7	6	5	4	3	2	1	0
Name	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **ACE7:** P4.7 A/D input select
0: Logic I/O
1: A/D input, AIN.7

Bit 6 **ACE6:** P4.6 A/D input select
0: Logic I/O
1: A/D input, AIN.6

Bit 5 **ACE5:** P4.5 A/D input select
0: Logic I/O
1: A/D input, AIN.5

Bit 4 **ACE4:** P4.4 A/D input select
0: Logic I/O
1: A/D input, AIN.4

Bit 3 **ACE3:** P4.3 A/D input select
0: Logic I/O
1: A/D input, AIN.3

Bit 2 **ACE2:** P4.2 A/D input select
0: Logic I/O
1: A/D input, AIN.2

Bit 1 **ACE1:** P4.1 A/D input select
0: Logic I/O
1: A/D input, AIN.1

Bit 0 **ACE0:** P4.0 A/D input select
0: Logic I/O
1: A/D input, AIN.0

ADPGA Register
SFR Address: F4h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	ADGN2	ADGN1	ADGN0
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 Unimplemented, read as “0”

Bit 2~0 **ADGN2~ADGN0:** PGA gain select
000: PGA off
001: 0.5
010: 1
011: 2
100: 4
101: 8
110: 12
111: 16
These three bits are used to select the PGA internal gain setting to allow greater A/D Converter input voltage dynamic range.

A/D Operation

Controlling the power on/off function of the A/D converter circuitry is implemented using the ADOFF bit in the ADCR0 register. This bit must be zero to power on the A/D converter. When the ADOFF bit is cleared to zero to power on the A/D converter internal circuitry a certain delay, as indicated in the timing diagram, must be allowed before an A/D conversion is initiated. Even if no pins are selected for use as A/D inputs by clearing the ACE7~ACE0 bits in the ADCR2 register, if the ADOFF bit is zero then some power will still be consumed. In power conscious applications it is therefore recommended that the ADOFF is set high to reduce power consumption when the A/D converter function is not being used.

The reference voltage supply to the A/D Converter can be supplied from either the positive power supply pin, VCCA3, internal voltage reference or from an external reference sources supplied on pin VREF. The desired selection is made using the VREFAS and VREFIS bits.

The START bit in the ADCR0 register is used to start and reset the A/D converter. When the microcontroller sets this bit from low to high and then low again, an analog to digital conversion cycle will be initiated. When the START bit is brought from low to high but not low again, the EOCB bit in the ADCR0 register will be set high and the analog to digital converter will be reset. It is the START bit that is used to control the overall start operation of the internal analog to digital converter.

The EOCB bit in the ADCR0 register is used to indicate when the analog to digital conversion process is complete. This bit will be automatically set to “0” by the microcontroller after a conversion cycle has ended. In addition, the corresponding A/D interrupt request flag will be set in the interrupt control register, and if the interrupts are enabled, an appropriate internal interrupt signal will be generated. This A/D internal interrupt signal will direct the program flow to the associated A/D internal interrupt address for processing. If the A/D internal interrupt is disabled, the microcontroller can be used to poll the EOCB bit in the ADCR0 register to check whether it has been cleared as an alternative method of detecting the end of an A/D conversion cycle.

A/D Converter Clock Source

The clock source for the A/D converter, which originates from the system clock f_{SYS}, can be chosen to be either f_{SYS} or a subdivided version of f_{SYS}. The division ratio value is determined by the ADCK2~ADCK0 bits in the ADCR1 register.

Although the A/D clock source is determined by the system clock f_{SYS}, and by bits ADCK2~ADCK0, there are some limitations on the maximum A/D clock source speed that can be selected. As the minimum value of permissible A/D clock period, t_{ADCK}, is 0.5μs, care must be taken for system clock frequencies equal to or greater than 4MHz. For example, if the system clock operates at a frequency of 4MHz, the ADCK2~ADCK0 bits should not be set to “000”. Doing so will give A/D clock periods that are less than the minimum A/D clock period which may result in inaccurate A/D conversion values. Refer to the following table for examples, where values marked with an asterisk * show where, depending upon the device, special care must be taken, as the values may be less than the specified minimum A/D Clock Period.

A/D Clock Period Examples

f_{SYS}	A/D Clock Period (t_{ADCK})							
	ADCK2, ADCK1, ADCK0 =000 (f_{SYS})	ADCK2, ADCK1, ADCK0 =001 (f_{SYS}/2)	ADCK2, ADCK1, ADCK0 =010 (f_{SYS}/4)	ADCK2, ADCK1, ADCK0 =011 (f_{SYS}/8)	ADCK2, ADCK1, ADCK0 =100 (f_{SYS}/16)	ADCK2, ADCK1, ADCK0 =101 (f_{SYS}/32)	ADCK2, ADCK1, ADCK0 =110 (f_{SYS}/64)	ADCK2, ADCK1, ADCK0 =111
1MHz	1μs	2μs	4μs	8μs	16μs	32μs	64μs	Undefined
2MHz	500ns	1μs	2μs	4μs	8μs	16μs	32μs	Undefined
4MHz	250ns*	500ns	1μs	2μs	4μs	8μs	16μs	Undefined
8MHz	125ns*	250ns*	500ns	1μs	2μs	4μs	8μs	Undefined
12MHz	83ns*	167ns*	333ns*	667ns	1.33μs	2.67μs	5.33μs	Undefined
16MHz	62.5ns*	125ns*	250ns*	500ns	1μs	2μs	4μs	Undefined
32MHz	31.25ns*	62.5ns*	125ns*	250ns*	500ns	1μs	2μs	Undefined

A/D Input Pins

All of the A/D analog input pins are pin-shared with the I/O pins on Port 4 function. The ACE7~ACE0 bits in the ADCR2 registers, determine whether the input pins are setup as A/D converter analog inputs or I/O function. If the ACE7~ACE0 bits for its corresponding pin is set high then the pin will be setup to be an A/D converter input and the original pin functions disabled. In this way, pins can be changed under program control to change their function between A/D inputs and I/O function. All pull-high resistors, which are setup through register programming, will be automatically disconnected if the pins are setup as A/D inputs. Note that it is not necessary to first setup the A/D pin as an input in the P4 port control register to enable the A/D input as when the ACE7~ACE0 bits enable an A/D input, the status of the port control register will be overridden.

The A/D converter has its own reference voltage pin, VREF, however the reference voltage can also be supplied from the power supply pin or internal voltage reference, a choice which is made through the VREFAS and VREFIS bits in the ADCR1 register. The analog input values must not be allowed to exceed the value of V_{REF}.

Temperature Sensor

A temperature sensor circuit is provided to measure the temperature which the designer can use to adjust some measured parameters. The temperature sensor output voltage is proportional to the temperature increment and can be amplified by the PGA. The accompanying diagram illustrates the basic relationship between the measured temperature and the voltage output. However, the designer should consider that the temperature sensor output voltage might be affected by the manufacturing process.

The ADC temperature sensor input channel is selected by the ACS4 bit. The TSEN bit in the ADCR1 register controls the temperature sensor enable/disable function. When the function is disabled, the temperature sensor defaults to an unknown state and any A/D conversion performed on the sensor will generate undefined data.

A/D Reference Voltage Source

The A/D can obtain its reference voltage from three different sources, the VCCA3 power supply pin, an externally supplied reference voltage supplied on pin VREF or from the internal voltage reference generator. Two bits control which reference source is selected, these are the VREFIS and VREFAS bits.

A/D Converter Voltage Reference Select

VREFIS	VREFAS	Reference Source
0	0	VCCA3 pin
0	1	Externally supplied on VREF pin
1	0	VCCA3 pin
1	1	Internal Voltage Reference Generator

Summary of A/D Conversion Steps

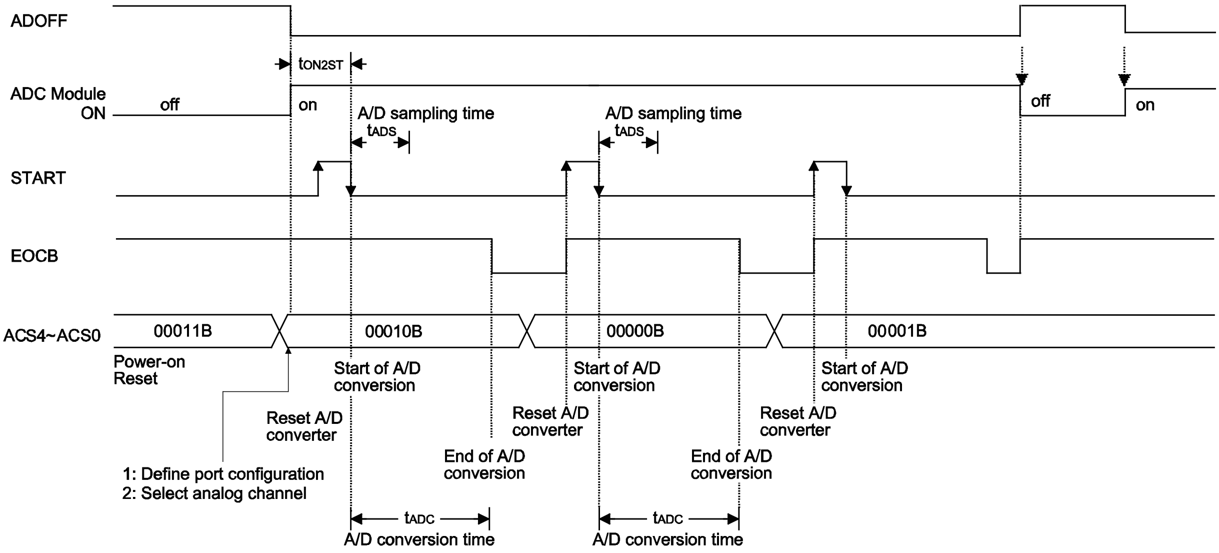
The following summarises the individual steps that should be executed in order to implement an A/D conversion process.

- Step 1
  Select the required A/D conversion clock by correctly programming bits ADCK2~ADCK0 in the ADCR1 register and select the converted data storage format using the ADRFS bit.
- Step 2
  Enable the A/D by clearing the ADOFF bit in the ADCR0 register to zero and select the PGA gain using the ADPGA register according to the dynamic range of the analog input signal.
- Step 3
  Select which channel is to be connected to the internal A/D converter by correctly programming the ACS4~ACS0 bits which are also contained in the ADCR1 and ADCR0 register.
- Step 4
  Select which pins are to be used as A/D inputs and configure them by correctly programming the ACE7~ACE0 bits in the ADCR2 register.
- Step 5
  If the interrupts are to be used, the interrupt control registers must be correctly configured to ensure the A/D converter interrupt function is active. The master interrupt control bit, EAL, and the A/D converter interrupt bit, EADC, must both be set high to do this.
- Step 6
  The analog to digital conversion process can now be initialised by setting the START bit in the ADCR0 register from low to high and then low again. Note that this bit should have been originally cleared to zero.
- Step 7
  To check when the analog to digital conversion process is complete, the EOCB bit in the ADCR0 register can be polled. The conversion process is complete when this bit changes from high to low. When this occurs the A/D data registers ADRL and ADRH can be read to obtain the conversion value. As an alternative method, if the interrupts are enabled and the stack is not full, the program can wait for an A/D interrupt to occur.

Note: When checking for the end of the conversion process, if the method of polling the EOCB bit in the ADCR0 register is used, the interrupt enable step above can be omitted.

A/D Conversion Timing

The accompanying diagram shows graphically the various stages involved in an analog to digital conversion process and its associated timing. After an A/D conversion process has been initiated by the application program, the microcontroller internal hardware will begin to carry out the conversion, during which time the program can continue with other functions. The time taken for the A/D conversion is 16 t_{ADCK} where t_{ADCK} is equal to the A/D clock period.



A/D Conversion Timing

Programming Considerations

During microcontroller operations where the A/D converter is not being used, the A/D internal circuitry can be switched off to reduce power consumption, by setting bit ADOFF high in the ADCR0 register. When this happens, the internal A/D converter circuits will not consume power irrespective of what analog voltage is applied to their input lines. If the A/D converter input lines are used as normal I/Os, then care must be taken as if the input voltage is not at a valid logic level, then this may lead to some increase in power consumption.

A/D Transfer Function

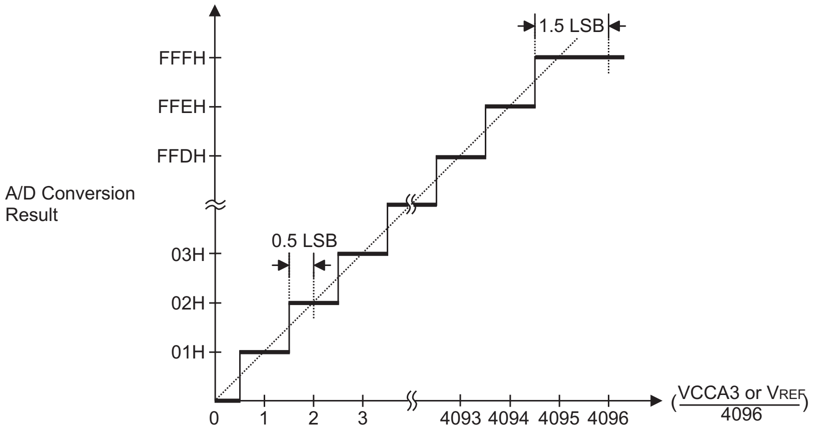
As the converted data is 12-bit wide, its full-scale converted digitised value is equal to FFFH. Since the full-scale analog input value is equal to the VCCA3 or V_{REF} voltage, this gives a single bit analog input value of VCCA3 or V_{REF} divided by 4096.

$$1\ \text{LSB}=(\text{VCCA3 or } V_{REF})/4096$$

The A/D Converter input voltage value can be calculated using the following equation:

$$\text{A/D input voltage}=\text{PGA Gain}\times\text{A/D digital value}\times(\text{VCCA3 or } V_{REF})/4096$$

The diagram shows the ideal transfer function between the analog input value and the digitised output value for the A/D converter. Except for the digitised zero value, the subsequent digitised values will change at a point 0.5 LSB below where they would change without the offset, and the last full scale digitised value will change at a point 1.5 LSB below the VCCA3 or V_{REF} level.



Ideal A/D Transfer Function (PGA=1)

31 Digital to Analog Converter – DAC

All devices include a Digital to Analog Converter permitting the conversion of a 12-bit digital value into an analog voltage. An additional programmable attenuation control function provides further flexibility over the overall input/output transfer function.

DAC Register Description

A pair of data register, DAL and DAH, store the 12-bit digital value which is to be converted, while an additional control register, DACTRL, controls the attenuation level, reference voltage select and enable/disable control.

DAH Register
SFR Address: B5h

Bit	7	6	5	4	3	2	1	0
Name	D11	D10	D9	D8	D7	D6	D5	D4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	0	0	0	0	0	0	0

Bit 7~0 **D11~D4:** DAC output data bit 11~4

DAL Register
SFR Address: B4h

Bit	7	6	5	4	3	2	1	0
Name	D3	D2	D1	D0	—	—	—	—
R/W	R/W	R/W	R/W	R/W	—	—	—	—
POR	0	0	0	0	—	—	—	—

Bit 7~4 **D3~D0:** DAC output data bit 3~0

Bit 3~0 Unimplemented, read as “0”

DACTRL Register
SFR Address: B3h

Bit	7	6	5	4	3	2	1	0
Name	VOL2	VOL1	VOL0	—	—	—	VREFDS	DACEN
R/W	R/W	R/W	R/W	—	—	—	R/W	R/W
POR	0	0	0	—	—	—	0	0

Bit 7~5 **VOL2~VOL0:** DAC attenuation control

There are 8 levels of DAC attenuation selected using these three bits. The accompanying table illustrates the relationship between the DAC attenuation control bits and the DAC data output.

