



A/D Flash 8051 Core Low Pin Count MCU

HT85F2230/HT85F2240

Revision: V1.00 Date: January 15, 2015

www.holtek.com

Table of Contents

1 Features	11
CPU Features	11
Peripheral Features	12
2 General Description	13
3 Selection Table	14
4 Block Diagram	15
5 Pin Assignment	16
6 Pin Descriptions	17
7 Absolute Maximum Ratings	19
8 D.C. Characteristics	19
9 A.C. Characteristics	21
10 ADC Electrical Characteristics	22
11 Comparator Electrical Characteristics	23
12 Power on Reset Electrical Characteristics	24
13 System Architecture	24
14 Program Counter	25
15 Stack	25
16 Arithmetic and Logic Unit – ALU	26
17 Flash Program Memory	27
Structure	27
Special Vectors	27
In-Circuit Programming – ICP	28
On-Chip Debug Support – OCDS	28
In-Application Programming – IAP	29
Flash Program Memory Resistors	29
Flash Memory Read/Write Operations	33
Unlocking the Flash Memory	33
Page Erase Operation	34
Byte Read Operation	35
Byte Write Operation	36
Program Memory Protection	38
Memory Protection Control Bytes	38

18 RAM Data Memory	41
Structure.....	41
Register Banks	44
Bit Addressable Space	44
Special Function Registers.....	46
ACC Register – Accumulator.....	47
B Register	47
SP Register – Stack Pointer	47
DPL, DPH, DPL1, DPH1 Registers – Data Pointer Registers	47
Data Pointer Select Registers	48
Data Pointer Control Register.....	48
Program Status Word	50
19 Oscillators	51
System Oscillator Overview	51
System Clock Configuration	51
External High Speed Crystal Oscillator – HXT	51
Internal High Speed RC Oscillator – HIRC.....	52
Internal Low Speed RC Oscillator – LIRC	52
20 Operating Modes and System Clocks	53
System Clocks Description.....	53
Operation Modes.....	57
NORMAL Mode	58
IDLE Mode.....	58
Power - Down Mode	58
Power Control Register	58
Standby Current Considerations	59
Wake-up	59
21 Watchdog Timer.....	60
Watchdog Registers	61
Watchdog Timer Clock Source	64
Watchdog Timer Operation	64
22 Low Voltage Detector – LVD	66
LVD Register	66
LVD Operation.....	66

23 Reset and Initialisation.....	67
Reset Overview	67
Reset Operations	68
Reset Source Register – RSTSRC.....	68
Power-on Reset.....	69
RESET Pin Reset.....	70
Low Voltage Reset – LVR.....	71
Watchdog Reset	72
Software Resets	72
SRST Register Software Reset.....	73
WDTCR Register Software Reset	73
LVRCCR Register Software Reset.....	74
Reset Initial Conditions.....	74
24 Interrupts	79
Interrupt Registers.....	79
Interrupt Operation	85
Interrupt Priority.....	86
Priority Levels	86
Priority Control Registers.....	87
External Interrupt.....	90
A/D Converter Interrupt	90
Timer/Counter Interrupt.....	90
Time Base Interrupts	91
LVD Interrupt	92
Interrupt Wake-up Function	93
Programming Considerations.....	93
25 Input/Output Ports	94
Input/Output Port Overview	94
Register Description	95
PnM0/PnM1 Registers – Port Mode Registers.....	96
P0WAKE Register – Port 0 Wake-up	98
SRCR Register – Slew Rate Control.....	99
Pin-remapping Function	99
PRM Register – Pin-remapping Control Register.....	99
I/O Pin Structures	100
Quasi-bidirectional I/O	100
Push-pull Output	101
Open-drain Output	101
Input Only	101
Programming Considerations.....	102

26 Timer/Event Counters	103
Timer/Event Counter Summary	103
27 Timer/Event Counters 0, 1	104
Introduction.....	104
Timer 0/Timer 1 Register Description.....	104
Mode 0 – 13-bit Counter/Timer Mode Operation.....	109
Mode 1 – 16-bit Counter/Timer Mode Operation.....	110
Mode 2 – 8-bit Auto-reload Counter/Timer Mode Operation	111
Mode 3 – Two 8-Bit Timers/Counters Mode Operation – Timer 0 Only.....	112
28 Timer 2 with Additional 2-channel PCA	113
Introduction.....	113
Timer 2	114
Timer function	114
Event Counter function	114
Gated Timer function	115
Timer 2 with PCA.....	115
Timer 2 Register Description	116
Capture Modes.....	118
Capture On Edge Mode.....	118
Capture On Write Mode.....	118
Compare Modes.....	119
Compare Mode 0.....	119
Compare Mode 1	120
Reload Mode	122
Programmable Clock Output Mode.....	123
29 Analog to Digital Converter – ADC	124
A/D Overview	124
A/D Converter Register Description	125
A/D Converter Data Registers – ADRL, ADRH	125
A/D Converter Control Registers – ADCR0, ADCR1, ADCR2, ADPGA	125
A/D Operation.....	129
A/D Converter Clock Source	130
A/D Input Pins	130
Temperature Sensor.....	131
A/D Reference Voltage Source	131
Summary of A/D Conversion Steps.....	132
A/D Conversion Timing.....	133
Programming Considerations.....	133
A/D Transfer Function	134

30 Voltage Reference Generator	135
Voltage Reference Generator Operation.....	135
31 Instruction Set.....	136
Introduction.....	136
Read-Modify-Write Instruction.....	141
32 Package Information	142
20-pin DIP (300mil) Outline Dimensions	143
20-pin SOP (300mil) Outline Dimensions.....	145
20-pin SSOP (150mil) Outline Dimensions	146
24-pin SKDIP (300mil) Outline Dimensions	147
24-pin SOP (300mil) Outline Dimensions.....	150
24-pin SSOP (150mil) Outline Dimensions	151

List of Tables

Program Memory Register List	29
FMAR0 Register – Flash Program Memory Address Register 0	29
FMAR1 Register – Flash Program Memory Address Register 1	30
FMAR2 Register – Flash Program Memory Address Register 2	30
FMDR Register – Flash Program Memory Data Register	30
FMCR Register – Flash Program Memory Control Register	31
FMKEY Register – Flash Program Memory Unlock Key Data Register	32
FMSR Register – Flash Program Memory Status Register	32
HT85F2240 Program Memory Contents.....	39
HT85F2230 Program Memory Contents.....	39
Security Bytes	40
General Purpose Data RAM, 20H~2FH, Bit Address Map	44
Special Function Register Bit Addresses Map.....	45
Special Function Register Map.....	46
DPS Register – Data Pointer Select Register.....	48
DPC Register – Data Pointer Control Register.....	49
PSW Register – Program Status Word Register	50
Crystal Recommended Capacitor Values	52
System Clock Control Register – SCCR.....	55
High Speed Oscillator Control Register – HSOCR	56
HIRC Frequency Select Control Register – HFSCR.....	56
Low Speed Oscillator Control Register – LSOCR	57
PCON Register – Power Control Register.....	58
WDT Register Contents	61
IEN0 Register	61
IEN1 Register	62
WDTREL Register	62
WDTCR Register	63
IP0 Register	63
Watchdog Timer Enable/Disable Control.....	64
LVDCCR Register	66
Reset Source Summary.....	67
RSTSRC Register.....	68
LVRCCR Register	71
IP0 Register	72
T2CON1 Register	73
Software Reset Summary	73
SRST Register	74
WDTCR Register	74
LVRCCR Register.....	75
Interrupt Register Bit Naming Conventions	80
Interrupt Register Contents.....	80
IEN0 Register	81

IEN1 Register	81
IEN2 Register	82
IEN3 Register	82
IRCON Register.....	83
T2CON Register	84
IRCON2 Register.....	84
TCON Register	85
Low byte of Interrupt Priority Register 0: IP0	88
High byte of Interrupt Priority Register 0: IP0H.....	88
Low byte of Interrupt Priority Register 1: IP1	89
High byte of Interrupt Priority Register 1: IP1H	89
Low byte of Interrupt Priority Register 2: IP2	89
High byte of Interrupt Priority Register 2: IP2H	90
Low byte of Interrupt Priority Register 3: IP3	90
How byte of Interrupt Priority Register 3: IP3H	90
External Interrupt Trigger Type	91
TBCR Register.....	93
I/O Port Function Summary	95
I/O Register List.....	96
P0 Register	96
P1 Register	96
P3 Register	97
P0M0 Register	97
P0M1 Register	97
P1M0 Register	98
P1M1 Register	98
P2M0 Register	98
P2M1 Register	98
P3M0 Register	98
P3M1 Register	98
P0WAKE Register.....	99
SRCR Register	100
PRM Register	100
Timer Function Summary.....	104
Timer0/Timer1 Register List.....	105
TL0 Register	106
TH0 Register	106
TL1 Register	107
TH1 Register	107
TMOD Register.....	108
TCON Register	109
TMPRE Register.....	110
13-bit Counter Data	110
Timer 2 with PCA Modules Operating Modes Summary.....	114
Timer 2 with PCA Modules I/O Pins.....	114
Timer 2 Register List.....	117

CCEN Register	117
T2CON Register	118
T2CON1 Register	118
A/D Converter Register List	126
A/D Data Registers	126
ADCR0 Register	127
ADCR1 Register	128
ADCR2 Register	129
ADPGA Register	130
A/D Clock Period Examples	131
A/D Converter Voltage Reference Select	132
Internal Voltage Reference Enable/Disable Control	136
Notes on Data Addressing Modes	137
Notes on Program Addressing Modes	137
Arithmetic Operations	138
Logic Operations	139
Data transfer Operations	140
Program Branches	141
Boolean Manipulation	142

List of Figures

Stack Block Diagram	25
Program Memory Structure	27
Unlock Procedure Flowchart.....	33
Page Erase Flowchart	34
Byte Read Flowchart	35
Byte Write Flowchart (FMCR.0=1, FMCR.6=0)	36
Byte Write Flowchart (FMCR.0=1, FMCR.6=1)	37
Internal Data Memory Structure.....	42
HT85F2240/HT85F2230 XDATA.....	43
DPTRn Registers Control Block Diagram.....	47
Crystal/Resonator Oscillator – HXT	52
System Clock Configurations.....	54
Watchdog Timer.....	60
Watchdog Timer Refresh Operation	65
Power-On Reset Timing	69
Interrupt Structure	85
Quasi-bidirectional I/O Structure.....	100
Push-pull Output Structure	101
Open-drain Output Structure	101
Input Only Structure	101
Mode 0 and Mode 1 Block Diagram – Timer 0, 1	110
Mode 2 Block Diagram – Timer 0, 1	111
Mode 3 Block Diagram – Timer 0	112
Timer 2 with PCA Modules Block Diagram	114
Capture Modes Block Diagram.....	118
Compare Mode 0 – Module 0~1	119
Compare Match Mode 0 Timing Diagram – T2CM=0	120
Compare Mode 1 – Module0~1	120
Compare Match Mode 1 Timing Diagram – T2CM=1	121
Reload Mode – Module 0.....	122
Timer2 Clock Output Block Diagram.....	123
Programmable Clock Output Timing Diagram – Module 0	123
A/D Converter Structure	124
Temperature vs Voltage Diagram	131
A/D Conversion Timing	133
Ideal A/D Transfer Function (PGA Gain=1).....	134
Voltage Reference Generator Block Diagram.....	135

1 Features

CPU Features

- Operating Voltage:
 - $f_{\text{SYS}}=8\text{MHz}$: 2.2V~5.5V
 - $f_{\text{SYS}}=12\text{MHz}$: 2.7V~5.5V
 - $f_{\text{SYS}}=16\text{MHz}$: 3.3V~5.5V
 - $f_{\text{SYS}}=24\text{MHz}$: 4.5V~5.5V
- High performance 1-T architecture 8051 core
- Up to 24MIPS with 24MHz system clock at $V_{\text{DD}}=5\text{V}$
- Flexible Power-down and wake-up functions to reduce power consumption
- Oscillator types:
 - External high frequency crystal – HXT
 - Internal high frequency RC – HIRC
 - Internal low frequency RC – LIRC
- Multi-mode operation: Normal, Idle and Power-Down Modes
- Fully integrated internal 12/16/20MHz oscillator requires no external components
- 8051 compatible instruction set
- Dual 16-bit data pointers with addition arithmetic operation

Peripheral Features

- Program Memory Capacity: 2K×8 ~ 4K×8
- Data Memory Capacity: 352×8
- Watchdog Timer function
- Up to 21 bidirectional I/O lines
- Two pin-shared external interrupts
- Three 16-bit programmable Timer/Counters
- 16-bit Programmable Counter Array with 2 Capture/Compare Modules
- Multi-channel 12-bit resolution A/D converter
- Single Time-Base function for generation of fixed time interrupt signal
- Internal Temperature Sensor
- Low voltage reset function
- Low voltage detect function
- Wide range of available package types

2 General Description

The HT85F2230 and HT85F2240 are A/D Flash type low pin count 1-T architecture 8051 based microcontrollers. These devices are designed for general use, however in having an internal A/D converter, they can directly interface to analog signals, such as those from sensors. Offering users the convenience of Flash Memory multi-programming features, these devices also include a wide range of functions and features.

The device's main features include a multi-channel 12-bit A/D Converter and multiple timers which provide timing, capture, event counting and programmable clock output functions. 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 external and internal low and high speed oscillator functions are provided with the internal oscillators requiring no external components for their implementation. 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 minimise power consumption.

The inclusion of flexible I/O programming features and Time-Base function along with many other features ensure that the devices will find excellent use in applications such as sensor signal processing, motor driving, industrial control, consumer products and subsystem controllers in addition to many others.

As these MCUs are Flash memory type devices, they offer the advantages of easy and effective in-circuit program updates. In addition, an EV chip, in the form of the HT85V2240 device, with its OCDS (On-Chip Debug Support) interface, is supplied for full in-circuit emulation purposes.

3 Selection Table

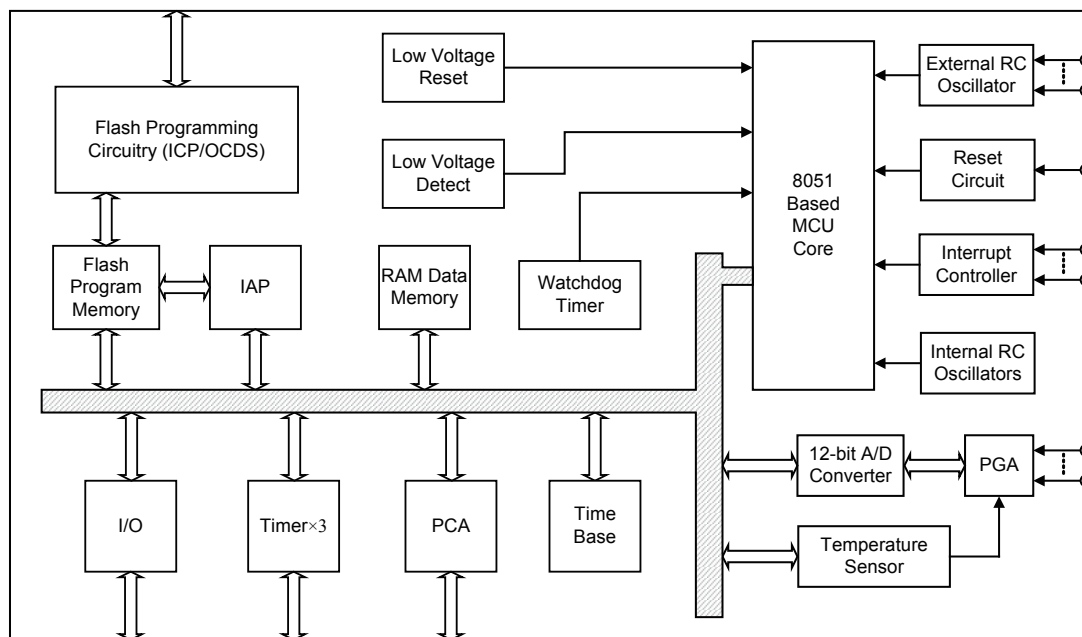
Most features are common to all devices, the main feature distinguishing them are Program Memory capacity, I/O count and package types. The following table summarises the main features of each device.

Part No.	V _{DD}	Program Memory	Data Memory	I/O	Ext. Interrupt	A/D	Temp. Sensor	16-bit Timer	16-bit PCA	Time Base	Package
HT85F2230	2.2V~5.5V	2K×8	352×8	17	2	12-bit×9	√	3	CCU×2	1	20DIP, 20SOP, 20SSOP
HT85F2240	2.2V~5.5V	4K×8	352×8	21	2	12-bit×9	√	3	CCU×2	1	20DIP, 20SOP, 20SSOP, 24SKDIP, 24SOP, 24SSOP

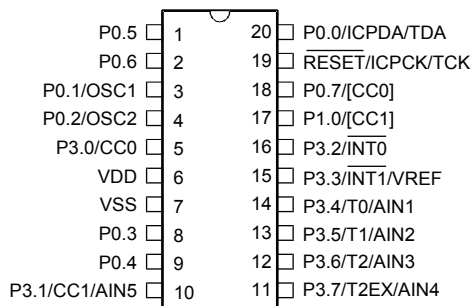
Note: CCU stands for Compare/Capture Unit.

4 Block Diagram

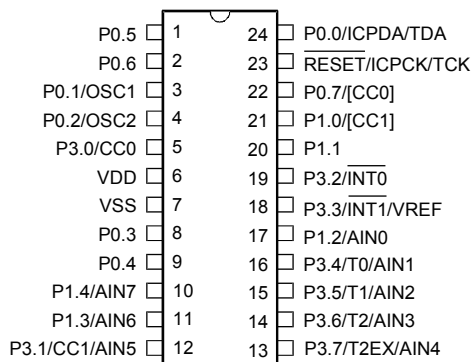
The following block diagram illustrates the main functional blocks.



5 Pin Assignment



HT85F2240/HT85F2230/HT85V2240
20 DIP-A/SOP-A/SSOP-A



HT85F2240/HT85V2240
24 SKDIP-A/SOP-A/SSOP-A

- 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. Both real IC and OCDS EV IC share the same package.
3. The TDA and TCK pins are the OCDS dedicated pins and only available for the HT85V2262 device which is the OCDS EV chip for both the HT85F2240 and HT85F2230 devices.

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 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, only for EV chip
P0.1/OSC1	P0.1	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up.
	OSC1	HSOCR	HXT	—	HXT pin
P0.2/OSC2	P0.2	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up.
	OSC2	HSOCR	—	HXT	HXT pin
P0.3~P0.6	P0.3~P0.6	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up.
P0.7/[CC0]	P0.7	P0M0 P0M1 P0WAKE	ST	CMOS	General purpose I/O. Register selected I/O mode and wake-up.
	CC0	PRM	ST	CMOS	Compare/Capture input/output for PCA module 0
P1.0/[CC1]	P1.0	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	CC1	PRM	ST	CMOS	Compare/Capture input/output for PCA module 1
P1.1	P1.1	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
P1.2/AIN0	P1.2	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN0	ADCR2	AN	—	ADC Input Channel 0
P1.3/AIN6	P1.3	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN6	ADCR2	AN	—	ADC Input Channel 6
P1.4/AIN7	P1.4	P1M0 P1M1	ST	CMOS	General purpose I/O. Register selected I/O mode
	AIN7	ADCR2	AN	—	ADC Input Channel 7
P3.0/CC0	P3.0	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	CC0	PRM	ST	CMOS	Compare/Capture input/output for PCA module 0