DACOUT VOL	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000	D11	D11B	D11B	D11B	D11B	D11B	D11B	D11B	D10	D9	D8	D7
001	D11	D11B	D11B	D11B	D11B	D11B	D11B	D10	D9	D8	D7	D6
010	D11	D11B	D11B	D11B	D11B	D11B	D10	D9	D8	D7	D6	D5
011	D11	D11B	D11B	D11B	D11B	D10	D9	D8	D7	D6	D5	D4
100	D11	D11B	D11B	D11B	D10	D9	D8	D7	D6	D5	D4	D3
101	D11	D11B	D11B	D10	D9	D8	D7	D6	D5	D4	D3	D2
110	D11	D11B	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
111	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Bit 4~2 Unimplemented, read as “0”

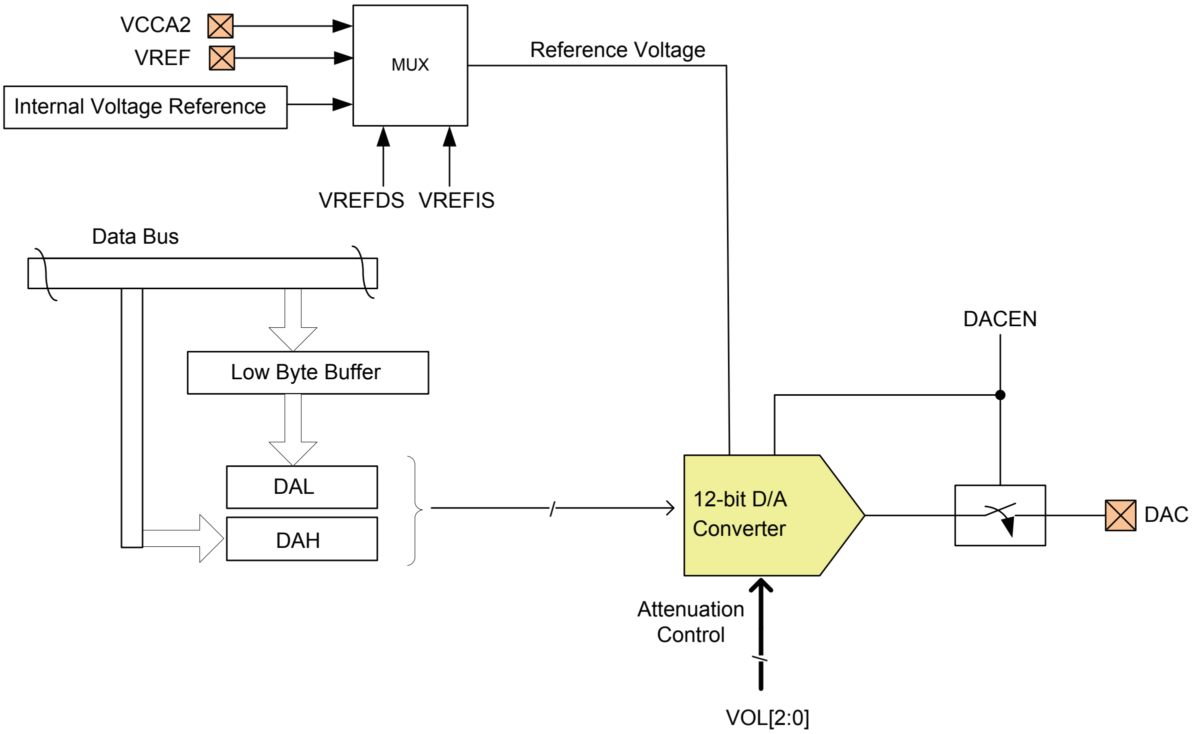
Bit 1 **VREFDS:** DAC reference voltage select bit
0: VCCA2 external power supply pin
1: Externally supplied on VREF pin or internal voltage reference generator

Bit 0 **DACEN:** DAC enable control bit
0: Disable
1: Enable

DAC Operation

The DAL and DAH registers contain the digital value to be converted. The DACEN bit in the DACTRL register provides overall enable/disable control. When set high, the DAC output pin will be enabled and the original I/O pin shared function disabled. Clearing this bit to zero will disable the DAC and reduce any associated power consumption. The DAC attenuation control is provided by the VOL0~VOL2 bits in the DACTRL register providing an 8-level attenuation control. These bits rotate the digital DAC value thus providing a divide or multiply by two functions for each shift left or right. If the DAC circuit is not enabled, any DAH/DAL values will be invalid. The VREFDS and VREFIS bits select if the DAC reference is to be sourced from VCCA2, VREF pin or the internal reference voltage. Bits 0~3 of the DAL register are always read as zero.

The accompanying diagram illustrates the DAC basic operational block diagram.



DAC Basic Operational Block Diagram

DAC Reference Voltage Source

The DAC can obtain its reference voltage from three different sources, the VCCA2 power supply pin, an externally supplied reference voltage supplied on pin VREF or from the internal voltage reference generator. Two bits control which reference source is selected, these are the VREFIS and VREFDS bits.

DAC Converter Voltage Reference Select

VREFIS	VREFDS	Reference Source
0	0	VCCA2 pin
0	1	Externally supplied on VREF pin
1	0	VCCA2 pin
1	1	Internal Voltage Reference Generator

Programming Considerations

Note that data written to the two DAC registers must be implemented in a specific way. Any data written into the DAH register will load both the data into the DAL and DAH registers simultaneously and influence the DAC output at the same time. However writing data to the DAL register will only place the data into a low byte buffer and not directly into the DAL register. For this reason, writing data to the DAL register should be followed by a write instruction to the DAH register.

32 Voltage Reference Generator

All devices include a bandgap circuit based internal voltage reference generator which can supply a temperature stable reference voltage for use by the internal A/D converter and DAC.

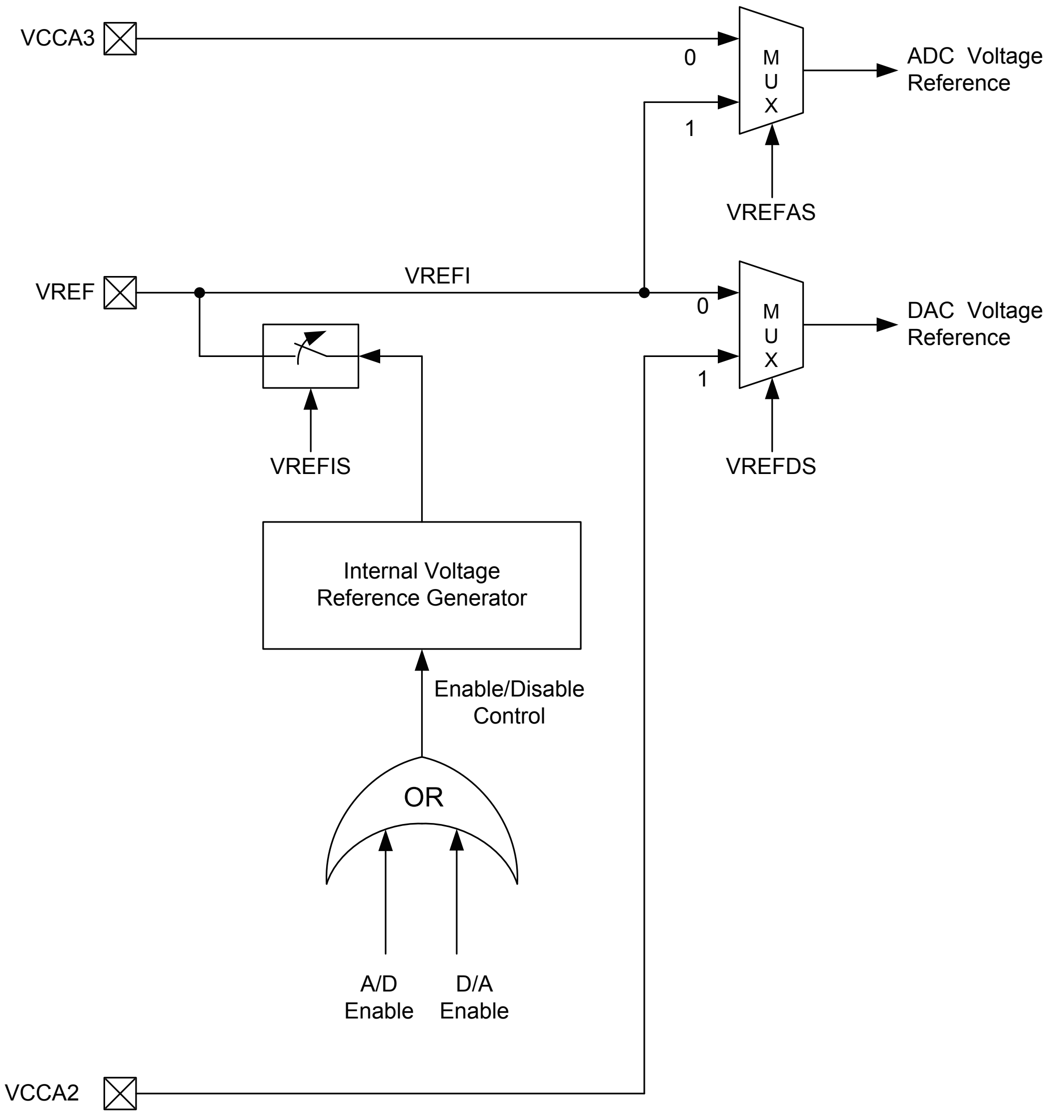
Voltage Reference Generator Operation

The voltage reference circuit will be automatically enabled when either the A/D converter or DAC is enabled. If both the A/D converter and DAC are disabled then the generator will be disabled thus conserving power.

Internal Voltage Reference Enable/Disable Control

ADC	DAC	Voltage Reference
Disabled	Disabled	Disabled
Enabled	Disabled	Enabled
Disabled	Enabled	Enabled
Enabled	Enabled	Enabled

The internal Voltage Reference Generator output is pin VREF and can be used as a reference source for other circuits if loaded lightly. A suitable capacitor should be connected to this pin to enhance voltage stability. If the internal Voltage Reference Generator is enabled then the VREF pin will act as an output pin and must be treated accordingly. However if the internal Voltage Reference Generator is disabled, then the VREF pin will act as an input pin to enable an externally supplied reference voltage to be provided if required.



Voltage Reference Generator Block Diagram

The A/D converter and DAC reference voltage is selected by the VREFIS, VREFAS and VREFDS control bits. When the VREFIS bit is enabled the internal voltage reference will be routed to pin VREF and can be selected for use by the A/D converter or DAC, using the VREFAS and VREFDS bits.

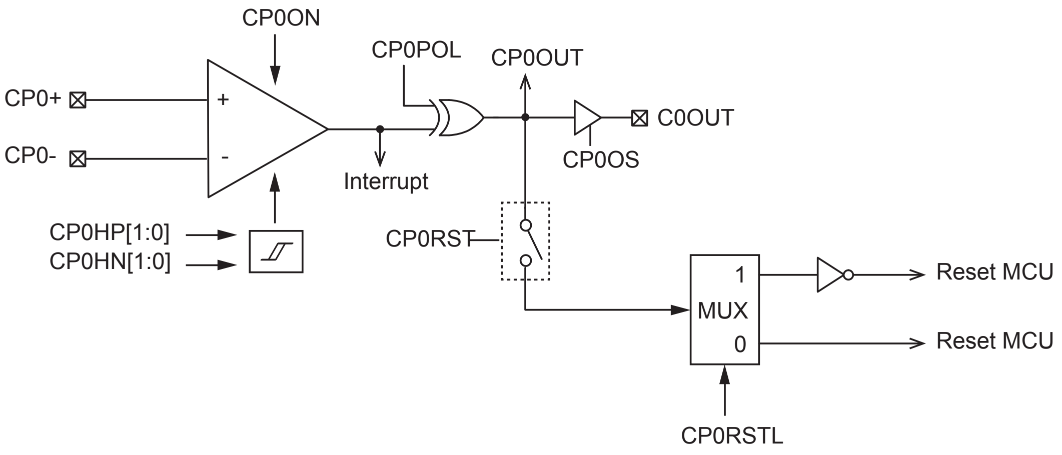
33 Comparators

Two independent analog comparators are contained within these devices. These functions offer flexibility via their register controlled features such as power-down, polarity select, hysteresis, interrupt, wake-up, output path selection etc. In sharing their pins with normal I/O pins the comparators do not waste precious I/O pins if there functions are unused. One comparator also has the ability to reset the MCU.

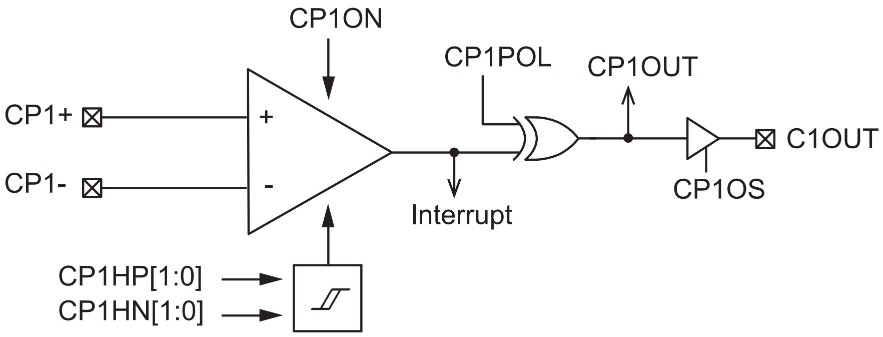
Comparator Operation

Each device contains two comparator functions which are used to compare two analog voltages and provide an output based on their difference. Full control over the two internal comparators is provided via four control registers, CP0CR, CP1CR, CPHCR and CPICR. The comparator output is recorded via a bit in their respective control register, but can also be transferred out onto a shared I/O pin. Additional comparator functions include, output polarity, hysteresis functions and power-down control.

Any pull-high resistors connected to the shared comparator input pins will be automatically disconnected when the comparator is enabled. As the comparator inputs approach their switching level, some spurious output signals may be generated on the comparator output due to the slow rising or falling nature of the input signals. This can be minimised by selecting the hysteresis function will apply a small amount of positive feedback to the comparator. Ideally the comparator should switch at the point where the positive and negative inputs signals are at the same voltage level, however, unavoidable input offsets introduce some uncertainties here. The hysteresis function, if enabled, also increases the switching offset value. The Comparator Hysteresis control function is selected by the CPHCR register. In addition, the comparator 0 provides the Comparator Output Reset MCU function which is decided by the CP0RST and CP0RSTL bits in the CP0CR register.



Comparator 0



Comparator 1

Comparator Registers

There are four registers for overall comparator operation. The CP0CR and CP1CR registers are used to control the respective comparators settings for the Comparator 0 and Comparator 1 while the CPHCR register is used to manage the hysteresis selection for these two comparators. In addition, the CPICR register control the comparators interrupt settings. The accompanying register table illustrates the control registers list.

Comparator Registers List

Register Name	Bit							
	7	6	5	4	3	2	1	0
CP0CR	—	CP0ON	CP0POL	CP0OUT	CP0OS	CP0RSTL	CP0RST	—
CP1CR	—	CP1ON	CP1POL	CP1OUT	CP1OS	—	—	—
CPHCR	CP1HP1	CP1HP0	CP1HN1	CP1HN0	CP0HP1	CP0HP0	CP0HN1	CP0HN0
CPICR	CP1IF	CP1IEN	CP1P1	CP1P0	CP0IF	CP0IEN	CP0P1	CP0P0

CP0CR Register
SFR Address: DEh

Bit	7	6	5	4	3	2	1	0
Name	—	CP0ON	CP0POL	CP0OUT	CP0OS	CP0RSTL	CP0RST	—
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	—
POR	—	0	0	0	1	0	0	—

Bit 7 Unimplemented, read as “0”

Bit 6 **CP0ON:** Comparator 0 On/Off control
0: Off
1: On
This is the Comparator 0 on/off control bit. If the bit is zero the comparator will be switched off and no power consumed even if analog voltages are applied to its inputs. For power sensitive applications this bit should be cleared to zero if the comparator is not used or before the device enters the Power-Down or IDLE mode.

Bit 5 **CP0POL:** Comparator 0 output polarity
0: Output not inverted
1: Output inverted
This is the comparator 0 polarity bit. If the bit is zero then the CP0OUT bit will reflect the non-inverted output condition of the comparator. If the bit is high the comparator CP0OUT bit will be inverted.

Bit 4 **CP0OUT:** Comparator 0 output bit
CP0POL=0
0: CP0+ < CP0-
1: CP0+ > CP0-
CP0POL=1
0: CP0+ > CP0-
1: CP0+ < CP0-
This bit stores the comparator 0 output bit. The polarity of the bit is determined by the voltages on the comparator 0 inputs and by the condition of the CP0POL bit.

Bit 3 **CP0OS:** Comparator 0 output path select
0: C0OUT pin
1: Internal use
This is the comparator 0 output path select control bit. If the bit is set to “0” and the CP0ON bit is “1” the comparator 0 output is connected to an external C0OUT pin. If the bit is set to “1” or the CP0ON bit is “1” the comparator 0 output signal is only used internally by the device allowing the shared comparator output pin to retain its normal I/O operation.

Bit 2 **CP0RSTL:** Output reset signal select
0: CP0OUT=0 will reset the MCU
1: CP0OUT=1 will reset the MCU
The CP0RST bit should be set high first to enable this function.

Bit 1 **CP0RST:** Comparator 0 output to reset MCU function control
0: Disable
1: Enable

Bit 0 Unimplemented, read as “0”

CP1CR Register
SFR Address: DFh

Bit	7	6	5	4	3	2	1	0
Name	—	CP1ON	CP1POL	CP1OUT	CP1OS	—	—	—
R/W	—	R/W	R/W	R/W	R/W	—	—	—
POR	—	0	0	0	1	—	—	—

Bit 7 Unimplemented, read as "0"

Bit 6 **CP1ON:** Comparator 1 On/Off control
0: Off
1: On

This is the Comparator 1 on/off control bit. If the bit is zero the comparator will be switched off and no power consumed even if analog voltages are applied to its inputs. For power sensitive applications this bit should be cleared to zero if the comparator is not used or before the device enters the Power-Down or IDLE mode.

Bit 5 **CP1POL:** Comparator 1 output polarity
0: Output not inverted
1: Output inverted

This is the comparator 1 polarity bit. If the bit is zero then the CP1OUT bit will reflect the non-inverted output condition of the comparator. If the bit is high the comparator CP1OUT bit will be inverted.