Pin Name	Function	OPT	I/T	O/T	Description
P3.1/CC1/AIN5	P3.1	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	CC1	PRM	ST	CMOS	Compare/Capture input/output for PCA module 1
	AIN5	ADCR2	AN	—	ADC Input Channel 5
P3.2/ $\overline{\text{INT0}}$	P3.2	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	$\overline{\text{INT0}}$	IEN0 TCON	ST	—	External Interrupt 0 Input
P3.3/ $\overline{\text{INT1}}$ /VREF	P3.3	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	$\overline{\text{INT1}}$	IEN0 TCON	ST	—	External Interrupt 1 Input
	VREF	ADCR1	AN	—	Reference Voltage for ADC
P3.4/T0/AIN1	P3.4	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	T0	—	ST	—	Timer 0 External Input
	AIN1	ADCR2	AN	—	ADC Input Channel 1
P3.5/T1/AIN2	P3.5	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	T1	—	ST	—	Timer 1 External Input
	AIN2	ADCR2	AN	—	ADC Input Channel 2
P3.6/T2/AIN3	P3.6	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	T2	—	ST	—	Timer 2 External Input
	AIN3	ADCR2	AN	—	ADC Input Channel 3
P3.7/T2EX/AIN4	P3.7	P3M0 P3M1	ST	CMOS	General purpose I/O. Register selected I/O mode.
	T2EX	—	ST	—	Timer 2 capture trigger
	AIN4	ADCR2	AN	—	ADC Input Channel 4
$\overline{\text{RESET}}$ /ICPCK/ TCK	$\overline{\text{RESET}}$	—	ST	—	Reset pin
	ICPCK	—	ST	—	ICP Clock Input
	TCK	—	ST	—	Debug Clock Input, only for EV chip
VDD	VDD	—	PWR	—	Positive Power supply for CORE
VSS	VSS	—	PWR	—	Negative Power supply

Note: I/T: Input type; O/T: Output type
OPT: Optional by register option
PWR: Power; ST: Schmitt Trigger input
CMOS: CMOS output; AN: Analog input pin
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	$-50^{\circ}C$ to $125^{\circ}C$
Operating Temperature	$-40^{\circ}C$ to $85^{\circ}C$
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

$T_a=25^{\circ}C$

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD1}	Operating Voltage (High Frequency Internal RC OSC)	—	$f_{OSC}=f_{SYS}=12MHz$ HFSCR[1:0]=00b	2.7	—	5.5	V
			$f_{OSC}=f_{SYS}=16MHz$ HFSCR[1:0]=10b	3.3	—	5.5	
			$f_{OSC}=f_{SYS}=20MHz$ HFSCR[1:0]=11b	4.5	—	5.5	
V_{DD2}	Operating Voltage (Crystal OSC)	—	$f_{OSC}=f_{SYS}=8MHz$	2.2	—	5.5	V
			$f_{OSC}=f_{SYS}=12MHz$	2.7	—	5.5	
			$f_{OSC}=f_{SYS}=16MHz$	3.3	—	5.5	
			$f_{OSC}=f_{SYS}=24MHz$	4.5	—	5.5	
I_{DD1}	Operating Current (High Frequency Internal RC OSC)	3V	No load, $f_{OSC}=f_{SYS}=12MHz$, ADC off, WDT enable	—	8	12	mA
		5V		—	16	25	
		5V	No load, $f_{OSC}=f_{SYS}=16MHz$, ADC off, WDT enable	—	20	30	mA
		5V	No load, $f_{OSC}=f_{SYS}=20MHz$, ADC off, WDT enable	—	28	40	mA
I_{DD2}	Operating Current (Crystal OSC)	3V	No load, $f_{OSC}=f_{SYS}=8MHz$, ADC off, WDT enable	—	6.0	8.5	mA
		5V		—	12.5	20	
		3V	No load, $f_{OSC}=f_{SYS}=12MHz$, ADC off, WDT enable	—	8	12	mA
		5V		—	16	25	
		5V	No load, $f_{OSC}=f_{SYS}=16MHz$, ADC off, WDT enable	—	20	30	mA
		5V	No load, $f_{OSC}=f_{SYS}=24MHz$, ADC off, WDT enable	—	28	40	mA
I_{STB1}	Standby Current (Power-Down Mode) (HIRC off, HXT off)	3V	No load, All peripherals off	—	—	1.5	μA
		5V		—	—	2.5	

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
I _{STB2}	Standby Current (Idle) (HIRC off, HXT on)	3V	No load, f _{OSC} =4MHz (Crystal OSC) f _{SYS} off, ADC off, LVD/LVR disable, WDT enable	—	1.5	2.5	mA
		5V		—	3.5	5.0	
V _{IL1}	Input Low Voltage (except RESET pin)	—	quasi-bidirection mode	0	—	0.2V _{DD}	V
V _{IH1}	Input High Voltage (except RESET pin)	—	quasi-bidirection mode	0.8V _{DD}	—	V _{DD}	V
V _{IL2}	Input Low Voltage (RESET pin)	—	—	0	—	0.4V _{DD}	V
V _{IH2}	Input High Voltage (RESET pin)	—	—	0.9V _{DD}	—	V _{DD}	V
I _{OL}	I/O Port Sink Current	2.2V	V _{OL} =0.4V	3.0	6.0	—	mA
		3.3V		4.5	9.0	—	
		5.0V		6.0	12.0	—	
I _{OH1}	I/O Port Source Current (Push-pull Mode)	2.2V	V _{OH} =0.9V _{DD}	-0.5	-1.0	—	mA
		3.3V		-1.0	-2.0	—	
		5.0V		-2.0	-4.0	—	
I _{OH2}	I/O Port Source Current (Quasi-bidirection Mode)	2.2V	V _{OH} =0.9V _{DD}	-20	-40	—	μA
		3.3V		-40	-80	—	
		5.0V		-80	-160	—	
I _{IL}	Logical 0 Input Current (Quasi-bidirection Mode)	5V	V _{IN} =0.4V	—	—	-50	μA
I _{TL}	Logical 1 to 0 Transition Current (Quasi-bidirection Mode)	5V	V _{IN} =2.4V	—	—	-950	μA
I _{LI}	Input Leakage Current (Input Mode)	5V	0.45V<V _{IN} <V _{DD} -0.3	—	—	±1	μA
V _{BG}	Bandgap Reference with Buffer Voltage	—	—	-3%	1.1	+3%	V
I _{BG}	Additional Power Consumption if Reference with Buffer is Used	—	—	—	200	300	μA
I _{LVR}	Additional Power Consumption if LVR is Used	3V	LVR enable	—	75	100	μA
		5V		—	75	100	
I _{LVD}	Additional Power Consumption if LVD is Used	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	—	0.4	—	8	MHz
		2.7V~5.5V		0.4	—	12	
		3.3V~5.5V		0.4	—	16	
		4.5V~5.5V		0.4	—	24	
f _{SYS2}	System Clock (High Frequency Internal RC Oscillator)	2.7V~5.5V	Ta=25°C HFSCR[1:0]=00b	-3%	12	+3%	MHz
		3.3V~5.5V	Ta=25°C HFSCR[1:0]=10b		16		
		4.5V~5.5V	Ta=25°C HFSCR[1:0]=11b		20		
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}	12/16/20MHz Internal RC Oscillator	2.2V~5.5V	Ta=25°C HFSCR[1:0]=00b	-3%	12	+3%	MHz
f _{TIMER}	Timer Input Frequency (T0~T2)	2.2V~5.5V	f _{sys} =8MHz	0	—	2	MHz
		3.3V~5.5V	f _{sys} =16MHz	0	—	4	
		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 _{HIO}	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	—	V _{REF} available	0	—	V _{REF}	V
		—	V _{REF} 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} =0.5μs	-2	—	+2	LSB
INL	Integral Non-linearity	—	AV _{DD} =5V V _{REF} =AV _{DD} t _{ADCK} =0.5μs	-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 Comparator Electrical Characteristics

Ta=25°C

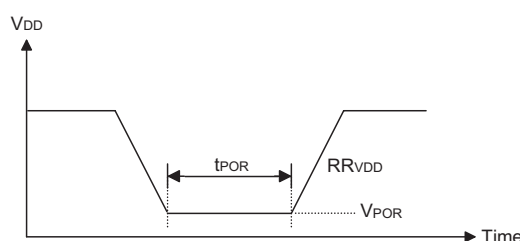
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
—	Comparator operating voltage	—	—	2.2	—	5.5	V
—	Comparator operating current	5V	LVDCR=00h, ADCR1=08h, i.e. select internal bandgap voltage output (x2) as VREFI	—	—	10	μA
—	Comparator power-down current	5V	Comparator disable	—	—	0.1	μA
V _{CMPOS}	Comparator input offset voltage	5V	—	-10	—	+10	mV
V _{HP1}	Positive Hysteresis 1	5V	CP0HP[1:0]=00b	—	0	1	mV
V _{HP2}	Positive Hysteresis 2	5V	CP0HP[1:0]=01b	3	6	10	mV
V _{HP3}	Positive Hysteresis 3	5V	CP0HP[1:0]=10b	6	13	20	mV
V _{HP4}	Positive Hysteresis 4	5V	CP0HP[1:0]=11b	12	25	40	mV
V _{HN1}	Negative Hysteresis 1	5V	CP0HN[1:0]=00b	—	0	1	mV
V _{HN2}	Negative Hysteresis 2	5V	CP0HN[1:0]=01b	3	6	10	mV
V _{HN3}	Negative Hysteresis 3	5V	CP0HN[1:0]=10b	6	13	20	mV
V _{HN4}	Negative Hysteresis 4	5V	CP0HN[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	With 100mV overdrive (Note)	—	4	—	μs
		5V					

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).

12 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



13 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 and A/D control system with maximum reliability and flexibility.

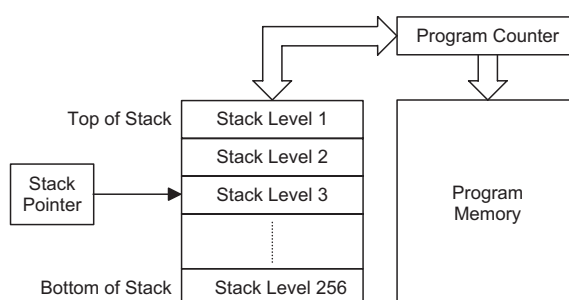
14 Program Counter

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.

15 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 Internal 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 writable. 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.

16 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

17 Flash Program Memory

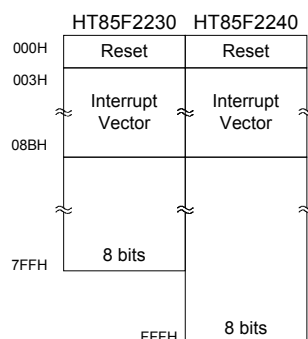
The Program Memory is the location where the user code or program is stored. For these devices the Program Memory is Flash type, which means it can be programmed and re-programmed a large number of times, allowing the user the convenience of code modification on the same device. By using the appropriate 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 2K×8 to 4K×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 0000H 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	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, HT85V2240, is provided which includes all the HT85F2240 functions as well as an “On-Chip Debug” interface for emulation of the HT85F2240/2230 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 Registers

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 way as any other Special Function Register.

Program Memory Register List

Register 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

When writing to the Flash Memory Address Registers, they must be written in the order of FMAR2 register first and FMAR0 register last.

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 with a value of 0xFF within a page, a write operation is allowed. But for those written bytes with any values 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 each of the security bytes 00h~1Fh in the ID block page 0 can only be written once, the only method to release the protected bytes is by “ICP Whole Chip Erase”.
- 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 using the application program.
- Bit 1 **FMCR.1:** Flash Memory Byte Read control bit
0: Disable
1: Enable
This bit should be cleared using the application program.
- Bit 0 **FMCR.0:** Flash Memory Byte Write control bit
0: Disable
1: Enable
This bit should be cleared using the application program.

Note that the valid combinations for FMCR.2~FMCR.0 are 100, 010 and 001, all other combinations should be ignored.

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 rule or not. The security rule means to write a FFH value to SECURITY1[N] or SECURITY2[N] to set the sector N in the unprotected mode before a flash memory manipulation. Otherwise, the manipulation violates the security rule and the FMSEF flag will be set.

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

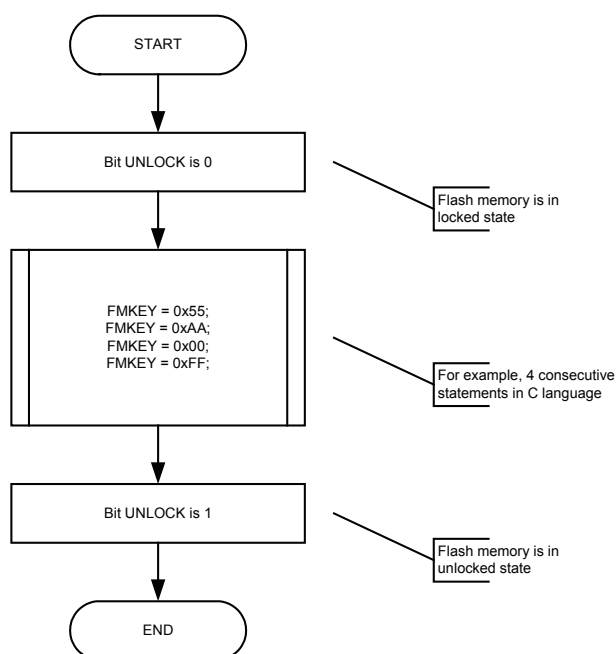
- 0: Not erasing or rewriting
- 1: Busy

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

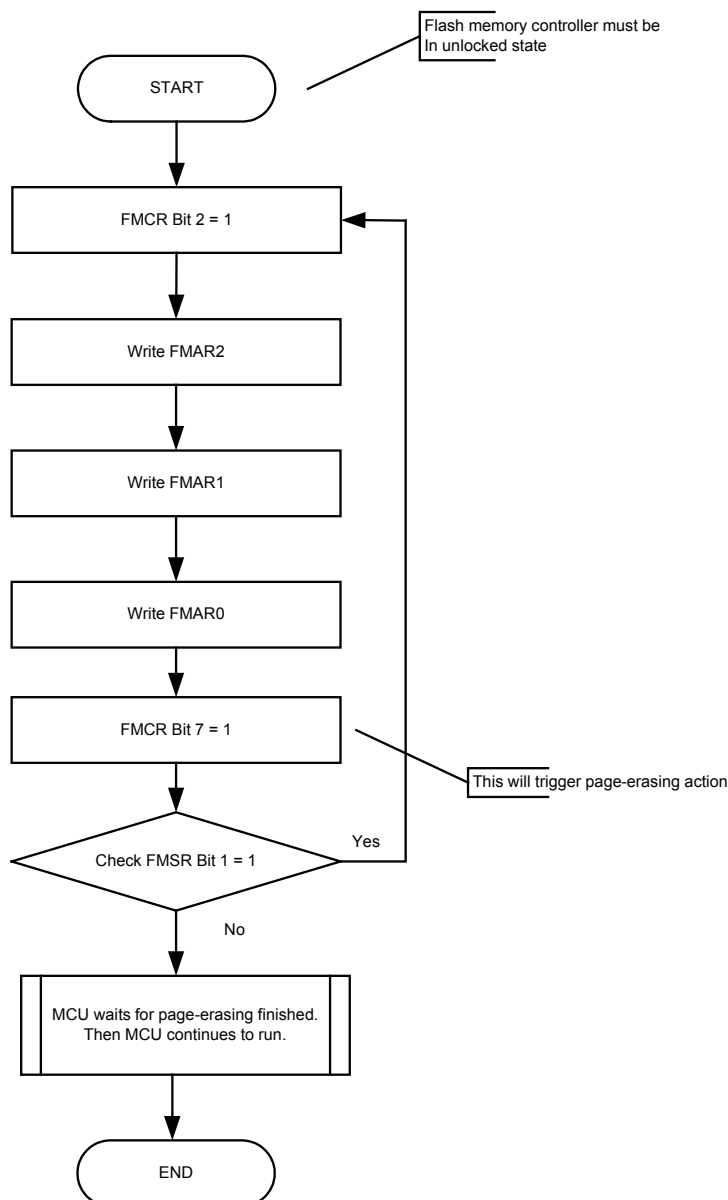
Before writing data to the Flash Memory it must first be unlocked. This is implemented by writing a correct data sequence to the Flash Memory Unlock key register, FMKEY. It is recommended to write the data sequence to the FMKEY register in 4 consecutive instructions. The following flowchart illustrates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



Unlock Procedure Flowchart

Page Erase Operation

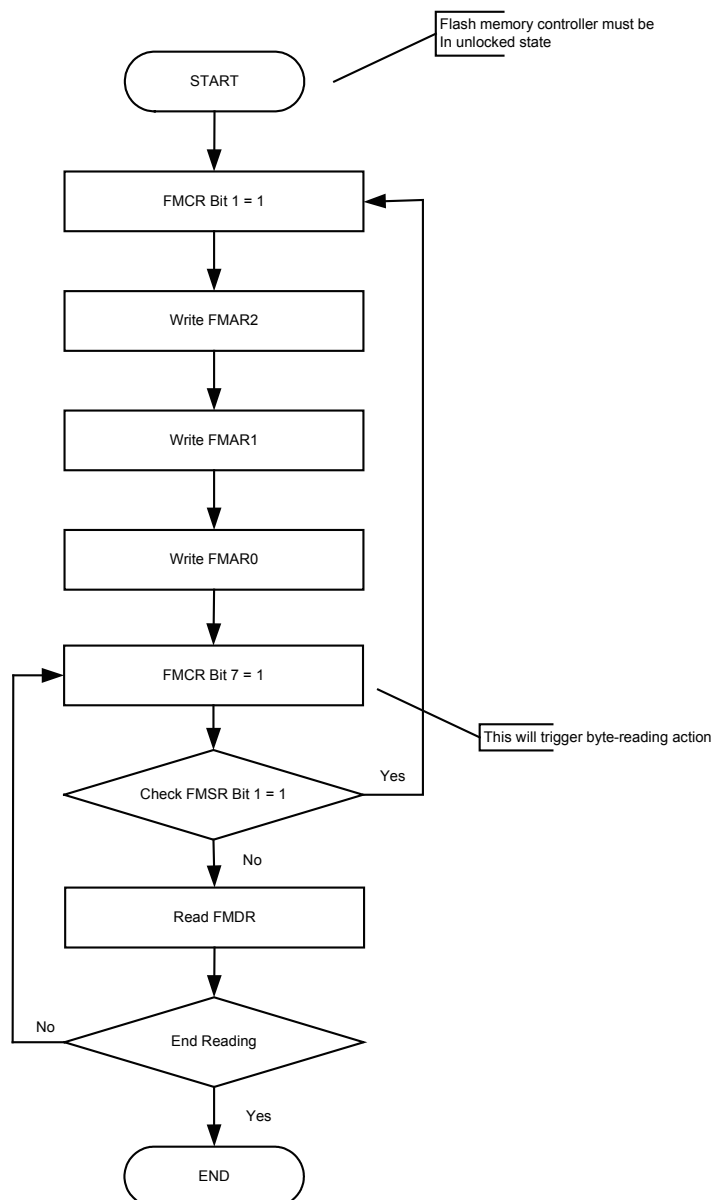
The Flash Memory must be first unlocked before implementing a page erase procedure. The flash memory address is setup using the control registers, FMAR0, FMAR1 and FMAR2. The Flash Memory Page Erase function is selected by the control bit, FMCR.2, in the FMCR register. Setting the FMCR.7 bit high will 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.