Bit 4 **CP1OUT:** Comparator 1 output bit
CP1POL=0
0: CP1+ < CP1-
1: CP1+ > CP1-
CP1POL=1
0: CP1+ > CP1-
1: CP1+ < CP1-

This bit stores the comparator 1 output bit. The polarity of the bit is determined by the voltages on the comparator 0 inputs and by the condition of the CP1POL bit.

Bit 3 **CP1OS:** Comparator 1 output path select
0: C1OUT pin
1: Internal use

This is the comparator 0 output path select control bit. If the bit is set to "0" and the CP1ON bit is "1" the comparator 0 output is connected to an external C1OUT pin. If the bit is set to "1" or the CP1ON bit is "1" the comparator 0 output signal is only used internally by the device allowing the shared comparator output pin to retain its normal I/O operation.

Bit 2~0 Unimplemented, read as "0"

CPHCR Register
SFR Address: BDh

Bit	7	6	5	4	3	2	1	0
Name	CP1HP1	CP1HP0	CP1HN1	CP1HN0	CP0HP1	CP0HP0	CP0HN1	CP0HN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **CP1HP1, CP1HP0:** Comparator 1 Positive Hysteresis voltage level Control bits
00: Disabled
01: 3mV
10: 6mV
11: 12mV

Bit 5~4 **CP1HN1, CP1HN0:** Comparator 1 Negative Hysteresis voltage level Control bits
00: Disabled
01: 3mV
10: 6mV
11: 12mV

Bit 3~2 **CP0HP1, CP0HP0:** Comparator 0 Positive Hysteresis voltage level Control bits
00: Disabled
01: 3mV
10: 6mV
11: 12mV

Bit 1~0 **CP0HN1, CP0HN0:** Comparator 0 Negative Hysteresis voltage level Control bits
00: Disabled
01: 3mV
10: 6mV
11: 12mV

CPICR Register
SFR Address: BEh

Bit	7	6	5	4	3	2	1	0
Name	CP1IF	CP1IEN	CP1P1	CP1P0	CP0IF	CP0IEN	CP0P1	CP0P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CP1IF:** Comparator 1 Output Transition Interrupt Request Flag
0: No request
1: Interrupt request
Note that this flag should be cleared using the application program.

Bit 6 **CP1IEN:** Comparator 1 Output Transition Interrupt enable control
0: Disable
1: Enable

Bit 5~4 **CP1P1, CP1P0:** Comparator 1 Output Transition Interrupt settings for interrupt
00: Interrupt disabled
01: Comparator transition output from high to low will cause an interrupt
10: Comparator transition output from low to high will generate an interrupt
11: Comparator transition output from low to high or high to low will generate an interrupt

Bit 3 **CP0IF:** Comparator 0 Output Transition Interrupt Request Flag
0: Not request
1: Interrupt request
Note that this flag should be cleared using the application program.

Bit 2 **CP0IEN:** Comparator 0 Output Transition Interrupt enable control
0: Disable
1: Enable

Bit 1~0 **CP0P1, CP0P0:** Comparator 0 Output Transition Interrupt settings for interrupt
00: Interrupt disabled
01: Comparator transition output from high to low will generate an interrupt
10: Comparator transition output from low to high will generate an interrupt
11: Comparator transition output from low to high or high to low will generate an interrupt

Comparator Interrupt

Each also possesses its own interrupt function. When any one of the changes state, its relevant interrupt flag will be set, and if the corresponding interrupt enable bit is set, then a jump to its relevant interrupt vector will be executed. Note that it is the changing state of the CP0OUT or CP1OUT bit and not the output pin which generates an interrupt. If the microcontroller is in the Power-Down or IDLE Mode and the Comparator is enabled, then if the external input lines cause the Comparator output to change state, the resulting generated interrupt flag will also generate a wake-up. If it is required to disable a wake-up from occurring, then the interrupt function should be disabled before entering the Power-Down or IDLE Mode. Each of the comparators has the compare output transition settings to decide the interrupt request conditions. There are three options, compare output rising, falling or both rising and falling conditions, decided by the CPnP1 and CPnP0 bits in the CPICR register.

Comparator Reset Function

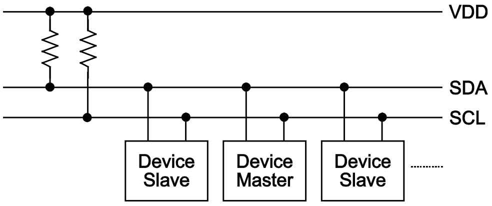
Comparator 0 has the ability to reset the device. The reset function of Comparator 0 can be enabled or disabled by the CP0RST bit in the CP0CR register after which the comparator polarity which will reset the device can be selected using the CP0RSTL bit.

Programming Considerations

If the comparator is enabled, it will remain active when the microcontroller enters the Power-Down or IDLE Mode, however as it will consume a certain amount of power, the user may wish to consider disabling it before the Power-Down or IDLE Mode is entered. As comparator pins are shared with normal I/O pins the I/O registers for these pins will be read as zero (port control register is “1”) or read as port data register value (port control register is “0”) if the comparator function is enabled.

34 I²C Serial Interface

The I²C interface is used to communicate with external peripheral devices such as sensors, EEPROM memory etc. Originally developed by Philips, it is a two line low speed serial interface for synchronous serial data transfer. The advantage of only two lines for communication, relatively simple communication protocol and the ability to accommodate multiple devices on the same bus has made it an extremely popular interface type for many applications.



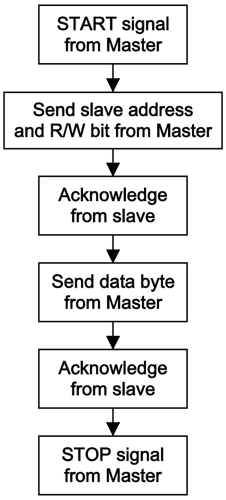
I²C Master Slave Bus Connection

I²C Interface Operation

The I²C serial interface is a two line serial interface. These lines are a serial data line, SDA, and serial clock line, SCL. As many devices may be connected together on the same bus, their outputs are both open drain types. For this reason it is necessary that external pull-high resistors are connected to these outputs. Note that no chip select line exists, as each device on the I²C bus is identified by a unique address which will be transmitted and received on the I²C bus.

When two devices communicate with each other on the bidirectional I²C bus, one is known as the master device and one as the slave device. Both master and slave can transmit and receive data, however, it is the master device that has overall control of the bus. The I²C meets the Philips I²C bus specification and supports all transfer modes from and to the I²C bus.

When the I²C is in the master mode, a variable baud rate setup is available using the I²C clock generator. The clock source is sourced from the system clock. The clock generator can be controlled using the I2CLK register. The clock generator is suppressed when the I²C is in the slave mode.



I²C Interface Operation Flow

I²C Registers

There are three control registers associated with the I²C bus, I2CCON, I2CADR and I2CSTA and one data register, I2CDAT. The I2CDAT register is used to store the data being transmitted and received on the I²C bus. Before the microcontroller writes data to the I²C bus, the actual data to be transmitted must be placed in the I2CDAT register. After the data is received from the I²C bus, the microcontroller can read it from the I2CDAT register. Any transmission or reception of data from the I²C bus must be made via the I2CDAT register. The I2CADR register holds the address of the device slave interface. This address is used by an external device when attempting to access it via I²C bus. The complete I²C interface operation is controlled by the I2CCON register. The I2CLK register controls the I²C clock frequency. In addition, the I²C Bus status can be reflected by the I2CSTA register.

I²C Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
I2CCON	—	ENS1	STA	STO	SI	AA	—	—
I2CLK	I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0
I2CSTA	IICS7	IICS6	IICS5	IICS4	IICS3	—	—	—
I2CDAT	D7	D6	D5	D4	D3	D2	D1	D0
I2CADR	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	GC

I2CCON Register
SFR Address: D8h

Bit	7	6	5	4	3	2	1	0
Name	—	ENS1	STA	STO	SI	AA	—	—
R/W	—	R/W	R/W	R/W	R/W	R/W	—	—
POR	—	0	0	0	0	0	—	—

Bit 7 Unimplemented, read as “0”

Bit 6 **ENS1:** I²C Enable Control
0: Disable
1: Enable
When the ENS1 bit is cleared to zero, the I²C interface will be disabled and will become high impedance and not affect the original pin-shared I/O pin function. When the ENS1 bit is set high, the I²C function is enabled and care should be taken regarding the related pin-shared I/O structure settings, such as disabling any internal pull up functions and any other circuits connected to these pins.

Bit 5 **STA:** I²C Start flag
0: No START condition on the I²C bus
1: START condition
When the STA bit is set high, the master device will check the I²C bus status first and if the bus is free a START condition will be generated.

Bit 4 **STO:** I²C Stop flag
0: No STOP condition on the I²C bus
1: Set STOP condition
When the STO bit is set high, the master device will transmit a STOP condition to the I²C bus.

Bit 3 **SI:** Serial Interrupt Request flag
0: No Interrupt request
1: Interrupt request
The SI bit will be set by hardware when one of the 25 out of 26 possible I^{2}C statuses is entered. The only state that does not set the SI bit is the state F8H, which indicates that no relevant state information is available. This bit must be cleared by the application program.

Bit 2 **AA:** I^{2}C Acknowledge Indication flag
0: No Acknowledge
1: Acknowledge
This bit indicates the type of acknowledge returned during the acknowledge cycle on the SCL pin. If this bit is cleared to low, a “no acknowledge” (high level on SDA) is returned during the acknowledge cycle. If the bit is set to high, an “acknowledge” (low level on SDA) is returned during the acknowledge cycle.
When AA=1, an “acknowledge” will be returned under the following conditions:
- The “own slave address” has been received
- The general call address has been received while the GC bit in the I2CADR register was set
- A data byte has been received while the I^{2}C was in the master receiver mode
- A data byte has been received while the I^{2}C was in the slave receiver mode

When AA= 0, a “not acknowledge” will be returned under the following conditions:
- A data byte has been received while the I^{2}C was in the master receiver mode
- A data byte has been received while the I^{2}C was in the slave receiver mode

Bit 1~0 Unimplemented, read as “0”

I2CLK Register
SFR Address: E9h

Bit	7	6	5	4	3	2	1	0
Name	I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	1	1	0	0	1

Bit 7~0 I^{2}C clock rate bit 7~0
The I^{2}C baud rate calculation is described as follow:

$$I^2C\_Baud\_Rate = \frac{f_{SYS}}{4*(I2CLK[7:0]+5)}$$

Here f_{SYS} is the system clock frequency.
For example, if the system clock is 12MHZ and I2CLK [7:0]=19h(25), then
I^{2}C_baud_rate=12MHz/(4*(25+5))=100Kbit/Sec

I2CSTA Register
SFR Address: DDh

Bit	7	6	5	4	3	2	1	0
Name	IICS7	IICS6	IICS5	IICS4	IICS3	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	1	1	1	1	1	—	—	—

Bit 7~3 **IICS7~IICS3:** I^{2}C Status Code

These Read-only bits are used to indicate the I^{2}C Status code. Refer to the I^{2}C Status Code section for details. The contents of the I2CSTA register is only defined when the SI bit is set high.

Bit 2~0 Unimplemented, read as "0"

I2CDAT Register
SFR Address: DAh

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Note: The I2CDAT register is used as the I^{2}C transmitted or received data register.

I2CADR Register
SFR Address: DBh

Bit	7	6	5	4	3	2	1	0
Name	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~1 **IICA6~IICA0:** I^{2}C slave address

IICA6~IICA0 is the I^{2}C slave address bit 6~bit 0.

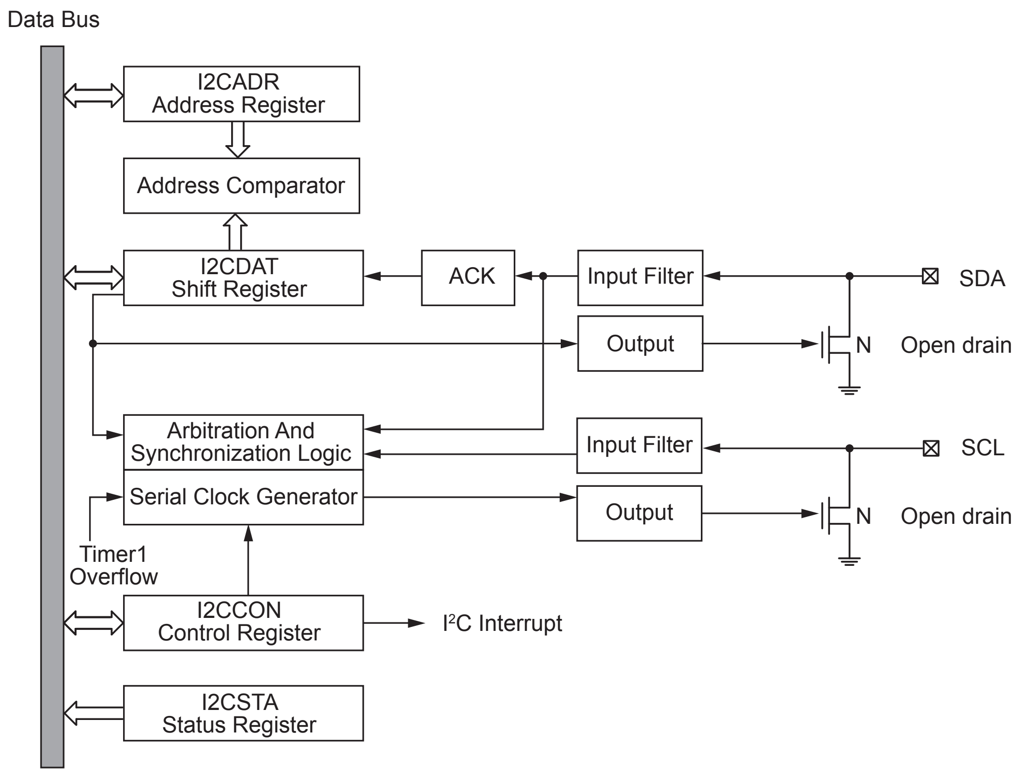
When a master device, which is connected to the I^{2}C bus, sends out an address, which matches the slave address in the I2CADR register, the slave device will be selected.

Bit 0 **GC:** General Call Address Acknowledge control bit

0: General Call Address is ignored
1: General Call Address is recognised

This bit is used to enable the General Call Address (00H) recognition function.

The I2CADR register contains the slave address for the I^{2}C interface. In the Slave mode, the IICA6~IICA0 bits represent a 7-bit slave address. The GC bit is used to enable the recognition of the general call address (0x00). If the GC bit is set to "1", the general call address recognition function will be enabled. Otherwise, the general call address will be ignored. In the master mode, the contents of this I2CADR register will be ignored.



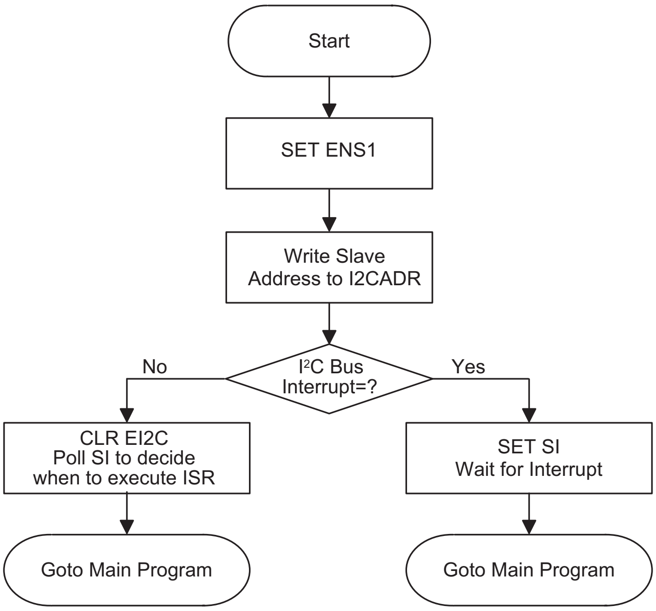
I^{2}C Block Diagram

I^{2}C Bus Communication

Communication on the I^{2}C bus requires four separate steps, a START signal, a slave device address transmission, a data transmission and finally a STOP signal. When a START signal is placed on the I^{2}C bus, all devices on the bus will receive this signal and be notified of the imminent arrival of data on the bus. The first seven bits of the data will be the slave address with the first bit being the MSB. If the address of the slave device matches that of the transmitted address, which means one of the I^{2}C states is matched, the SI bit in the I2CCON register will be set and an I^{2}C interrupt will be generated. After entering the interrupt service routine, the devices must first check the status of the I2CSTA register to determine the interrupt source originating condition. The SI bit is set by hardware when one of 25 out of 26 possible I^{2}C states is entered. The only state that does not set the SI bit is the state F8H, which indicates that no relevant state information is available The SI bit must be cleared by the application program.

During a data transfer, note that after the 7-bit slave address has been transmitted, the following bit, which is the 8th bit, is the read/write bit. This bit will be checked by the slave device to determine whether to go into transmit or receive mode. Before any transfer of data to or from the I^{2}C bus, the microcontroller must initialise the bus. The following are the steps to achieve this:

- Step 1
  Set the ENS1 bit in the I2CCON register high to enable the I^{2}C bus.
- Step 2
  Write the slave address of the device to the I^{2}C bus address register I2CADR.
- Step 3
  Set the EI2C interrupt enable bit of the interrupt control register to enable the I^{2}C interrupt.