Page Erase Flowchart

Byte Read Operation

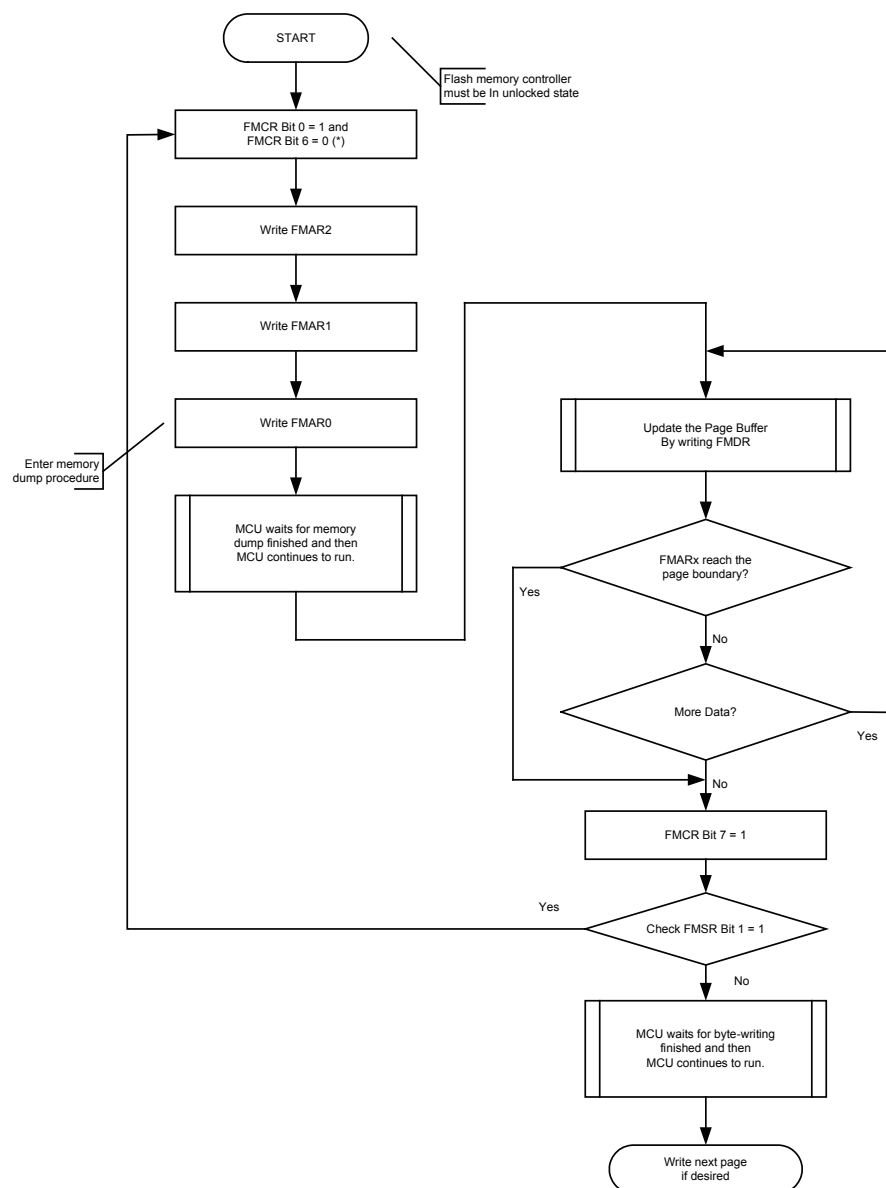
The Flash memory must be first unlocked before implementing a byte read procedure. The flash memory address is setup using the control registers, FMAR0, FMAR1 and FMAR2. The Flash Memory Page Read function is selected by the control bit, FMCR.1, in the FMCR register. When the FMCR.7 bit is set high the 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.



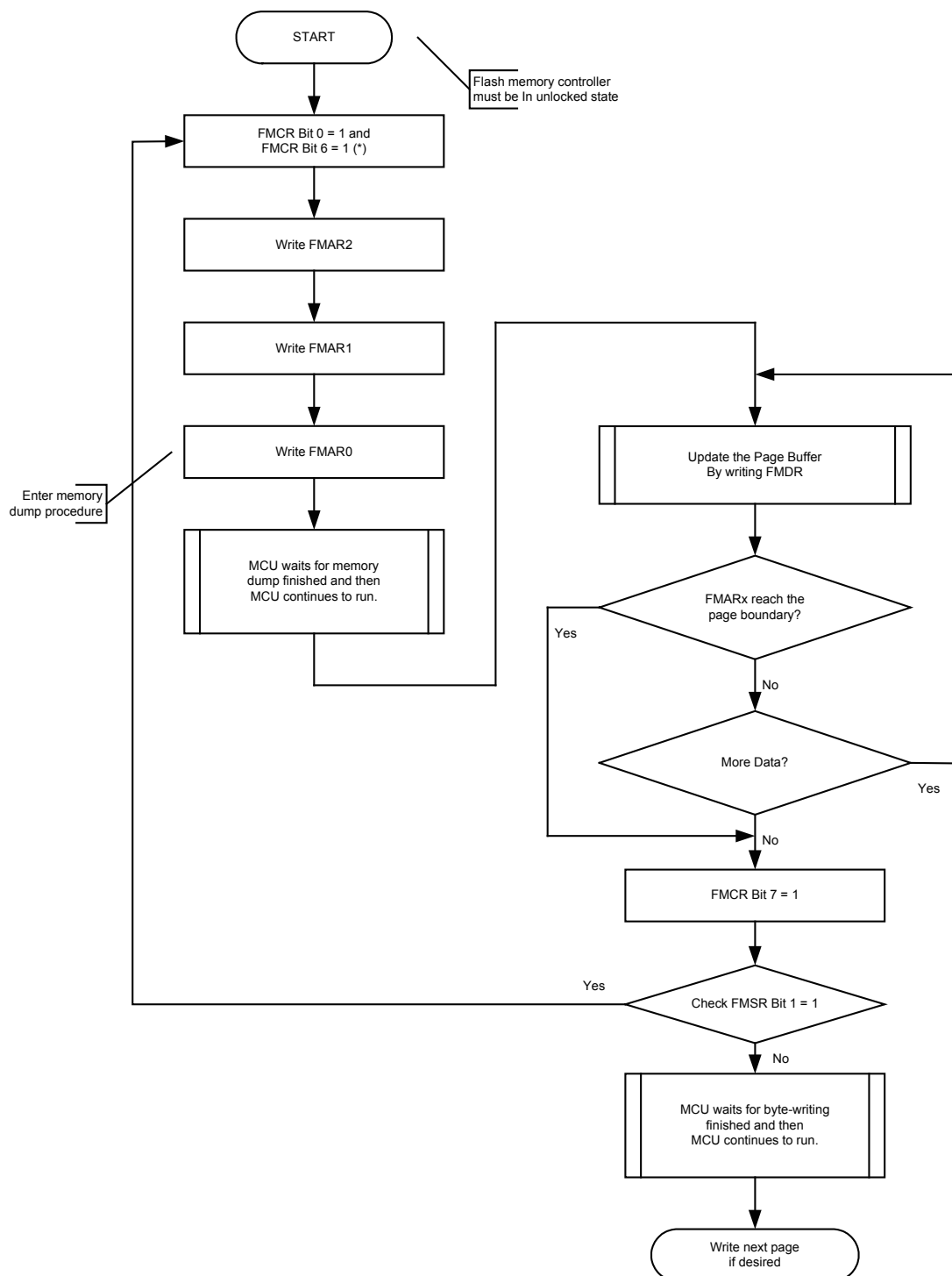
Byte Read Flowchart

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.



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



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

Program Memory Protection

The flash program memory is partitioned into two memory blocks. One is the main memory block and the other is the ID block. The ID block size is 64 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 according to the memory size. Each sector has a capacity of 1K 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

Take the HT85F2240 as an example, its inhibit bytes SECURITY1[0:3] are located at the address 0x00~0x03 of the ID block page 0. If a value, with the exception of 0FFH, has been written to SECURITY1[N], this sector can not be programmed, erased or read when in the ICP mode or when executing the OCDSINSTR instruction in the OCDS mode.

With regard to the IAP program, in the OCDS mode, when using the OCDSINSTR instruction to execute the IAP program, any sector with a security mechanism can be protected from being programmed, erased or read. If the IAP program is executed when running the main program, all sectors with security or not, can be programmed, erased or read.

With regard to the MOVC instruction, in the OCDS mode, when using the OCDSINSTR instruction to execute the MOVC instruction, any sector with security mechanism can not be read. When executing the MOVC instruction in the main program, all sectors with security or not, can be read.

The following table illustrates the protection status in the OCDS/ICP/IAP/MOVC modes when the SECURITY1[0:3] bytes are written with a value other than 0FFH:

SECURITY1[N] N=0~3		Program	Erase	Read	Protect Sector #	Remove Protection
IAP	ICP	X	X	N/A ⁽¹⁾	N	Erase All
	OCDS ⁽⁵⁾	X ⁽³⁾	X	X ⁽⁴⁾	N	Erase All
	Main Program	O ⁽³⁾	O	O	N	Erase All
M O V C	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 without this function

(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) In the OCDS mode, only when executing the OCDSINSTR instruction will the sectors be protected.

■ Security Type 2

Take the HT85F2240 as an example, its inhibit bytes SECURITY2[0:3] are located at the addresses 0x10~0x0B of the ID block page 0. If a value, with the exception of 0FFH, is written to SECURITY2[N], this sector can not be programmed, erased or read when in any mode.

The following table illustrates the protection state in the OCDS/ICP/IAP/MOVC modes when the SECURITY2[0:3] bytes are written with a value other than 0FFH:

SECURITY2[N] N=0~3		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
M O V C	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 without this function.

(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) In the OCDS mode, only when executing the OCDSINSTR instruction will the sectors be protected.

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

HT85F2240 Program Memory Contents

The HT85F2240 program memory is divided into 4 sectors, each with a capacity of 1k bytes.

Page	Address	Description
0	0x00~0x03	SECURITY1[0]~ SECURITY1[3]
	0x04~0x07	Not used
	0x08~0x0B	SECURITY2[0]~ SECURITY2[3]
	0x0C~0x0F	Not used
	0x10~0x1F	Reserved
1	0x20~0x2F	Reserved
	0x30~0x3F	Reserved

HT85F2230 Program Memory Contents

The HT85F2230 program memory is divided into 2 sectors, each with a capacity of 1k bytes.