Start
SET ENS1
Write Slave Address to I2CADR
I²C Bus Interrupt=?
No
Yes
CLR EI2C Poll SI to decide when to execute ISR
SET SI Wait for Interrupt
Goto Main Program
Goto Main Program

I^{2}C Bus Initialisation Flow Chart

I^{2}C Bus Start Signal

The START signal can only be generated by the master device connected to the I^{2}C bus and not by the slave device. This START signal will be detected by all devices connected to the I^{2}C bus. When detected, this indicates that the I^{2}C bus is busy and therefore the STA bit will be set. A START condition occurs when a high to low transition on the SDA line takes place when the SCL line remains high.

Slave Address

The transmission of a START signal by the master will be detected by all devices on the I^{2}C bus. To determine which slave device the master wishes to communicate with, the address of the slave device will be sent out immediately following the START signal. All slave devices, after receiving this 7-bit address data, will compare it with their own 7-bit slave address. If the address sent out by the master matches the internal address of the microcontroller slave device, or if the general call address, 00H, is received when the GC bit in the I2CADR register is set high, then an internal I^{2}C bus interrupt signal will be generated. The next bit following the address, which is the 8th bit, defines the read/write status. The slave device will then transmit an acknowledge bit, which is a low level, as the 9th bit. The slave device will also set the flag SI when the addresses match.

As an I^{2}C bus interrupt will take place if one of the possible I^{2}C states is matched when the program enters the interrupt subroutine, the I2CSTA register should be examined to see, for example, whether the interrupt source has come from a matching slave address or from the completion of a data byte transfer. When a slave address is matched, the device must be placed in either the transmit mode and then data written to the I2CDAT register, or in the receive mode where it must implement a dummy read from the I2CDAT register to release the SCL line. Refer to the I^{2}C Status Code section for details.

I^{2}C Bus Read/Write Signal

The Read/Write bit, so called as R/W bit, is located in the 8th bit of the address data in the I2CDAT register. The R/W bit is set high to indicate a read operation and cleared low to indicate a write operation. The direction bit defines whether the slave device wishes to read data from the I^{2}C bus or write data to the I^{2}C bus. The slave device should examine this bit to determine if it is to be a transmitter or a receiver. If the R/W bit is “1” then this indicates that the master device wishes to read data from the I^{2}C bus, therefore the slave device must be setup to send data to the I^{2}C bus as a transmitter. If the R/W bit is “0” then this indicates that the master wishes to send data to the I^{2}C bus, therefore the slave device must be setup to read data from the I^{2}C bus as a receiver.

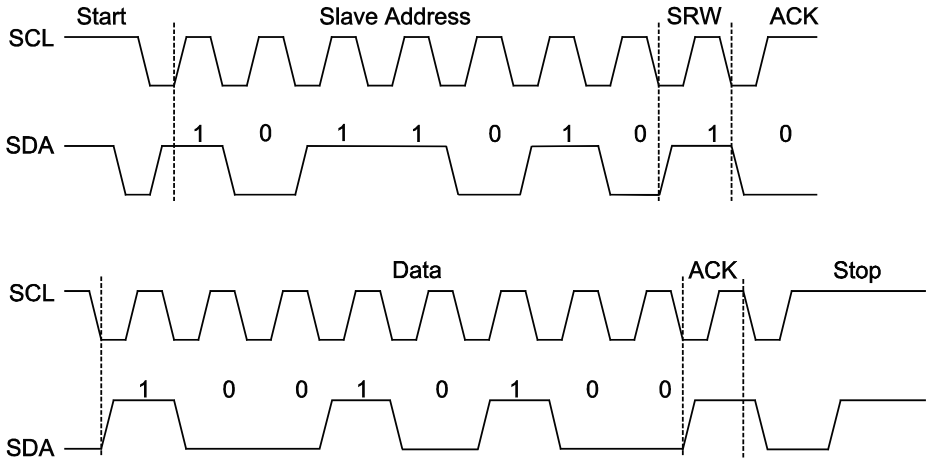
I^{2}C Bus Slave Address Acknowledge Signal

After the master has transmitted a calling address, any slave device on the I^{2}C bus, whose own internal address matches the calling address, must generate an acknowledge signal. The acknowledge signal will inform the master that a slave device has accepted its calling address. If no acknowledge signal is received by the master then a STOP signal must be transmitted by the master to end the communication. When the SI flag is high, the addresses have matched and the slave device must check the R/W bit, to determine if it is to be a transmitter or a receiver. If the R/W bit is high, the slave device should be setup to be a transmitter, and then the microcontroller slave device should be setup as a receiver.

I^{2}C Bus Data and Acknowledge Signal

The transmitted data is 8-bits wide and is transmitted after the slave device has acknowledged receipt of its slave address. The order of serial bit transmission is the MSB first and the LSB last. After receipt of 8-bits of data, the receiver must transmit an acknowledge signal, a low level, before it can receive the next data byte. If the slave transmitter does not receive an acknowledge bit signal from the master receiver, then the slave transmitter will release the SDA line to allow the master to send a STOP signal to release the I^{2}C Bus. The corresponding data will be stored in the I2CDAT register. If setup as a transmitter, the slave device must first write the data to be transmitted into the I2CDAT register. If setup as a receiver, the slave device must read the transmitted data from the I2CDAT register.

The AA bit in the I2CCON register bit indicates the type of acknowledge returned during the acknowledge cycle on the SCL pin. If this bit is cleared to zero, a "not acknowledge" (high level on SDA) is returned during the acknowledge cycle. If the bit is set to high, an "acknowledge" (low level on SDA) is returned during the acknowledge cycle. The slave device, which is setup as a transmitter will check the AA bit in the I2CCON register to determine if it is to send another data byte, if not then it will release the SDA line and await the receipt of a STOP signal from the master.



S=Start (1 bit)
SA=Slave Address (7 bits)
SR=SRW bit (1 bit)
M=Slave device send acknowledge bit (1 bit)
D=Data (8 bits)
A=ACK (RXAK bit for transmitter, TXAK bit for receiver 1 bit)
P=Stop (1 bit)

S	SA	SR	M	D	A	D	A	--------	S	SA	SR	M	D	A	D	A	--------	P

Note: * When a slave address is matched, the device must be placed in either the transmit mode and then write data to the I2CDAT register, or in the receive mode where it must implement a dummy read from the I2CDAT register to release the SCL line.

I^{2}C Communication Timing Diagram

I^{2}C Status Codes

The I2CSTA register reflects the current status of the I^{2}C interface. The three least significant bits of this register are always zero. There are 26 possible status codes, presented in the accompanying tables. When any one of 25 out of a total of 26 possible I^{2}C states is entered, an interrupt is requested. The only state that does not generate an interrupt is the F8h state. The contents of the I2CDAT register is only available when an I^{2}C interrupt takes place and the SI bit is set high. This register is read-only and should not be written to by the application program. In the table below, the term "SLA" means the slave address, "R" means the R/W bit=1 which are transferred together with the slave address, "W" means the R/W bit=0 transferred together with the slave address.

I^{2}C Status in Master Transmitter Mode

Status Code	Status of the I^{2}C	Application software response					Next action taken by the I^{2}C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
08H	START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK will be received
10H	Repeated START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK will be received
		Load SLA+W	X	0	0	X	SLA+R will be transmitted; I^{2}C will be switched to "Master receiver" mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK will be received
		No action	1	0	0	X	Repeated START will be transmitted;
		No action	0	1	0	X	STOP condition will be transmitted; the "STO" flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; the "STO" flag will be reset
20H	SLA+W has been transmitted; "not ACK" has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK will be received
		No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted; the "STO" flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; the "STO" flag will be reset
28H	Data byte in I2CDAT has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		No action	1	0	0	X	Repeated START will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted; the "STO" flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in I2CDAT has been transmitted	data byte	0	0	0	X	Data byte will be transmitted; ACK will be received
		No action	1	0	0	X	Repeated START will be transmitted;
		No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R/W or data bytes	No action	0	0	0	X	I^{2}C bus will be released; the "not addressed slave" state will be entered
		No action	1	0	0	X	A START condition will be transmitted when the bus becomes free

I^{2}C Status in Master Receiver Mode

Status Code	Status of the I^{2}C	Application software response					Next action taken by the I^{2}C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
08H	START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK will be received
10H	Repeated START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK will be received
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; I^{2}C will be switched to “master transmitter” mode
38H	Arbitration lost in “not ACK” bit	No action	0	0	0	X	I^{2}C bus will be released; I^{2}C will enter a “slave” mode
		No action	1	0	0	X	A start condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No action	0	0	0	0	Data byte will be received; “not ACK” will be returned
		No action	0	0	0	1	Data byte will be received; ACK will be returned
48H	SLA+R has been transmitted; “not ACK” has been received	No action	1	0	0	X	Repeated START condition will be transmitted
		No action	0	1	0	X	STOP condition will be transmitted; the “STO” flag will be reset
		No action	1	1	0	X	STOP condition followed by START condition will be transmitted; the “STO” flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received; “not ACK” will be returned
		Read data byte	0	0	0	1	Data byte will be received; ACK will be returned
58H	Data byte has been received; “not ACK” has been returned	Read data byte	1	0	0	X	Repeated START condition will be transmitted
		Read data byte	0	1	0	X	STOP condition will be transmitted; the “STO” flag will be reset
		Read data byte	1	1	0	X	STOP condition followed by START condition will be transmitted; the “STO” flag will be reset

I²C Status in Slave Receiver Mode

Status Code	Status of the I²C	Application software response					Next action taken by the I²C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
60H	Own SLA+W has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/W as master; own SLA+W has been received, ACK returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
70H	General call address (00H) has been received; ACK has been returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
78H	Arbitration lost in SLA+R/W as master; general call address has been received, ACK returned	No action	X	0	0	0	Data byte will be received and "not ACK" will be returned
		No action	X	0	0	1	Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK returned	read data byte	X	0	0	0	Data byte will be received and "not ACK" will be returned
		read data byte	X	0	0	1	Data byte will be received and ACK will be returned
88H	Previously addressed with own SLA; DATA byte has been received; "not ACK" returned	Read data byte	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		Read data byte	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		Read data byte	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free
90H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte	X	0	0	0	Data byte will be received and "not ACK" will be returned
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned
98H	Previously addressed with general call address; DATA has been received; not ACK returned	Read data byte	0	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address
		Read data byte	0	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized
		Read data byte	1	0	0	0	Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

Status Code	Status of the I^{2}C	Application software response					Next action taken by the I^{2}C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
A0H	STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No action	0	0	0	0	Switched to “not addressed slave” mode; no recognition of own slave address or general call address
		No action	0	0	0	1	Switched to “not addressed slave” mode; own slave address or general call address will be recognized
		No action	1	0	0	0	Switched to “not addressed slave” mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		No action	1	0	0	1	Switched to “not addressed slave” mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

I^{2}C Status in Slave Transmitter Mode

Status Code	Status of the I^{2}C	Application software response					Next action taken by the I^{2}C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/W as master own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B8H	Data byte has been transmitted; ACK has been received	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
C0H	Data byte has been transmitted; not ACK has been received	No action	0	0	0	0	Switched to “not addressed slave” mode; no recognition of own slave address or general call address
		No action	0	0	0	1	Switched to “not addressed slave” mode; own slave address or general call address will be recognized
		No action	1	0	0	0	Switched to “not addressed slave” mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		No action	1	0	0	1	Switched to “not addressed slave” mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

Status Code	Status of the I²C	Application software response					Next action taken by the I²C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
C8H	Last data byte has been transmitted; ACK has been received	No action	0	0	0	0	Switched to “not addressed slave” mode; no recognition of own slave address or general call address
		No action	0	0	0	1	Switched to “not addressed slave” mode; own slave address or general call address will be recognized
		No action	1	0	0	0	Switched to “not addressed slave” mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free
		No action	1	0	0	1	Switched to “not addressed slave” mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free

I²C Status: Miscellaneous States

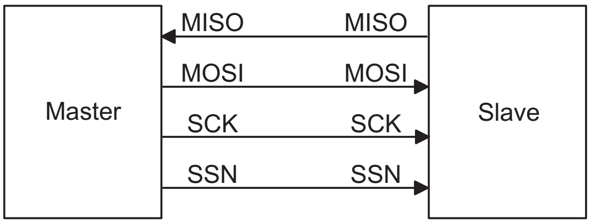
Status Code	Status of the I²C	Application software response					Next action taken by the I²C hardware
		to/from I2CDAT	to I2CCON				
			STA	STO	SI	AA	
F8H	No relevant state information available SI=0	No action	No action				Wait or proceed current transfer
00H	Bus error during MST or selected slave modes	No action	0	1	0	X	Only the internal hardware is affected in the “master” or “addressed slave” modes. In all cases, the bus is released and I²C is switched to the “not addressed slave” mode. The “STO” flag is reset.

35 Serial Interface – SPI

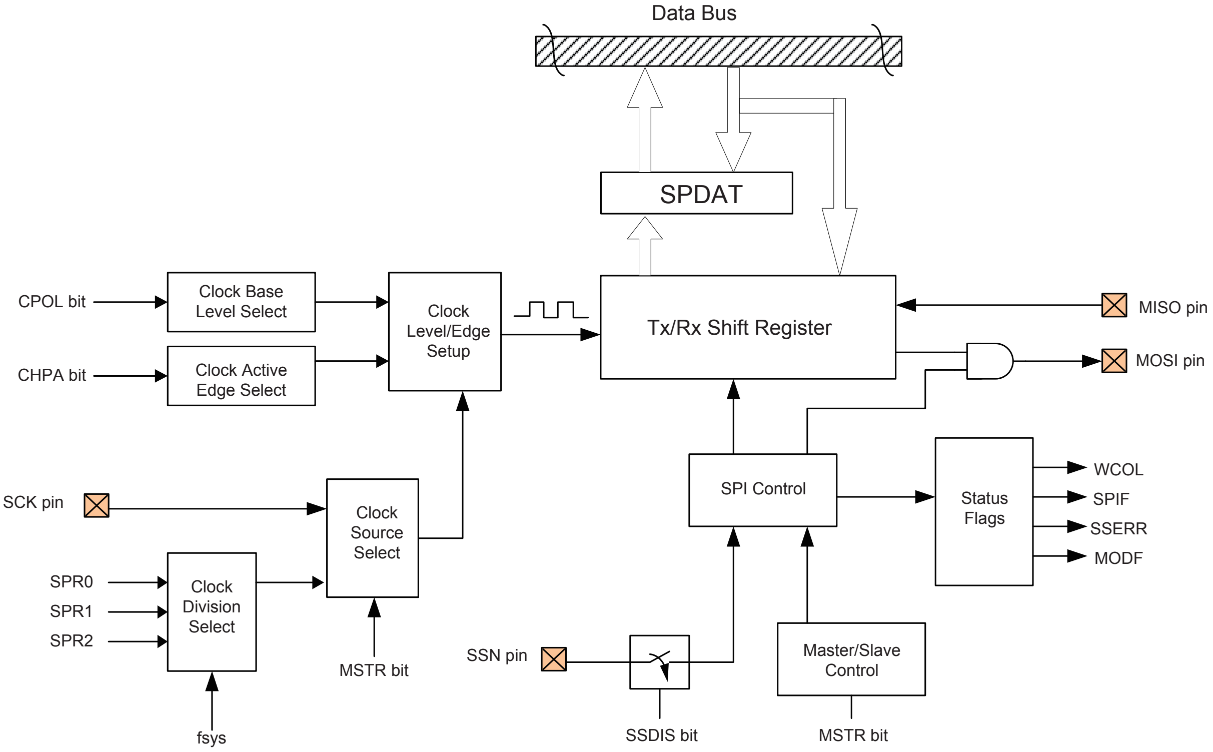
The SPI interface is often used to communicate with external peripheral devices such as sensors, Flash Memory or EEPROM memory devices etc. Originally developed by Motorola, the four line SPI interface is a synchronous serial data interface that has a relatively simple communication protocol simplifying the programming requirements when communicating with external hardware devices.

SPI Interface Operation

The SPI interface is a full duplex synchronous serial data link. The SPI interface is disabled or enabled using the SPEN bit in the SPCON register which configures the functionally shared pins as SPI pins and disables their logic I/O function. The SPI interface is a slave/master type, where the device can be either master or slave decided by the MSTR bit in the SPCON register. It is a four line interface with pin names, MOSI, MISO, SCK and SSN. Pins MOSI and MISO are the Serial Data Input and Serial Data Output lines, SCK is the Serial Clock line and SSN is the Slave Select line. Communication between devices connected to the SPI interface is carried out in a slave/master mode with all data transfer initiations being implemented by the master. The Master also controls the clock signal.



Single SPI Master and single Slave Connection



SPI Interface Block Diagram

SPI Features

The SPI function in the devices have the following features:

- Full duplex synchronous data transfer
- Three wire synchronous transfers
- Dual Master and Slave modes
- Seven SPI Master baud rates
- Slave Clock rate up to $f_{SYS}/8$
- Serial clock with programmable polarity and phase
- Master Mode fault error flag with MCU interrupt capability
- Write collision flag protection
- Data transmitted by MSB First, LSB Last mode

SPI Registers

There are three internal registers which control the overall operation of the SPI interface. These are the SPDAT data register, where the received data or transmitted data is stored and two control registers SPCON and SPSTA.