Page	Address	Description
0	0x00~0x01	SECURITY1[0]~ SECURITY1[1]
	0x02~0x07	Not used
	0x08~0x09	SECURITY2[0]~ SECURITY2[1]
	0x0A~0x0F	Not used
	0x10~0x1F	Reserved
1	0x20~0x2F	Reserved
	0x30~0x3F	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=0~3

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 been written to the SECURITY2[N] bytes, the corresponding sectors will be protected and can not 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	Sector N is not protected Can be erased and programmed. Can be read via flash control registers (IAP and OCDS ^(note1)) or the MOVC instructions.
0FFH	Other values except 0FFH	Sector N is inhibited from Programming/Erasing Can not be erased and programmed via ICP or flash control registers (OCDS ^(note1)). Can be erased and programmed via flash control registers (IAP). Can be read via flash control registers (IAP) or 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 via ICP or flash control registers (IAP and OCDS ^(note1)). Can not be read via ICP or flash control registers (IAP and OCDS ^(note1)) or the MOVC instructions.

- Note:** 1. Here "OCDS" stands for executing the OCDSINSTR instruction when in the OCDS mode.
2. Once SECURITY1[N] or SECURITY2[N] is protected, the only method to return to "unprotected state" is by "ICP Whole Chip Erase".

18

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.

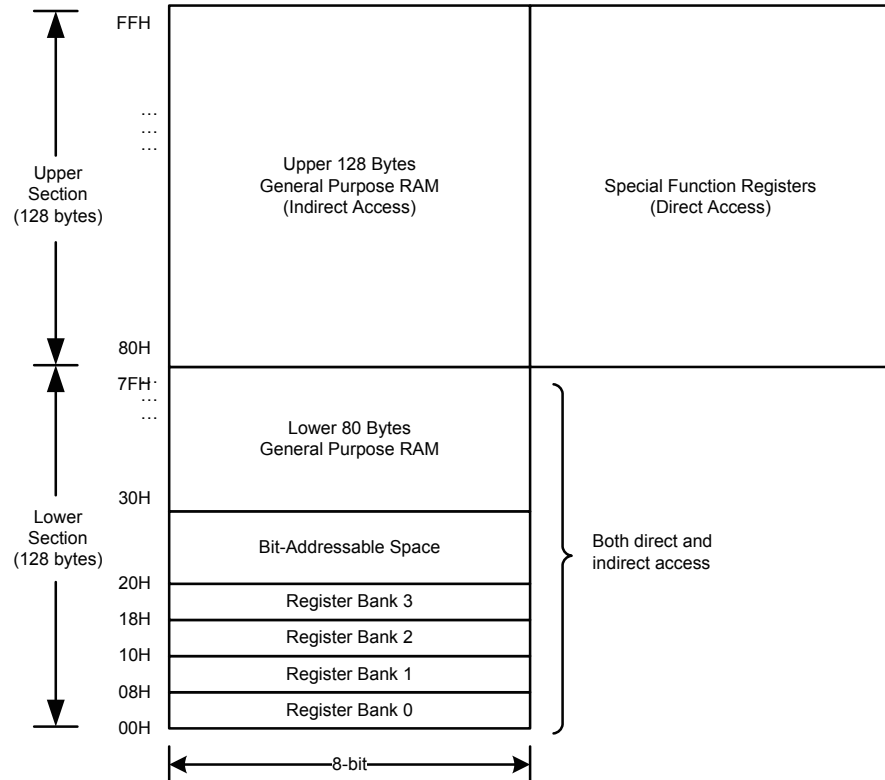
The upper section 128-byte RAM has an address range of 80H to FFH, and is assigned to both the General Purpose memory and the Special Function Registers. Although the address range is identical these two RAM sections are physically separate, they are distinguished by their different addressing methodology.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

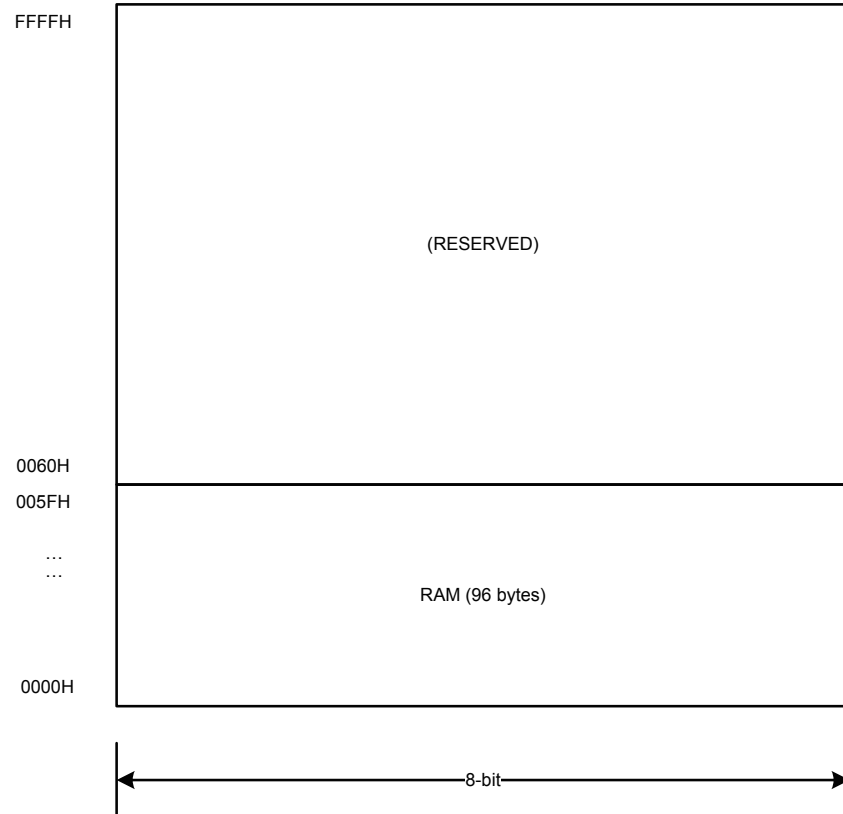
The XDATA is assigned as General Purpose Data RAM and can only be accessed using indirect addressing. The HT85F2240 and HT85F2230 have 96-bytes of XDATA.

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



HT85F2240/HT85F2230 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	—	—	—	—	—	—	—	—
E0h	ACC.0	ACC.1	ACC.2	ACC.3	ACC.4	ACC.5	ACC.6	ACC.7
D8h	—	—	—	—	—	—	—	—
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.6	—
C0h	—	—	IRCON.2	IRCON.3	—	—	IRCON.6	IRCON.7
B8h	IP0.0	IP0.1	IP0.2	IP0.3	—	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	—	IEN0.5	IEN0.6	IEN0.7
A0h	—	—	—	—	—	—	—	—
98h	—	—	—	—	—	—	—	—
90h	P1.0	P1.1	P1.2	P1.3	P1.4	—	—	—
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. “—” stands 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	—	—	LVRCR	LVDCR	SCCR	—	LSOCR	HSOCR
E0h	ACC	—	FMSR	—	IP1	IP1H	IP2	IP2H
D8h	—	—	—	—	—	—	—	HFSCR
D0h	PSW	—	—	—	—	PRM	—	—
C8h	T2CON	IEN3	CRCL	CRCH	TL2	TH2	IP3	IP3H
C0h	IRCON	CCEN	CCL1	CCH1	—	—	—	—
B8h	IP0	IP0H	—	—	—	—	—	IRCON2
B0h	P3	—	TBCR	—	—	—	P3M0	P3M1
A8h	IEN0	IEN1	—	—	—	—	P2M0	P2M1
A0h	P2	—	—	—	SRCR	—	P1M0	P1M1
98h	—	—	IEN2	—	—	—	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

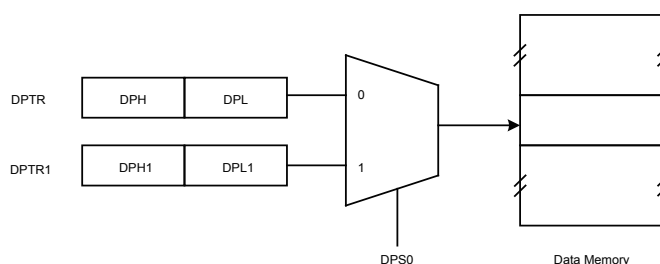
The B register is used as a general purpose register for these devices. It is used during multiplying and division instructions.

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, DPTR and DPTR1. 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 “MOVC 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” registers this field has to contain a different value pointing to itself, 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.

19 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. One external oscillator requiring some external components and 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	12/16/20MHz	—
Internal Low Speed RC	LIRC	WDT and Time Base Clock	32kHz	—

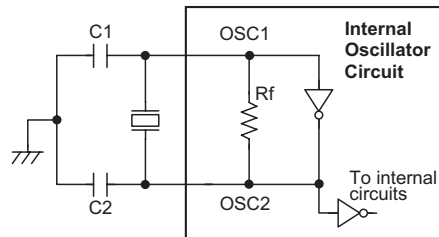
System Clock Configuration

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

External High Speed Crystal Oscillator – HXT

The External High Speed Crystal Oscillator is one of the high frequency oscillator choices, which is selected via the SCCR register. For most crystal oscillator configurations, 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.

For oscillator stability and to minimise the effects of noise and crosstalk, it is important to ensure that the crystal and any associated resistors and capacitors along with interconnecting lines are all located as close to the MCU as possible.



Crystal/Resonator Oscillator – HXT

Crystal Recommended Capacitor Values

Crystal Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
24MHz	10pF	10pF
20MHz	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 high speed RC oscillator is a fully integrated system oscillator requiring no external components. The internal RC oscillator has three frequencies of either 12MHz, 16MHz or 20MHz. 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. If the HIRC oscillator is used as the system oscillator, then the OSC1 and OSC2 pins should be left unconnected.

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 and requires no external components, however its actual frequency may vary with temperature and supply voltage.

20 Operating Modes and System Clocks

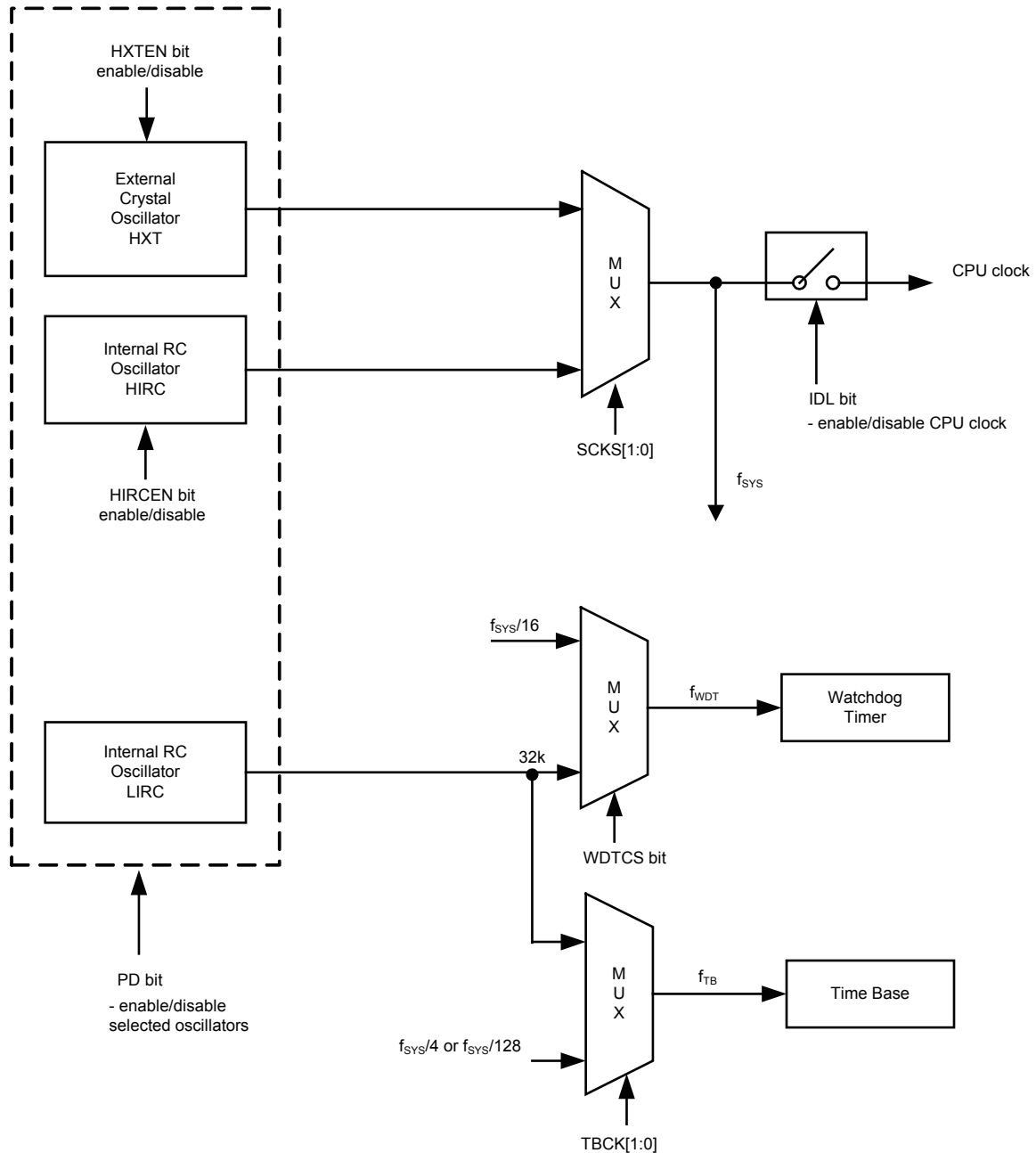
Present day applications require that their microcontrollers have high performance but often still demand that they consume as little power as possible, conflicting requirements that are especially important in battery powered portable applications. This usually requires the microcontroller can provide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

System Clocks Description

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

The MCU system clock is sourced from the high speed external crystal HXT oscillator or the internal HIRC oscillator. These oscillators can be used directly as the system clock.

The system clock, namely f_{SYS} , can also be used as a clock source for the peripheral functions, such as WDT, Time Base, Timers and ADC. 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 two sources. These are the internal HIRC oscillator or the external crystal HXT oscillator. 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 status 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 is an additional internal 32KHz low frequency clock for the peripheral circuits. It is the internal LIRC oscillator, which is enabled using the LIRCEN bit in the LSOCR register. There is a low frequency oscillator status bit, LIRCRDY, to indicate the “ready or not” status of the LIRC oscillator. This bit 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 by hardware when the LIRC is disabled and set high once the LIRC 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.

HIRC Frequency Select Control Register – HFSCR

SFR Address: DFh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	HFSCR1	HFSCR0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1~0 **HFSCR1~HFSCR0**: HIRC Frequency Selection bit

0x: 12MHz (default frequency)

10: 16MHz

11: 20MHz

Low Speed Oscillator Control Register – LSOCR

SFR Address: EEh

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

Bit 7~5 Unimplemented, read as “0”

Bit 4 **LIRCRDY**: LIRC oscillator ready indication flag

0: Not ready

1: Ready

This is the internal low frequency oscillator, LIRC, ready indication bit which indicates if the LIRC oscillator is stable or not. This bit will be automatically cleared to zero by hardware when the LIRC is disabled.

Bit 3~2 Unimplemented, read as “0”

Bit 1 **LIRCEN**: LIRC oscillator select bit

0: Enable

1: Disable

Bit 0 Unimplemented, read as “0”

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 and ADC. The slow speed oscillator will also continue to run and keep the WDT and Time Base functions active, if its clock source is 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 Internal RC Oscillator (LIRC)	On (LIRCEN=0) / Off	On (LIRCEN=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, PCA and ADC.

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 or HIRC.

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 source will also remain operational and can also provide a clock source for the WDT and Time Base functions, if it is enabled and if its 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/W	R/W	R/W
POR	0	—	—	—	1	0	0	0

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

Bit 6~4 Unimplemented, read as “0”

Bit 3 Unimplemented, read as “1”

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 enters 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:

■ In Idle mode:

- An external reset
- An external low level on any P0 I/O pin
- An external low level on INT0 and INT1 pins
- A system interrupt
- A WDT overflow

A system interrupt wake up can be generated by various peripheral interrupts, such as Timer 0~2 interrupt, ADC interrupt, Time base interrupt and LVD interrupt.

■ In Power down mode:

- An external reset
- An external low level on any P0 I/O pin
- An external low level on INT0 and INT1 pins

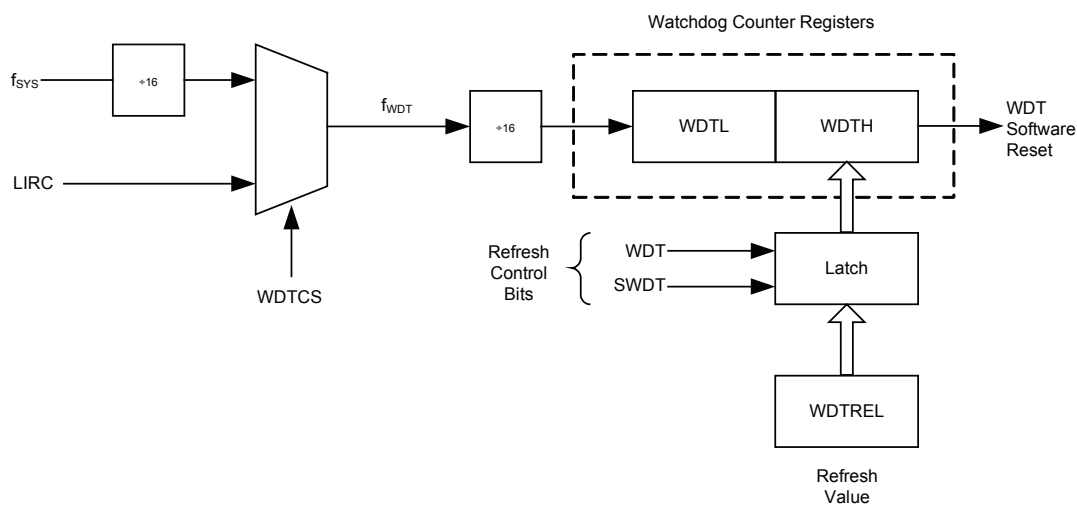
If the system is woken up by an external reset, the device will experience a full system reset, however, if the device is woken up by a WDT overflow, a Watchdog Timer reset will be initiated.

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 bit is 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 high before entering the IDLE or Power-Down mode, 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.

21 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 WDTCSR 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

Register Name	Bit							
	7	6	5	4	3	2	1	0
IEN0	(EAL)	WDT	(ET2)	—	(ET1)	(EX1)	(ET0)	(EX0)
IEN1	(EXEN2)	SWDT	—	—	—	—	(ECCU1)	(ECCU0)
WDTREL	D7	D6	D5	D4	D3	D2	D1	D0
WDTCSR	WE4	WE3	WE2	WE1	WE0	—	—	WDTCS
IP0	—	WDTS	(PT2)	—	(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	—	ET1	EX1	ET0	EX0
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 **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. The WDT bit must be set directly 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:** Timer 2 interrupt enable
Described elsewhere
- Bit 4 Unimplemented, read as “0”
- Bit 3 **ET1:** Timer 1 overflow interrupt enable
Described elsewhere
- Bit 2 **EX1:** External interrupt 1 enable
Described elsewhere
- Bit 1 **ET0:** Timer 0 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	—	—	—	—	ECCU1	ECCU0
R/W	R/W	R/W	—	—	—	—	R/W	R/W
POR	0	0	—	—	—	—	0	0

Bit 7 **EXEN2:** Timer 2 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 is immediately cleared by hardware. This bit is always read as “0”.

Bit 5~2 Unimplemented, read as “0”

Bit 1 **ECCU1:** CCU1 interrupt enable

Described elsewhere

Bit 0 **ECCU0:** CCU0 interrupt 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}) selection

0: LIRC

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	—	PT1	PX1	PT0	PX0
R/W	—	R/W	R/W	—	R/W	R/W	R/W	R/W
POR	—	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 low

Described elsewhere

Bit 4 Unimplemented, read as “0”

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 two sources selected by the WDTCS bit in the WDTCR register: a 32KHz clock or $f_{\text{SYS}}/16$. The 32KHz clock is sourced from the LIRC oscillator. 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 other Watchdog Timer clock source option is the $f_{\text{SYS}}/16$ clock.