SPI Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
SPCON	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
SPSTA	SPIF	WCOL	SSERR	MODF	—	—	—	—
SPDAT	D7	D6	D5	D4	D3	D2	D1	D0

The SPDAT register is used to store the data being transmitted and received. Before the device writes data to the SPI bus, the actual data to be transmitted must be placed in the SPDAT register. After the data is received from the SPI bus, the device can read the data from the SPDAT register. Any transmission or reception of data from the SPI bus must be made via the SPDAT register.

SPDAT Register
SFR Address: E3h

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

There are also two control registers for the SPI interface, SPCON and SPSTA. Register SPCON is used to control the enable/disable function, to assign the clock Polarity/Edge types, to enable/disable the SPI function and to set the data transmission clock frequency. Register SPSTA is used to indicate the SPI operational status, such as data transferred complete flag, write collision flag, Synchronous Serial Slave Error flag and SPI mode Fault detection flag etc. When any one of the SPIF, WCOL, SSERR and MODF flags in the SPSTA register is set high, an SPI interrupt will occur if the SPI interrupt function is enabled.

SPCON Register
SFR Address: E8h

Bit	7	6	5	4	3	2	1	0
Name	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	1	0	1	0	0

Bit 7, 1, 0 **SPR2, SPR1, SPR0:** Master Clock Select
000: Reserved
001: $f_{SYS}/4$
010: $f_{SYS}/8$
011: $f_{SYS}/16$
100: $f_{SYS}/32$
101: $f_{SYS}/64$
110: $f_{SYS}/128$
111: Master clock not generated

Bit 6 **SPEN:** SPI enable or disable
0: Disable
1: Enable

When set high, the SPI interface internal circuits will be enabled. All the relevant functionally shared pins will be enabled to have SPI functions and their original logical I/O functions will be disabled. When cleared to zero, the SPI interface will be disabled, and all the functionally shared pins will have a logical I/O function.

Bit 5 **SSDIS:** SSN pin disable control
0: Enable
1: Disable, SSN pin floating

When this bit is cleared to zero, the “SSN” input is enabled in both Master and Slave modes. When set high, the “SSN” input is disabled in both Master and Slave modes. In the Slave mode, this bit has no effect if “CPHA”=0. When the bit is high, no “MODF” interrupt request will be generated.

Bit 4 **MSTR:** SPI Master or Slave
0: Slave
1: Master

Bit 3 **CPOL:** SPI Clock Polarity
0: SCK low when clock is inactive
1: SCK high when clock is inactive

The CPOL bit determines the SPI clock polarity when not active.

Bit 2 **CPHA:** SPI Active Clock Edge Select
0: Data sampled on first clock edge
1: Data sampled on second clock edge

The CPHA and CPOL bits are used to setup the way that the clock signal transmits data on the SPI bus. These two bits must be configured before a data transfer is executed.

SPSTA Register
SFR Address: E1h

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	SSERR	MODF	—	—	—	—
R/W	R/W	R/W	R/W	R/W	—	—	—	—
POR	0	0	0	0	—	—	—	—

Bit 7 **SPIF:** SPI Transmit/Receive Complete Flag
0: Data is being transferred
1: SPI data transmission completed

The SPIF bit is the Transmit/Receive Complete flag and is set high automatically when an SPI data transmission is completed, it must be cleared using the application program. The SPIF bit can be also cleared by hardware when the data transfer is in progress. It can also be used to generate an interrupt.

Bit 6 **WCOL:** SPI Write Collision Flag
0: No collision
1: Collision

The WCOL flag is used to detect if a data collision has occurred. If this bit is high it means that data has been attempted to be written to the SPDAT register during a data transfer operation. A SPI interrupt will occur if the SPI interrupt function is enabled. This writing operation will be ignored if data is being transferred. It must be cleared using the application program.

Bit 5 **SSERR:** Synchronous Serial Slave Error Flag
0: No error
1: Error

This bit is set by hardware when the SSN pin input is selected to disable the Slave device status while the receive sequence is incomplete. A SPI interrupt will occur if the SPI interrupt function is enabled. This bit will be cleared by disabling the SPI module, clearing the SPEN bit in the SPCON register.

Bit 4 **MODF:** SPI Master/Slave Mode Mismatch Flag
0: No Mismatch
1: Mismatch

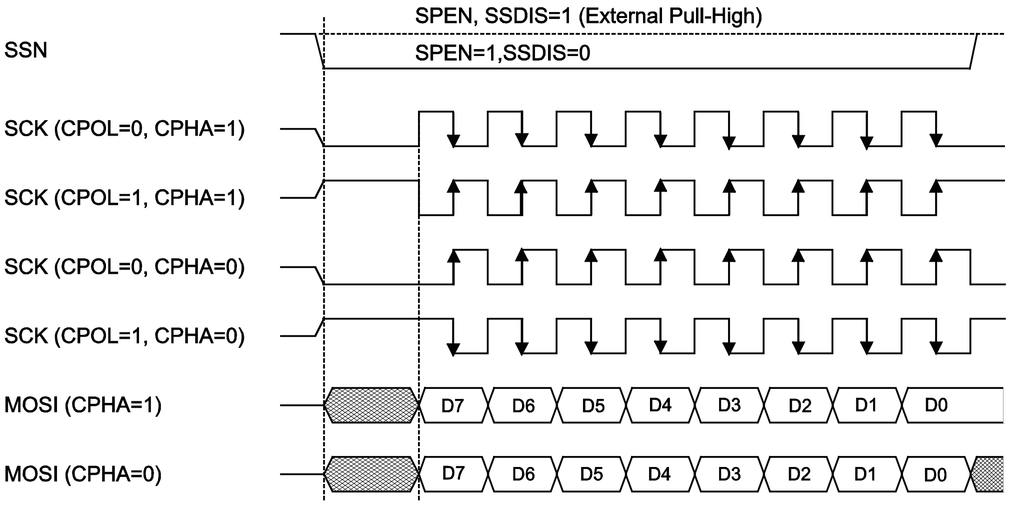
This bit is set by hardware when the Slave Select SSN pin level conflicts with actual Master/Slave mode of the SPI Master controller which is configured as a master while externally selected as a slave. A SPI interrupt will occur if the SPI interrupt function is enabled. It must be cleared using the application program.

Bit 3~0 Unimplemented, read as “0”

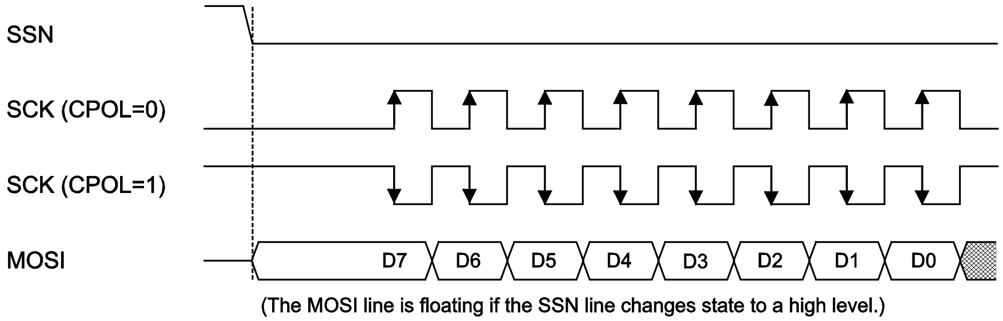
SPI Communication

The SPI interface is first enabled by setting the SPEN bit high. This enables the internal SPI circuitry and also enables all the SPI pins which also disabled all of the logical I/O functions. In the Master Mode, when data is written to the SPDAT register, transmission/reception will begin simultaneously. When the data transfer is complete, the SPIF flag will be set automatically, but must be cleared using the application program. In the Slave Mode, when the clock signal from the master has been received, any data in the SPDAT register will be transmitted and any data on the MISO pin will be shifted into the SPDAT register.

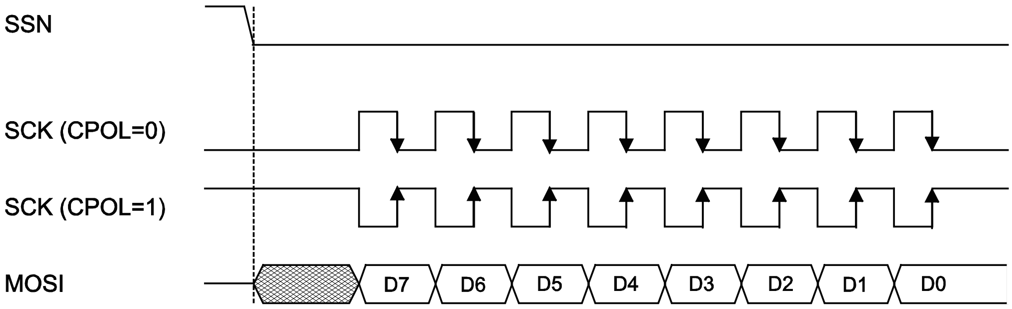
The master should output an SSN signal to enable the slave device before a clock signal is provided. The slave data to be transferred should be well prepared at the appropriate moment relative to the SSN signal depending upon the configurations of the CPOL bit and CPHA bit. The accompanying timing diagram shows the relationship between the slave data and SSN signal for various configurations of the CPOL and CPHA bits.



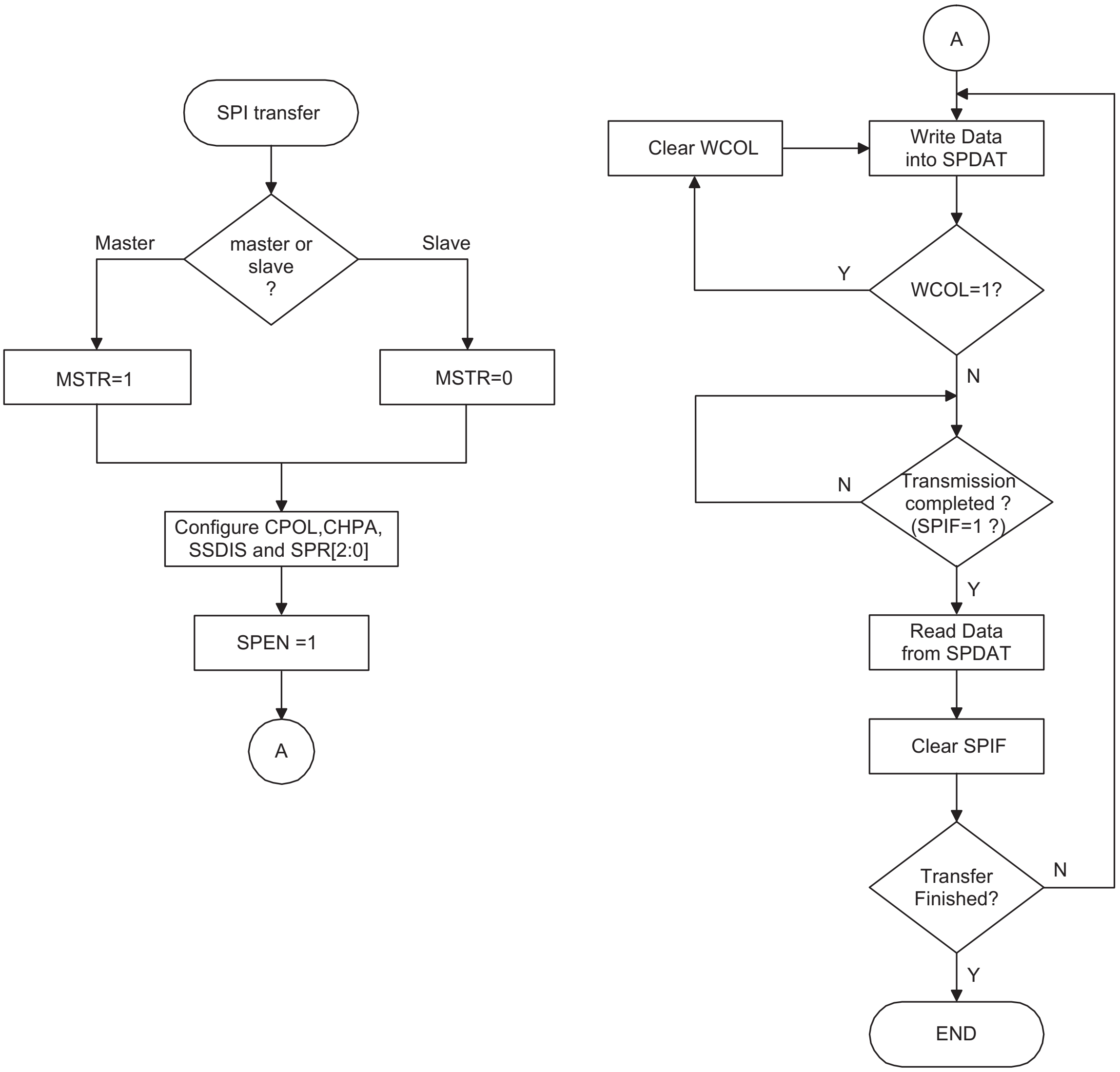
SPI Master Mode Timing


SSN
SCK (CPOL=0)
SCK (CPOL=1)
MOSI
D7
D6
D5
D4
D3
D2
D1
D0
(The MOSI line is floating if the SSN line changes state to a high level.)

SPI Slave Mode Timing – CPHA=0



SPI Slave Mode Timing – CPHA=1



SPI Transfer Control Flowchart

36 UART Serial Interfaces – UART0 and UART1

Two fully integrated serial communications UART interfaces, namely UART0 and UART1, enable the communication with external devices that contain a similar serial interface. The UART1 is only available on the HT85F2270/2280. Although what is known as a UART function essentially provides only asynchronous data transfer operations, UART0 also provides extended synchronous data transfer operations.

UART Overview

UART0 provides a flexible full-duplex synchronous/asynchronous receiver/transmitter and has four operating modes while UART1 provides a flexible full-duplex asynchronous receiver/transmitter with two operating modes. Both of them have programmable Baud Rates. The UART functions have many features and can transmit and receive data serially by transferring a frame of data with eight or nine data bits per transmission. The UART functions possesses their own internal interrupt which can be used to indicate when a data reception operation has occurred or when a data transmission operation has terminated.

UART0 Features

The integrated UART0 function contains the following features:

- Full-duplex, synchronous and asynchronous communication
- Four operating modes
- 8 or 9 bits character length
- Programmable Baud rate generator
- Separately enabled transmitter and receiver
- 2-byte Deep FIFO Receive Data Buffer
- Transmission and reception of interrupts
- Fully compatible with the standard 8051 serial channel

UART1 Features

The integrated UART1 function contains the following features:

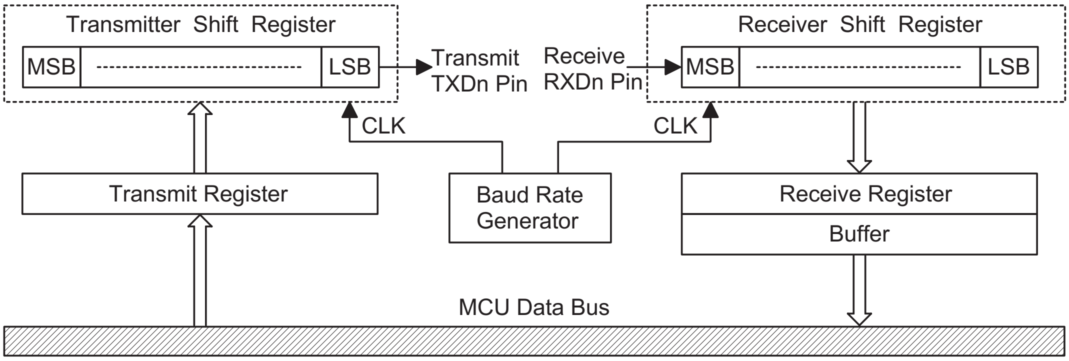
- Full-duplex, asynchronous communication
- Two operating modes
- 8 or 9 bits character length
- Programmable Baud rate generator
- Separately enabled transmitter and receiver
- 2-byte Deep FIFO Receive Data Buffer
- Transmission and reception of interrupts
- Fully compatible with the standard 8051 serial channel

Basic UART Data Transfer Scheme

The block diagram shows the overall UART data transfer structure arrangement. For data transmission, the actual data to be transmitted from the MCU is first transferred to the transmitted register by the application program. The data will then be transferred to the Transmit Shift Register from where it will be shifted out, LSB first, onto the transmitter pin, TXDn, at a rate controlled by the Baud Rate Generator. Only the transmitted register is mapped onto the MCU Data Memory, the Transmit Shift Register is not mapped and is therefore inaccessible to the application program.

For data reception, data to be received by the UART is accepted on the external receive pin, RXDn, from where it is shifted in, LSB first, to the Receiver Shift Register at a rate controlled by the Baud Rate Generator. When the shift register is full, the data will then be transferred from the shift register to the internal received register, where it is buffered and can be manipulated by the application program. Only the received register is mapped onto the MCU Data Memory, the Receiver Shift Register is not mapped and is therefore inaccessible to the application program.

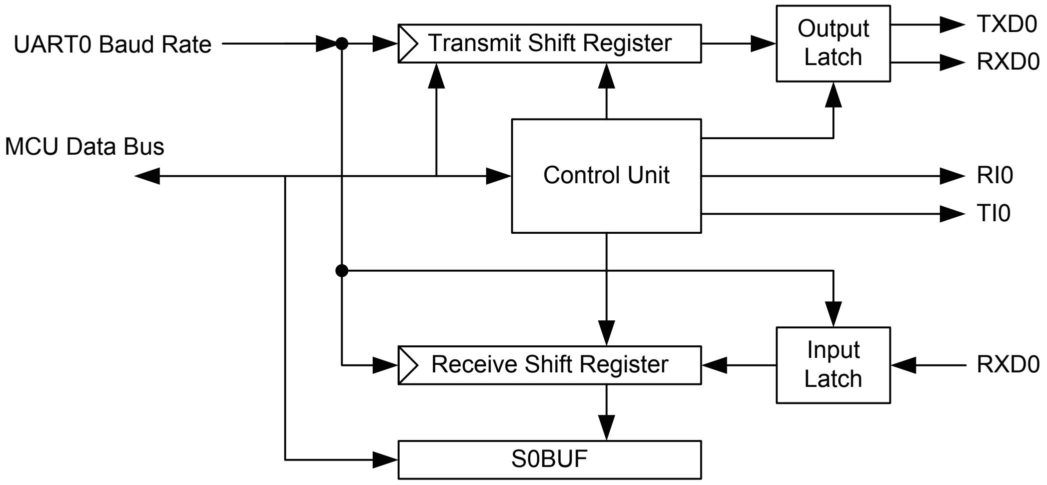
It should be noted that the actual register for data transmission and reception, only exists as a single shared register in the Data Memory, known as the S0BUF register in the UART0 or the S1BUF register in the UART1, and is used for both data transmission and data reception.



Basic UART Data Transfer Diagram

UART0 Operating Description

This section provides a more detailed description of the UART0 structure and operation. The following shows the overall UART0 block diagram.



UART 0 Block Diagram

UART0 External Pin Interfacing

To communicate with an external serial interface, the UART0 has two external pins known as TXD0 and RXD0. The UART0 provides four operating modes which can be categorised into two transmitter/receiver methods, so called Synchronous and Asynchronous. In Synchronous communication, the MCU must be the master device and the TXD0 pin is used to provide the shift clock while the RXD0 pin is used as the data transmitter/receiver pin. In Asynchronous communication, which does require a clock signal, the TXD0 pin is used to transmit data while the RXD0 pin is used as the data receive pin. The TXD0 and RXD0 pins are pin shared with I/O pins. When the UART0 function is disabled, controlled by the REN0 bit in the S0CON register, these two pins can be used as general purpose I/O pins.

UART0 Register Description

There are several control registers associated with the UART0 function. The S0CON register controls the overall function of the UART0, while the SBRCON, SPPRE, S0RELL and S0RELH registers control the Baud rate. The actual data to be transmitted and received on the serial interface is managed through the S0BUF data register. The SMOD bit in the PCON register is used to double the baud rate clock.

UART0 Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
S0RELL	S0REL.7	S0REL.6	S0REL.5	S0REL.4	S0REL.3	S0REL.2	S0REL.1	S0REL.0
S0RELH	—	—	—	—	—	—	S0REL.9	S0REL.8
S0BUF	D7	D6	D5	D4	D3	D2	D1	D0
SPPRE	—	—	—	—	S1PRE1	S1PRE0	S0PRE1	S0PRE0
SBRCON	BD	BD1	—	—	—	—	—	—
PCON	SMOD	—	—	—	—	GF0	PD	IDL

S0BUF Register – UART0 Data register
SFR Address: 99h

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 UART0 data buffer

S0CON Register – UART0 Control register
SFR Address: 98h

Bit	7	6	5	4	3	2	1	0
Name	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **SM0, SM1:** UART0 Operating mode select bits
00: Mode 0
01: Mode 1
10: Mode 2
11: Mode 3

The following table illustrates the corresponding mode descriptions and baud rates. In mode1 and mode 3, the variable baud rate is dependent on the system clock, the baud rate clock source and the prescaler selections. Operating mode details are described elsewhere.

Mode	Mode Name	Baud Rate	Synchronisation
Mode 0	8-bit shift register	$f_{SYS}/12$	Synchronous
Mode 1	8-bit UART	Variable	Asynchronous
Mode 2	9-bit UART	SP0CLK/32 or SP0CLK/64	Asynchronous
Mode 3	9-bit UART	Variable	Asynchronous

Note that the SP0CLK is described in the UART0 Baud Rate Setup section.

Bit 5 **SM20:** Multiprocessor communication enable control
0: Disable
1: Enable

Refer to the UART0 Multiprocessor Communication section for details.

Bit 4 **REN0:** UART0 serial data reception enable control
0: Disable
1: Enable

Bit 3 **TB80:** Ninth Transmit bit assignment
0: Low
1: High

This bit is only available in Mode 2 and Mode 3. It is not effective in Mode 0 and Mode 1. The bit is assigned using the application program.

Bit 2 **RB80:** Ninth Receive bit assignment
0: Low
1: High

This bit is used to assign the level of the ninth bit in Mode 2 and Mode 3. In mode 1, if the SM20 bit is zero, the RB80 bit is assigned as the level of the received stop bit. It is not available in Mode 0.

Bit 1 **TI0:** UART0 transmit interrupt flag
0: No interrupt request
1: Interrupt request

This bit must be cleared using the application program.

Bit 0 **RI0:** UART0 receive interrupt flag
0: No interrupt request
1: Interrupt request

This bit must be cleared using the application program.

S0RELL Register – UART0 Reload Low Register
SFR Address: AAh

Bit	7	6	5	4	3	2	1	0
Name	S0REL.7	S0REL.6	S0REL.5	S0REL.4	S0REL.3	S0REL.2	S0REL.1	S0REL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	0	1	1	0	0	1

S0RELH Register – UART0 Reload High Register
SFR Address: BAh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	S0REL.9	S0REL.8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	1	1

The UART0 Reload registers, S0RELL and S0RELH, are used to setup the UART0 baud rate generation. The UART0 baud rate setup range is 10-bit wide, consisting of 8 bits in S0RELL and 2 bits in S0RELH.

SPPRE Register – UART Clock Prescaler Register
SFR Address: A5h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	S1PRE1	S1PRE0	S0PRE1	S0PRE0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	1	1	1

Bit 7~4 Unimplemented, read as “0”

Bit 3~2 **S1PRE1, S1PRE0:** UART 1 Reload Counter Clock Select
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

Bit 1~0 **S0PRE1, S0PRE0:** UART 0 Reload Counter Clock Select
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

SBRCON Register
SFR Address: DCh

Bit	7	6	5	4	3	2	1	0
Name	BD	BD1	—	—	—	—	—	—
R/W	R/W	R/W	—	—	—	—	—	—
POR	0	0	—	—	—	—	—	—

Bit 7 **BD:** UART0 Baud rate select for mode 1 and mode 3
0: Timer 1 overflow baud rate generator
1: SRELL and SRELH register controlled baud rate generator
The SRELL and SRELH registers combine to form a 10-bit reload register pair for the Baud rate generator.

Bit 6 **BD1:** UART1 internal Baud rate generator enable control (only available on HT85F2280/2270)
0: Disable
1: Enable

Bit 5~0 Unimplemented, read as "0"

PCON Register
SFR Address: 87h

Bit	7	6	5	4	3	2	1	0
Name	SMOD	—	—	—	—	GF0	PD	IDL
R/W	R/W	—	—	—	R	R/W	R/W	R/W
POR	0	—	—	—	1	0	0	0

Bit 7 **SMOD:** UART0 double baud rate select
0: Not double
1: Double

Bit 6~3 Unimplemented

Bit 2 **GF0:** General Purpose Flag

Bit 1 **PD:** Power-Down Mode control bit
Described elsewhere

Bit 0 **IDL:** IDLE Mode control bit
Described elsewhere

UART0 Operating Modes

UART0 provides four operation modes, selected by the SM1 and SM0 bits in the S0CON register. There are one synchronous and three asynchronous modes, offering different baud rates and functional options. The following table illustrates the different operational mode list. When a transmit/receive data transfer operation has completed, a transmit/receive interrupt will take place and the interrupt request bit, TI0 or RI0, will be set high.

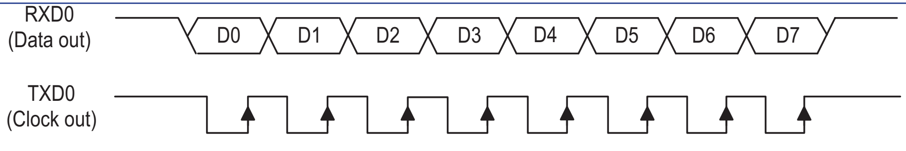
UART0 Operating Modes

Mode	Mode Name	Baud Rate	Synchronisation	Start/Stop Bits
Mode 0	8-bit shift register	$f_{SYS}/12$	Synchronous	None
Mode 1	8-bit UART	Variable	Asynchronous	1 Start, 1 Stop
Mode 2	9-bit UART	SP0CLK/32 or SP0CLK/64	Asynchronous	1 Start, 1 Stop
Mode 3	9-bit UART	Variable	Asynchronous	1 Start, 1 Stop

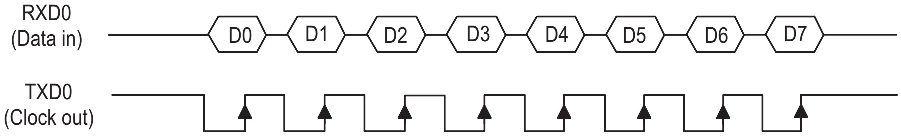
Mode 0

Mode 0 is an integrated half-duplex synchronous serial communication interface. The 8 bits of data are communicated via the RXD0 pin while the TXD0 pin provides the shift clock for this communication. The data will then be transferred to the Transmit Shift Register from where it will be shifted out, LSB first, onto the RXD0 pin at a fixed Baud rate of $f_{SYS}/12$. The Transmission is started by writing a data into the S0BUF register.

Data to be received by the UART is accepted on the external RXD0 pin, from where it is shifted in, LSB first, to the Receiver Shift Register at a fixed Baud rate of $f_{SYS}/12$. The reception is started by setting the REN0 bit in the S0CON register to "1". When the shift register is full, the data will then be transferred from the shift register to the internal S0BUF register, where it is buffered and can be manipulated by the application program.



Data Transmit Timing Diagram



Data Recerive Timing Diagram

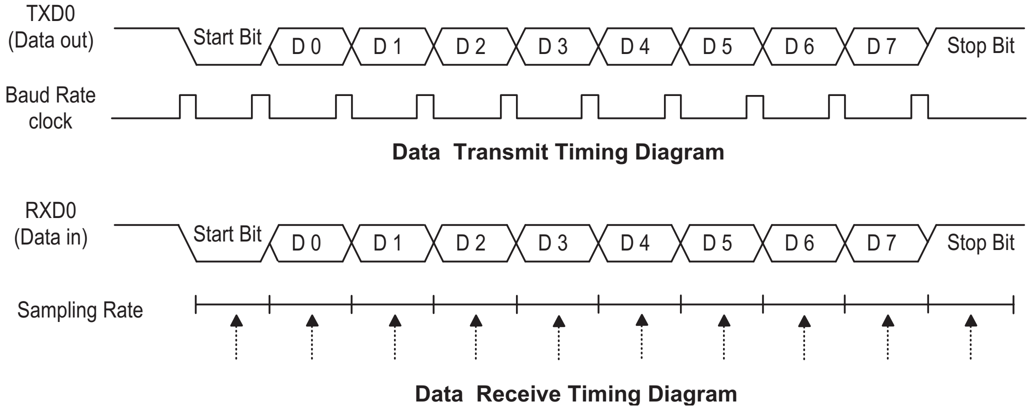
UART0 Mode 0 Timing Diagram

Mode 1

Mode 1 is an integrated full-duplex asynchronous serial communication interface with a variable baud rate. The 8 bits of data are received via the RXD0 pin while the TXD0 pin is the data transmit pin for this communication mode. For data reception, data received on the RXD0 pin will be shifted into the Receive Shift Register by the Baud rate generator. For data transmission, data in the Transmit Shift Register will be shifted out onto the TXD0 pin by the Baud rate generator. The Baud rate frequency is selected by the BD bit in the SBRCON register to be either sourced from the Timer1 overflow or to be setup by the S0RELL/S0RELH registers. In addition, the Baud rate can be doubled using the SMOD bit in the PCON register.

Data transmission is started by writing to the S0BUF register. The TXD0 pin outputs the serial data. The first bit transmitted is a start bit, always “0”, then 8 bits of data, after which a stop bit, always “1”, is transmitted.

Data to be received by the UART is accepted on the external RXD0 pin. When reception starts, the UART0 synchronises with the falling edge detected by the RXD0 pin. Input data is available after one byte of data is complete in the S0BUF register and the value of the STOP bit is available as the RB80 flag in the S0CON register. During the reception process, the S0BUF data and the RB80 bit will remain unchanged until the process is complete.



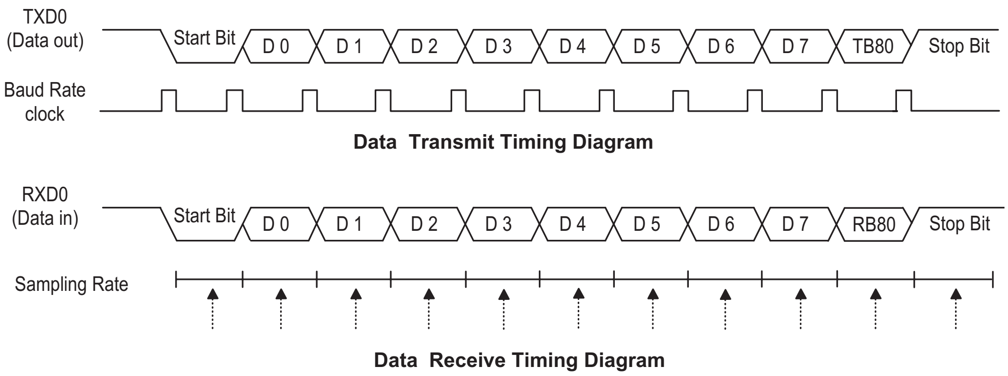
UART0 Mode 1 Timing Diagram

Mode 2

Mode 2 is an integrated full-duplex asynchronous serial communication interface. The 9 bits of data are received via the RXD0 pin while the TXD0 pin is the data transmit pin for this communication mode. For data reception, data received on the RXD0 pin will be shifted into the Receive Shift Register by the Baud rate generator. For data transmission, data in the Transmit Shift Register will be will be shifted out onto the TXD0 pin by the Baud rate generator. The Baud rate is fixed at a value of SP0CLK/32 or SP0CLK/64, depending on the setting of the SMOD bit in the PCON register.

Data transmission is started by writing to the S0BUF register. The TXD0 pin outputs the serial data. The first bit transmitted is a start bit, always "0", then 9 bits of data where the 9th bit is taken from the TB80 bit of the S0CON register, after which a stop bit, always "1", is transmitted.

Data to be received by the UART0 is accepted on the external RXD0 pin. When reception starts, the UART0 synchronises with the falling edge detected by the RXD0 pin. Input data is available after one byte of data is complete in the S0BUF register, and the 9th bit is available as the RB80 bit in the S0CON register. During the reception process, the S0BUF data and the RB80 bit will remain unchanged until the process is complete.



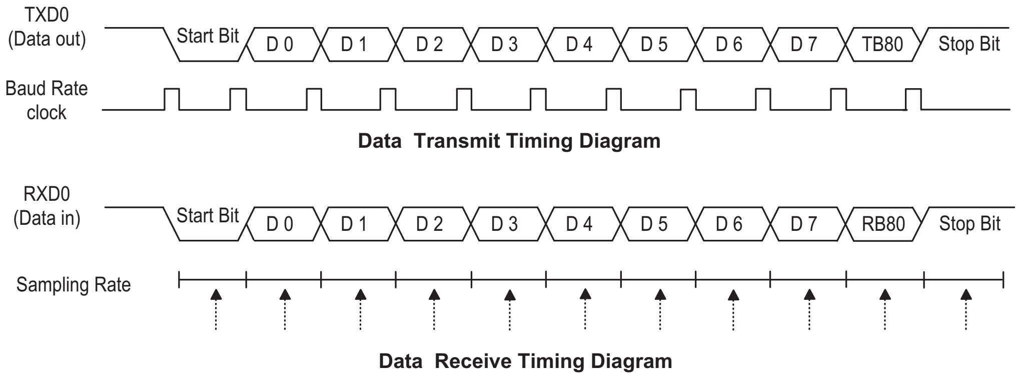
UART0 Mode 2 Timing Diagram

Mode 3

The only difference between Mode 2 and Mode 3 is that the internal Baud rate is variable in Mode 3, whereas it is fixed in Mode 2. Mode 3 is an integrated full-duplex asynchronous serial communication interface. The 9 bits of data are received via the RXD0 pin while the TXD0 pin is the data transmit pin for this communication mode. For data reception, data received on the RXD0 pin will be shifted into the Receive Shift Register by the Baud rate generator. For data transmission, data in the Transmit Shift Register will be will be shifted out onto the TXD0 pin by the Baud rate generator. The Baud rate frequency is selected by the BD bit in the SBRCON register to be either sourced from the Timer1 overflow or to be setup by the S0RELL/S0RELH registers. In addition, the Baud rate can be doubled using the SMOD bit in the PCON register.

Data transmission is started by writing to the S0BUF register. The TXD0 pin outputs the serial data. The first bit transmitted is a start bit, always "0", then 9 bits of data where the 9th bit is taken from the TB80 bit of the S0CON register, after which a stop bit, always "1", is transmitted.

Data to be received by the UART is accepted on the external RXD0 pin. When reception starts, the UART0 synchronizes with the falling edge detected by the RXD0 pin. Input data is available after one byte of data is complete in the S0BUF register, and the 9th bit is available as the RB80 bit in the S0CON register. During the reception process, the S0BUF data and the RB80 bit will remain unchanged until the process is complete.



UART0 Mode 3 Timing Diagram

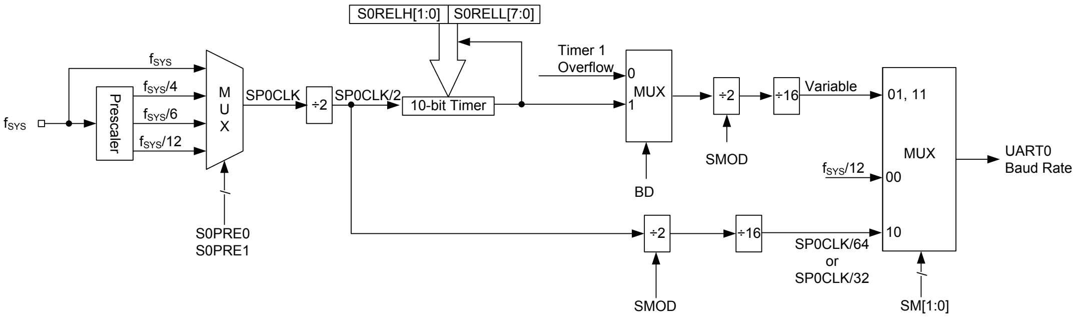
UART0 Multiprocessor Communication

As UART0 can receive 9 bits in Modes 2 and 3, it can be used for multiprocessor communication. When the SM20 bit in the S0CON register is set, the received interrupt is generated only when the 9th received bit, the RB80 bit in the S0CON register, is high. Otherwise, no interrupt is generated upon reception.

To utilise this feature for multiprocessor communication, the slave processors have their SM20 bit set high. The master processor transmits the slave's address, with the 9th bit set high, generating a reception interrupt in all of the slaves. The slave processors' software compares the received byte with their network address. If there is a match, the addressed slave clears its SM20 flag and the rest of the message is transmitted from the master with the 9th bit cleared to zero. The other slaves keep their SM20 set high so that they ignore the rest of the message sent by the master. In this way, there are reduced program overheads to distinguish the target slave MCU.

UART0 Baud Rate Setup

The UART0 operating Modes1 and 3, have a variable baud rate setup using the UART0 Baud rate generator. The clock source can be selected to be either the Timer 1 overflow or the system clock, decided by the BD bit in the SBRCON register. The baud rate generator can be controlled by the S0RELH and S0RELL registers and the clock is the output of the prescaler, defined by the S0PRE0 and S0PRE1 bits. Operating Mode 0 has a fixed baud rate of $f_{SYS}/12$ and operating Mode 2, has two baud rates, SP0CLK/64 and SP0CLK/32, selected by the SMOD bit in the PCON register.



UART0 Baud Rate Generator

The Variable baud rate, which is provided for Mode 1 and Mode 3, can be derived using the following two equations, depending upon the BD bit condition.

BD=0 – Timer1 overflow clock source.

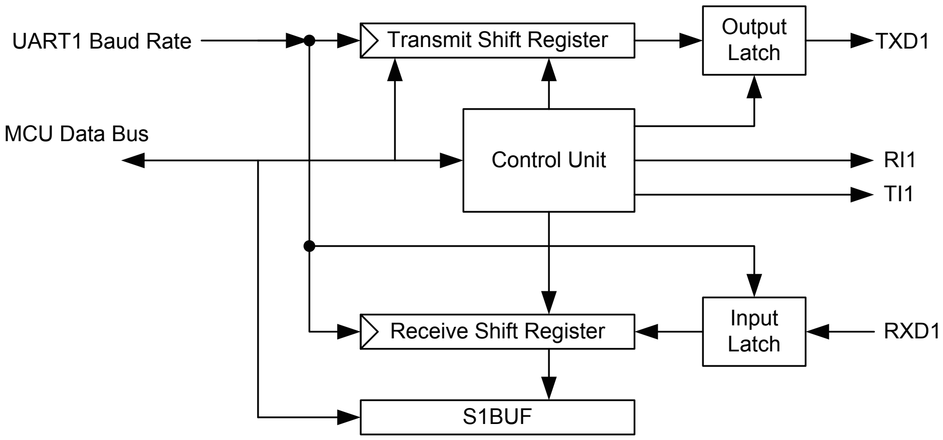
$$Baud\_Rate = \frac{2^{SMOD}}{32} * (Timer1\_Overflow\_Rate)$$

BD=1 – Register select clock Source.

$$Baud\_Rate = \frac{2^{SMOD}}{64 * (2^{10} - S0REL[9:0])} * (Freq\_of\_SP0CLK)$$

UART1 Operating Description

This section provides a more detailed description of the UART1 structure and operation. The following shows the overall UART1 block diagram.



UART1 Block Diagram

UART1 External Pin Interfacing

To communicate with an external serial interface, the internal UART1 each has two external pins known as TXD1 and RXD1. The UART1 provides two operating modes both of which use Asynchronous communication. The TXD1 pin is used to transmit data while the RXD1 pin is used as the data reception pin. The TXD1 and RXD1 pins are pin shared with I/O pins. When the UART1 function is disabled, controlled by the REN1 bit in the S1CON register, these two pins can be used as general purpose I/O pins.

UART1 Register Description

There are several control registers associated with the UART1 function. The S1CON register controls the overall function of the UART1, while the SPPRE, S1RELL and S1RELH registers control the Baud rate. The actual data to be transmitted and received on the serial interface is managed through the S1BUF data register.

UART1 Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
S1CON	SM	—	SM21	REN1	TB81	RB81	TI1	RI1
S1RELL	S1REL.7	S1REL.6	S1REL.5	S1REL.4	S1REL.3	S1REL.2	S1REL.1	S1REL.0
S1RELH	—	—	—	—	—	—	S1REL.9	S1REL.8
S1BUF	D7	D6	D5	D4	D3	D2	D1	D0
SPPRE	—	—	—	—	S1PRE1	S1PRE0	S0PRE1	S0PRE0

S1BUF Register – UART1 Data register
SFR Address: 9Ch

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 UART1 data buffer

S1CON Register – UART1 Control register
SFR Address: 9Bh

Bit	7	6	5	4	3	2	1	0
Name	SM	—	SM21	REN1	TB81	RB81	TI1	RI1
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	—	0	0	0	0	0	0

Bit 7 **SM:** UART1 Operating mode select bit
0: Mode A
1: Mode B
The following table illustrates the corresponding mode descriptions and baud rates. Operating mode details are described elsewhere.

SM bit	Mode	Mode Name	Baud Rate	Synchronisation
0	Mode A	9-bit UART	Variable	Asynchronous
1	Mode B	8-bit UART	Variable	Asynchronous

Bit 6 Unimplemented, read as “0”

Bit 5 **SM21:** Multiprocessor communication enable control
0: Disable
1: Enable
Refer to the UART1 Multiprocessor Communication section for details.

Bit 4 **REN1:** UART1 serial data reception enable control
0: Disable
1: Enable

Bit 3 **TB81:** Ninth Transmit bit assignment
0: Low
1: High
This bit is only available in Mode A. It is not effective in Mode B. The bit is assigned using the application program.

Bit 2 **RB81:** Ninth Receive bit assignment
0: Low
1: High
This bit is used to assign the level of the ninth bit in Mode A. In mode A, if the SM21 bit is zero, the RB81 bit is assigned as the level of the received stop bit. It is not available in Mode B.

Bit 1 **TI1:** UART1 transmit interrupt flag
0: No interrupt request
1: Interrupt request
This bit must be cleared using the application program.

Bit 0 **RI1:** UART1 receive interrupt flag
0: No interrupt request
1: Interrupt request
This bit must be cleared using the application program.

S1RELL Register – UART1 Reload Low Register
SFR Address: 9Dh

Bit	7	6	5	4	3	2	1	0
Name	S1REL.7	S1REL.6	S1REL.5	S1REL.4	S1REL.3	S1REL.2	S1REL.1	S1REL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

S1RELH Register – UART1 Reload High Register
SFR Address: BBh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	S1REL.9	S1REL.8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	1	1

The UART1 Reload registers, S1RELL and S1RELH, are used to setup the UART1 baud rate generation. The UART1 baud rate setup range is 10-bit wide, consisting of 8 bits in S1RELL and 2 bits in S1RELH.

SPPRE Register – UART Clock Prescaler Register
SFR Address: A5h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	S1PRE1	S1PRE0	S0PRE1	S0PRE0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	1	1	1

Bit 7~4 Unimplemented, read as “0”

Bit 3~2 **S1PRE1, S1PRE0:** UART 1 Reload Counter Clock Select
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

Bit 1~0 **S0PRE1, S0PRE0:** UART 0 Reload Counter Clock Select
00: $f_{SYS}/12$
01: $f_{SYS}/6$
10: $f_{SYS}/4$
11: f_{SYS}

UART1 Operating Modes

UART1 provides two operational modes, selected by the SM bit in the S1CON register. There are two asynchronous modes, offering different baud rates and functional options. The following table illustrates the different operational mode list. When the transmit/receive data process is complete, the transmit/receive interrupt will take place and the interrupt request bit, TI1 or RI1, will be set high.

UART1 Operating Modes

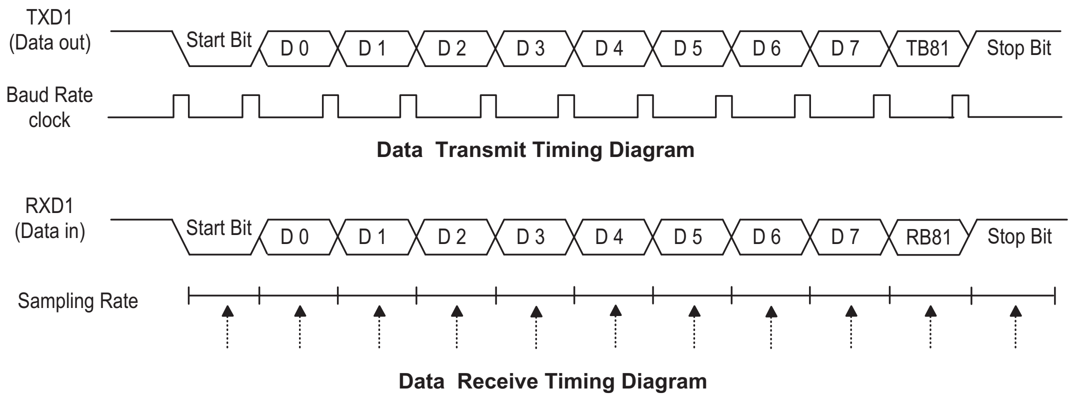
SM bit	Mode	Mode Name	Baud Rate	Synchronisation	Start/Stop Bits
1	Mode B	8-bit UART	Variable	Asynchronous	1 Start, 1 Stop
0	Mode A	9-bit UART	Variable	Asynchronous	1 Start, 1 Stop

Mode A

Mode A is an integrated full-duplex asynchronous serial communication interface. The 9 bits of data are received via the RXD1 pin while the TXD1 pin provides the data transmit for this communication. The data will then be transferred to the Transmit Shift Register from where it will be shifted out onto the TXD1 pin by the UART1 Baud rate generator. The internal Baud rate generator is enabled/disabled by the BD1 bit of the SBRCON register. The S1RELL/S1RELH registers must be used to setup the Baud rate generator.

Data transmission is started by writing to the S1BUF register. The TXD1 pin outputs the serial data. The first bit transmitted is a start bit, always “0”, and then 9 bits of data where the 9th bit is taken from the TB81 bit of the S1CON register, after which a stop bit, always “1”, is transmitted.

Data to be received by the UART1 is accepted on the external RXD1 pin. When reception starts, the UART1 synchronises with the falling edge detected by the RXD1 pin. Input data is available after one byte of data is complete in the S1BUF register, and the ninth bit is available as the RB81 bit in the S1CON register. During the reception process, the S1BUF data and the RB81 bit will remain unchanged until the process is complete.



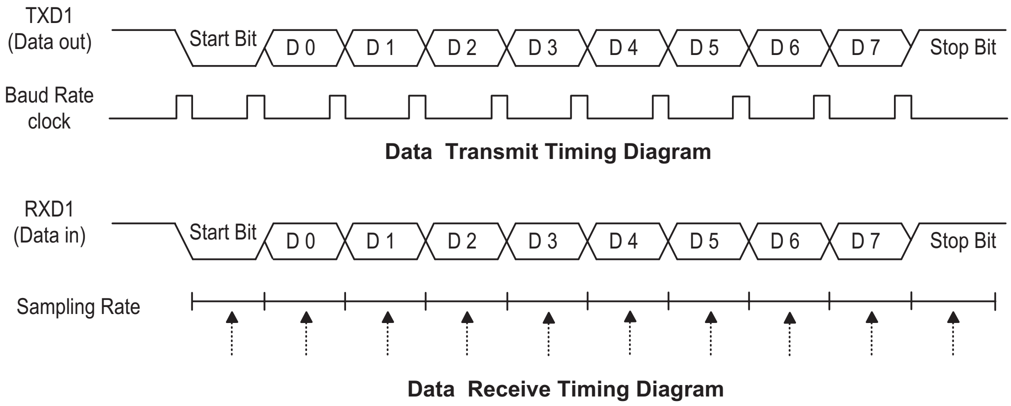
UART1 Mode A Timing Diagram

Mode B

Mode B is an integrated full-duplex asynchronous serial communication interface. The 8 bits of data are received via the RXD1 pin while the TXD1 pin provides the data transmit for this communication. The data will then be transferred to the Transmit Shift Register from where it will be shifted out onto the TXD1 pin by the UART1 Baud rate generator. The internal Baud rate generator is enabled/disabled by the BD1 bit of the SBRCON register. The S1RELL/S1RELH registers must be used to setup the Baud rate generator.

Data transmission is started by writing data to the S1BUF register. The TXD1 pin outputs data. The first bit transmitted is a start bit, always "0", then 8 bits of data, after which a stop bit, always "1", is transmitted.

Data to be received by the UART1 is accepted on the external RXD1 pin. When reception starts, UART1 synchronises with the falling edge detected by the RXD1 pin. Input data is available after one byte of data is complete in the S1BUF register and the value of the STOP bit is available as the RB81 flag in the S1CON register. During the reception process, the S1BUF data and the RB81 bit will remain unchanged until the process is complete.



UART1 Mode B Timing Diagram

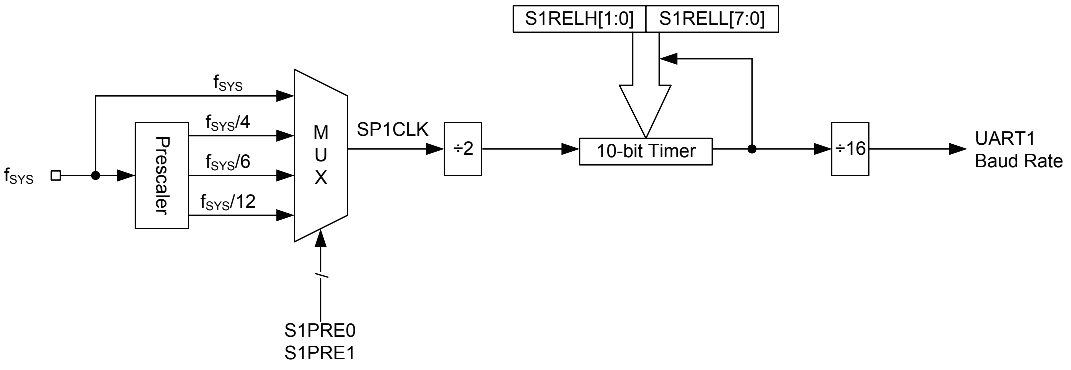
UART1 Multiprocessor Communication

As UART1 can receive 9 bits in Mode A, it can be used for multiprocessor communication. When the SM21 bit in the S1CON register is set, the receive interrupt is generated only when the 9th received bit, the RB81 bit in the S1CON register, is high. Otherwise, no interrupt is generated upon reception.

To utilise this feature for multiprocessor communication, the slave processors have their SM21 bit set to high. The master processor transmits the slave's address, with the 9th bit set high, generating a reception interrupt in all of the slaves. The slave processors' software compares the received byte with their network address. If there is a match, the addressed slave clears its SM21 flag and the rest of the message is transmitted from the master with the 9th bit set to low. The other slaves keep their SM21 set high so that they ignore the rest of the message sent by the master. In this way, there are reduced program overheads to distinguish the target slave MCU.

UART1 Baud Rate Setup

The UART1 operating Modes A and B, have a variable baud rate setup using the UART1 Baud rate generator. The clock source is sourced from the system clock. The baud rate generator can be controlled using the S1RELH and S1RELL registers and the clock is selected using the S1PRE0 and S1PRE1 bits.



UART1 Baud Rate Generator

The Variable baud rate can be derived using the following equation:

$$Baud\_Rate = \frac{1}{32 * (2^{10} - S1REL[9:0])} * (Freq\_of\_SP1CLK)$$

37 Instruction Set

Introduction

All instructions are binary code compatible and perform the same functions as they do within the industry standard 8051. The following tables give a summary of instruction cycles of the HT85XXX microcontroller core.

The following two tables contain notes on mnemonics used in Instruction set.

Notes on Data Addressing Modes

Symbol	Description
Rn	Working register R0~R7.
direct	One of 128 internal RAM locations or any Special Function Register.
@Ri	Indirect internal or external RAM location addressed by register R0 or R1.
#data	8-bit constant included in instruction (immediate operand).
#data 16	16-bit constant included as bytes 2 and 3 of instruction (immediate operand).
bit	One of 128 software flags located in internal RAM, or any flag of bit-addressable Special Function Registers, including I/O pins and status word.
A	Accumulator.

Notes on Program Addressing Modes

Symbol	Description
addr16	Destination address for LCALL or LJMP, can be anywhere within the 64-Kbyte page of program memory address space.
addr11	Destination address for ACALL or AJMP, within the same 2-Kbyte page of program memory as the first byte of the following instruction.
rel	SJMP and all conditional jumps include an 8-bit offset byte. Its range is +127/-128 bytes relative to the first byte of the following instruction.

The following tables show instruction hexadecimal codes, number of bytes and machine cycles that each instruction takes to be executed. Note the number of cycles is given for no program memory wait states.

Arithmetic Operations

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to Accumulator	0X28-0X2F	1	1
ADD A,direct	Add directly addressed data to Accumulator	0X25	2	2
ADD A,@Ri	Add indirectly addressed data to Accumulator	0X26-0X27	1	2
ADD A,#data	Add immediate data to Accumulator	0X24	2	2
ADDC A,Rn	Add register to Accumulator with carry flag	0X38-0X3F	1	1
ADDC A,direct	Add directly addressed data to Accumulator with carry flag	0X35	2	2
ADDC A,@Ri	Add indirectly addresses data to Accumulator with carry flag	0X36-0X37	1	2
ADDC A,#data	Add immediate data to Accumulator with carry flag	0X34	2	2
SUBB A,Rn	Subtract register from Accumulator with borrow	0X98-0X9F	1	1
SUBB A,direct	Subtract directly addressed data from Accumulator with borrow	0X95	2	2
SUBB A,@Ri	Subtract indirectly addressed data from Accumulator with borrow	0X96-0X97	1	2
SUBB A,#data	Subtract immediate data from Accumulator with borrow	0X94	2	2
INC A	Increment Accumulator	0X04	1	1
INC Rn	Increment register	0X08-0X0F	1	1
INC direct	Increment directly addressed location	0X05	2	3
INC @Ri	Increment indirectly addressed location	0X06-0X07	1	3
INC DPTR	Increment data pointer	0XA3	1	1
DEC A	Decrement Accumulator	0X14	1	1
DEC Rn	Decrement register	0X18-0X1F	1	1
DEC direct	Decrement directly addressed location.	0X15	2	3
DEC @Ri	Decrement indirectly addressed location	0X16-0X17	1	3
MUL AB	Multiply A and B	0XA4	1	4
DIV AB	Divide A by B	0X84	1	4
DA A	Decimal adjust Accumulator	0XD4	1	1

Logic Operations

Mnemonic	Description	Code	Bytes	Cycles
ANL A,Rn	AND register to Accumulator	0X58-0X5F	1	1
ANL A,direct	AND directly addressed data to Accumulator	0X55	2	2
ANL A,@Ri	AND indirectly addressed data to Accumulator	0X56-0X57	1	2
ANL A,#data	AND immediate data to Accumulator	0X54	2	2
ANL direct,A	AND Accumulator to directly addressed location	0X52	2	3
ANL direct,#data	AND immediate data to directly addressed location	0X53	3	4
ORL A,Rn	OR register to Accumulator	0X48-0X4F	1	1
ORL A,direct	OR directly addressed data to Accumulator	0X45	2	2
ORL A,@Ri	OR indirectly addressed data to Accumulator	0X46-0X47	1	2
ORL A,#data	OR immediate data to Accumulator	0X44	2	2
ORL direct,A	OR Accumulator to directly addressed location	0X42	2	3
ORL direct,#data	OR immediate data to directly addressed location	0X43	3	4
XRL A,Rn	Exclusive OR register to Accumulator	0X68-0X6F	1	1
XRL A,direct	Exclusive OR directly addressed data to Accumulator	0X65	2	2
XRL A,@Ri	Exclusive OR indirectly addressed data to Accumulator	0X66-0X67	1	2
XRL A,#data	Exclusive OR immediate data to Accumulator	0X64	2	2
XRL direct,A	Exclusive OR Accumulator to directly addressed location	0X62	2	3
XRL direct,#data	Exclusive OR immediate data to directly addressed location	0X63	3	4
CLR A	Clear Accumulator	0XE4	1	1
CPL A	Complement Accumulator	0XF4	1	1
RL A	Rotate Accumulator left	0X23	1	1
RLC A	Rotate Accumulator left through carry	0X33	1	1
RR A	Rotate Accumulator right	0X03	1	1
RRC A	Rotate Accumulator right through carry	0X13	1	1
SWAP A	Swap nibbles within the Accumulator	0XC4	1	1

Data transfer Operations

Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to Accumulator	0XE8-0XEF	1	1
MOV A,direct	Move directly addressed data to Accumulator	0XE5	2	2
MOV A,@Ri	Move indirectly addressed data to Accumulator	0XE6-0XE7	1	2
MOV A,#data	Move immediate data to Accumulator	0X74	2	2
MOV Rn,A	Move Accumulator to register	0XF8-0XFF	1	1
MOV Rn,direct	Move directly addressed data to register	0XA8-0XAF	2	2
MOV Rn,#data	Move immediate data to register	0X78-0X7F	2	2
MOV direct,A	Move Accumulator to direct byte	0XF5	2	2
MOV direct,Rn	Move register to direct byte	0X88-0X8F	2	2
MOV direct1,direct2	Move directly addressed data to directly addressed location	0X85	3	3
MOV direct,@Ri	Move indirectly addressed data to directly addressed location	0X86-0X87	2	2
MOV direct,#data	Move immediate data to directly addressed location	0X75	3	3
MOV @Ri,A	Move Accumulator to indirectly addressed location	0XF6-0XF7	1	1
MOV @Ri,direct	Move directly addressed data to indirectly addressed location	0XA6-0XA7	2	2
MOV @Ri,#data	Move immediate data to indirectly addressed location	0X76-0X77	2	2
MOV DPTR,#data16	Load data pointer with a 16-bit immediate	0X90	3	3
MOVC A,@A+DPTR	Load Accumulator with a code byte relative to DPTR	0X93	1	4
MOVC A,@A+PC	Load Accumulator with a code byte relative to PC	0X83	1	4
MOVX A,@Ri	Move external RAM (8-bit addr.) to Accumulator	0XE2-0XE3	1	3
MOVX A,@DPTR	Move external RAM (16-bit addr.) to Accumulator	0XE0	1	3
MOVX @Ri,A	Move Accumulator to external RAM (8-bit addr.)	0XF2-0XF3	1	3
MOVX @DPTR,A	Move Accumulator to external RAM (16-bit addr.)	0XF0	1	3
PUSH direct	Push directly addressed data onto stack	0XC0	2	2
POP direct	Pop directly addressed location from stack	0XD0	2	3
XCH A,Rn	Exchange register with Accumulator	0XC8-0XCF	1	1
XCH A,direct	Exchange directly addressed location with Accumulator	0XC5	2	3
XCH A,@Ri	Exchange indirect RAM with Accumulator	0XC6-0XC7	1	2
XCHD A,@Ri	Exchange low-order nibbles of indirect and Accumulator	0XD6-0XD7	1	2

Program Branches

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx10001b	2	2(*2)/3
LCALL addr16	Long subroutine call	0X12	3	3(*2)/4
RET	Return from subroutine	0X22	1	5
RETI	Return from interrupt	0X32	1	5
AJMP addr11	Absolute jump	xxx00001	2	2(*2)/3
LJMP addr16	Long jump	0X02	3	3(*2)/4
SJMP rel	Short jump (relative addr.)	0X80	2	3(*2)/4
JMP @A+DPTR	Jump indirect relative to the DPTR	0X73	1	3
JZ rel	Jump if Accumulator is zero	0X60	2	3/4(*1)
JNZ rel	Jump if Accumulator is not zero	0X70	2	3/4(*1)
JC rel	Jump if carry flag is set	0X40	2	3/4(*1)
JNC rel	Jump if carry flag is not set	0X50	2	3/4(*1)
JB bit, rel	Jump if directly addressed bit is set	0X20	3	4/5(*1)
JNB bit, rel	Jump if directly addressed bit is not set	0X30	3	4/5(*1)
JBC bit, direct rel	Jump if directly addressed bit is set and clear bit	0X10	3	4/5(*1)
CJNE A, direct rel	Compare directly addressed data to Accumulator and jump if not equal	0XB5	3	4/5(*1)
CJNE A, #data rel	Compare immediate data to Accumulator and jump if not equal	0XB4	3	4/5(*1)
CJNE Rn, #data rel	Compare immediate data to register and jump if not equal	0XB8-0XBF	3	4/5(*1)
CJNE @Ri, #data rel	Compare immediate to indirect and jump if not equal	0XB6-0XB7	3	5/6(*1)
DJNZ Rn, rel	Decrement register and jump if not zero	0XD8-0XDF	2	3/4(*1)
DJNZ direct, rel	Decrement directly addressed location and jump if not zero	0XD5	3	4/5(*1)
NOP	No operation	0X00	1	1

Note: (*1) If the condition is true, the machine cycle will add 1.

(*2) If program execute ACALL/LCALL/AJMP/LJMP/SJMP and jump to the next address, the machine cycle will decrease 1.

Boolean Manipulation

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	0XC3	1	1
CLR bit	Clear directly addressed bit	0XC2	2	3
SETB C	Set carry flag	0XD3	1	1
SETB bit	Set directly addressed bit	0XD2	2	3
CPL C	Complement carry flag	0XB3	1	1
CPL bit	Complement directly addressed bit	0XB2	2	3
ANL C, bit	AND directly addressed bit to carry flag	0X82	2	2
ANL C,/bit	AND complement of directly addressed bit to carry	0XB0	2	2
ORL C, bit	OR directly addressed bit to carry flag	0X72	2	2
ORL C,/bit	OR complement of directly addressed bit to carry	0XA0	2	2
MOV C, bit	Move directly addressed bit to carry flag	0XA2	2	2
MOV bit, C	Move carry flag to directly addressed bit	0X92	2	3

Read-Modify-Write Instruction

Instructions that read a byte from SFR or internal RAM, modify it and rewrite it back, are called “Read-Modify-Write” instructions. When the destination is an I/O port (P0-P3), or a Port bit, these instructions read the output latch rather than the pin. Below table is RMW instruction set.

Mnemonic	Description	Code	Bytes	Cycles
ANL direct, A	AND accumulator to direct	0x52	2	3
ANL direct, #data	AND immediate data to direct	0x53	3	4
ORL direct, A	OR accumulator to direct	0x42	2	3
ORL direct, #data	OR immediate data to direct	0x43	3	4
XRL direct, A	Exclusive OR accumulator to direct	0x62	2	3
XRL direct, #data	Exclusive OR immediate data to direct	0x63	3	4
JBC bit, rel	Jump if bit is set and clear bit	0x10	3	4/5(*)
CPL bit	Complement bit	0xB2	2	3
INC direct	Increment direct	0x05	2	3
INC @Ri	Increment indirect	0x06-0x07	1	3
DEC direct	Decrement direct	0x15	2	3
DEC @Ri	Decrement indirect	0x16-0x17	1	3
DJNZ direct, rel	Decrement and jump if not zero	0xD5	3	4/5(*)
MOV bit, C	Move carry flag and direct bit	0x92	2	3
CLR bit	Clear bit	0xC2	2	3
SETB bit	Set bit	0xD2	2	3

Note: (*) If the condition is true, the machine cycle will add 1.

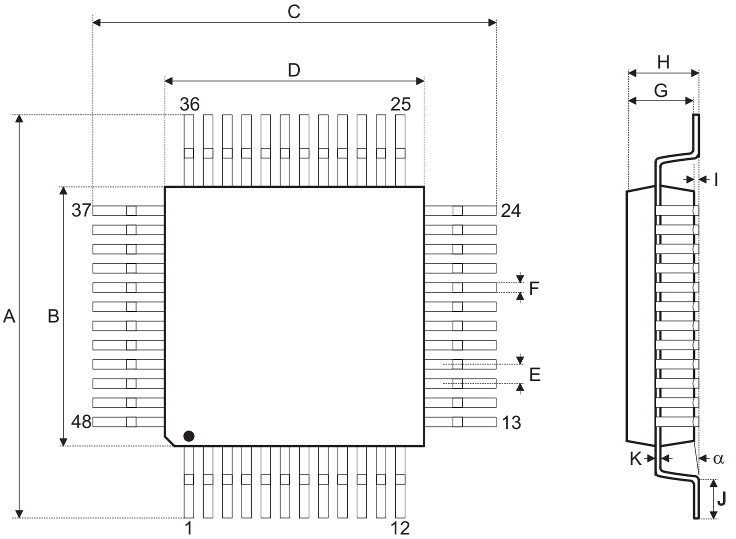
38 Package Information

Note that the package information provided here is for consultation purposes only. As this information may be updated at regular intervals users are reminded to consult the Holtek website for the latest version of the package information.

Additional supplementary information with regard to packaging is listed below. Click on the relevant section to be transferred to the relevant website page.

- Further Package Information (include Outline Dimensions, Product Tape and Reel Specifications)
- Packing Meterials Information
- Carton information

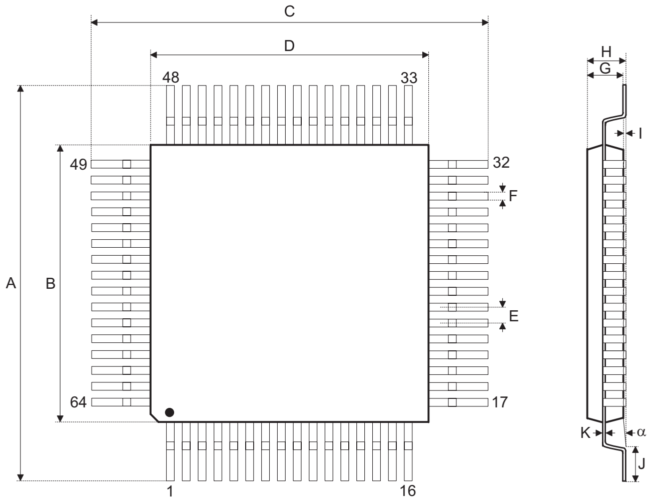
48-pin LQFP (7mm×7mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.354 BSC	—
B	—	0.276 BSC	—
C	—	0.354 BSC	—
D	—	0.276 BSC	—
E	—	0.020 BSC	—
F	0.007	0.009	0.011
G	0.053	0.055	0.057
H	—	—	0.063
I	0.002	—	0.006
J	0.018	0.024	0.030
K	0.004	—	0.008
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	9.00 BSC	—
B	—	7.00 BSC	—
C	—	9.00 BSC	—
D	—	7.00 BSC	—
E	—	0.50 BSC	—
F	0.17	0.22	0.27
G	1.35	1.40	1.45
H	—	—	1.60
I	0.05	—	0.15
J	0.45	0.60	0.75
K	0.09	—	0.20
α	0°	—	7°

64-pin LQFP (7mm × 7mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.354 BSC	—
B	—	0.276 BSC	—
C	—	0.354 BSC	—
D	—	0.276 BSC	—
E	—	0.016 BSC	—
F	0.005	0.007	0.009
G	0.053	0.055	0.057
H	—	—	0.063
I	0.002	—	0.006
J	0.018	0.024	0.030
K	0.004	—	0.008
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	9.0 BSC	—
B	—	7.0 BSC	—
C	—	9.0 BSC	—
D	—	7.0 BSC	—
E	—	0.4 BSC	—
F	0.13	0.18	0.23
G	1.35	1.40	1.45
H	—	—	1.60
I	0.05	—	0.15
J	0.45	0.60	0.75
K	0.09	—	0.20
α	0°	—	7°

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at http://www.holtek.com.tw.