Watchdog Timer Operation

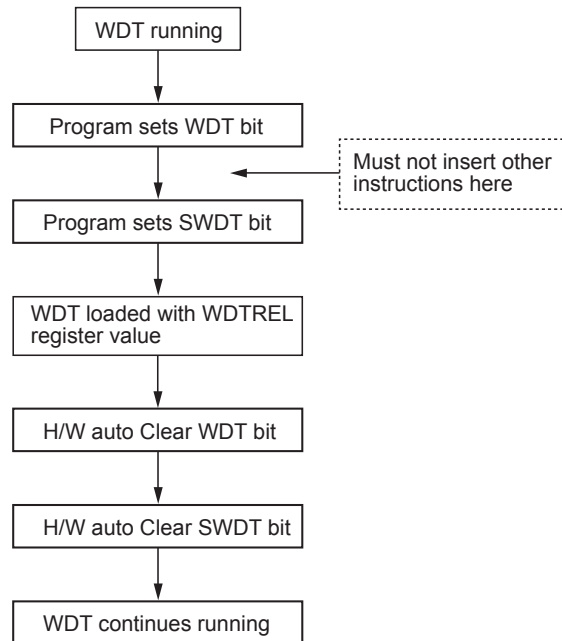
The Watchdog Timer operates by providing a device reset when its 16-bit timer overflows. The WDT is formed of two 8-bit registers, WDTL and WDTM, both of which are inaccessible to the application program. The WDTM register of the Watchdog Timer is reloaded with the contents of the WDTREL register. In 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

22 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.

23 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	RESET	XRSTF	RSTSRC	Hardware Reset
3	Low-Voltage Reset	LVR	LVRF	RSTSRC	Low V_{DD} voltage
4	LVRCCR Register Setting Software Reset	—	LRF	RSTSRC	Write to LVRCCR register
5	Watchdog Reset	WDT	WDTS	IP0	Watchdog overflow
6	WDTCR Register Setting Software Reset	—	WRF	RSTSRC	Write to WDTCR register
7	SRST Register Setting Software Reset	—	SRSTREQ	SRST	Write to SRST register

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. Six 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.

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

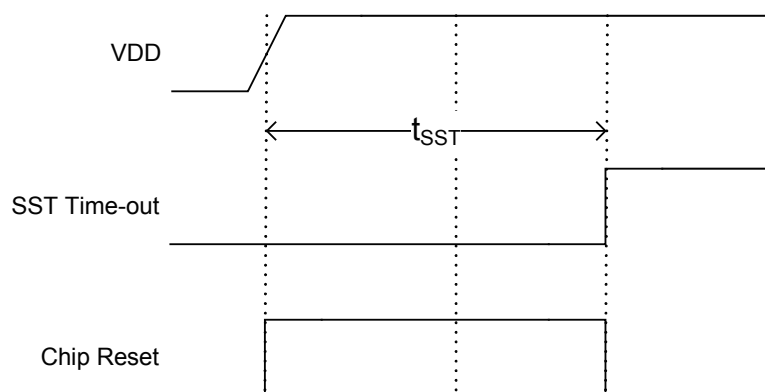
Bit	7	6	5	4	3	2	1	0
Label	—	LRF	WRF	—	—	LVRF	XRSTF	PORF
R/W	—	R	R	—	—	R	R	R
POR	—	0	0	—	—	x	0	1

- Bit 7 Unimplemented, read as “0”
- Bit 6 **LRF:** LVRCCR Register Setting Software Reset Indication Flag
 0: No LVRCCR Setting Software Reset
 1: LVRCCR Software Reset
- Bit 5 **WRF:** WDTCR Register Setting Software Reset Indication Flag
 0: No WDTCR Setting Software Reset
 1: WDTCR Setting Software Reset
- Bit 4~3 Unimplemented, read as “0”
- 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.

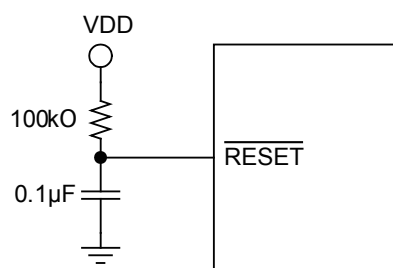
Although the microcontroller has an internal RC reset function, if the VDD power supply rise time is not fast enough or does not stabilise quickly at power-on, the internal reset function may be incapable of providing proper reset operation. For this reason it is recommended that an external RC network 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.



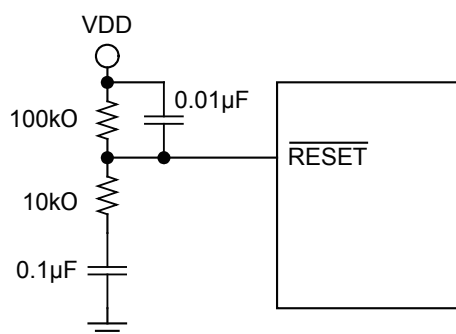
Power-On Reset Timing

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 $0.9V \sim V_{LVR}$ 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 high. For a valid LVR signal, a low voltage, i.e., a voltage in the range between $0.9V \sim V_{LVR}$ 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 LVS_n bits in the LVRCR register. If the LVS7~LVS0 bits are changed to some different 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 high. After power on the register will have the value of 00110011B. 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	0	1	1	0	0	1	1

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

01010101: 2.1V

00110011: 2.55V (default)

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 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	—	PT1	PX1	PT0	PX0
R/W	—	R/W	R/W	—	R/W	R/W	R/W	R/W
POR	—	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 Unimplemented, read as “0”

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

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 LVRCCR register.

Software Reset Summary

Software Reset Name	Operation Objects		Operation
	Register	Bit	
SRST Register setting	SRST	SRSTREQ	Write two successive “1” values to this bit
WDTCR Register setting	WDTCR	WE4~WE0	Write value other than “10101” or “01010”
LVRCCR Register setting	LVRCCR	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 last 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}) selection

Described elsewhere

LVRCCR Register Software Reset

An LVRCCR software reset will be generated when a value other than “01010101”, “00110011”, “10011001” and “10101010”, exist in the LVRCCR register. The LRF bit in the RSTSRC register will be set high when this occurs, thus indicating the generation of a LVRCCR software reset. The LVRCCR register value will be reset to a value of 00110011B after any reset other than the LVR reset, and will remain unchanged after an LVR reset.

LVRCCR 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	0	1	1	0	0	1	1

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

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
WDTRREL	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	--00_0000b	--00_0000b	--00_0000b	--00_0000b
P1	---1_1111b	---1_1111b	---1_1111b	---1_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
IEN2	----_--0-b	----_--0-b	----_--0-b	----_--0-b
P0M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P0M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
SRCR	----_0000b	----_0000b	----_0000b	----_0000b
SPPRE	----_--11b	----_--11b	----_--11b	----_--11b
P1M0	---0_0000b	---0_0000b	---0_0000b	---0_0000b
P1M1	---0_0000b	---0_0000b	---0_0000b	---0_0000b
IEN0	000-_0000b	000-_0000b	000-_0000b	000-_0000b
IEN1	00--_--00b	00--_--00b	00--_--00b	00--_--00b
P3	1111_1111b	1111_1111b	1111_1111b	1111_1111b
TBCR	0-00_-111b	0-00_-111b	0-00_-111b	0-00_-111b
P3M0	0000_0000b	0000_0000b	0000_0000b	0000_0000b
P3M1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
IP0	-00-_0000b	-00-_0000b	-10-_0000b	-00-_0000b
IP0H	--0-_0000b	--0-_0000b	--0-_0000b	--0-_0000b
IRCON2	----_00-0b	----_00-0b	----_00-0b	----_00-0b
IRCON	00--_00--b	00--_00--b	00--_00--b	00--_00--b
CCEN	----_0000b	----_0000b	----_0000b	----_0000b
CCL1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
CCH1	0000_0000b	0000_0000b	0000_0000b	0000_0000b
T2CON	-0-0_0000b	-0-0_0000b	-0-0_0000b	-0-0_0000b

Register Name	Power-On Reset	External Reset	WDT Time-out Reset	Software Reset
IEN3	----_00--b	----_00--b	----_00--b	----_00--b
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	----_00--b	----_00--b	----_00--b	----_00--b
IP3H	----_00--b	----_00--b	----_00--b	----_00--b
PSW	0000_0000b	0000_0000b	0000_0000b	0000_0000b
PRM	----_---0b	----_---0b	----_---0b	----_---0b
HFSCR	----_---00b	----_---00b	----_---00b	----_---00b
ACC	0000_0000b	0000_0000b	0000_0000b	0000_0000b
FMSR	0---_0000b	0---_0000b	0---_0000b	0---_0000b
IP1	----_---00b	----_---00b	----_---00b	----_---00b
IP1H	----_---00b	----_---00b	----_---00b	----_---00b
IP2	----_---0-b	----_---0-b	----_---0-b	----_---0-b
IP2H	----_---0-b	----_---0-b	----_---0-b	----_---0-b
LVRCCR	0011_0011b	0011_0011b	0011_0011b	0011_0011b
LVDCR	0---_---000b	0---_---000b	0---_---000b	0---_---000b
SCCR	----_---00b	----_---00b	----_---00b	----_---00b
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_-x01b	-000_-010b	-000_-000b	-000_-000b

Register Name	WDTCR Reset	LVR Reset	LVRCCR Reset
Program Counter	0000h	0000h	0000h
P0	1111_1111b	1111_1111b	1111_1111b
SP	0000_0111b	0000_0111b	0000_0111b
DPL	0000_0000b	0000_0000b	0000_0000b
DPH	0000_0000b	0000_0000b	0000_0000b
DPL1	0000_0000b	0000_0000b	0000_0000b
DPH1	0000_0000b	0000_0000b	0000_0000b
WDTREL	0000_0000b	0000_0000b	0000_0000b
PCON	0---_1000b	0---_1000b	0---_1000b
TCON	0000_0000b	0000_0000b	0000_0000b
TMOD	0000_0000b	0000_0000b	0000_0000b
TL0	0000_0000b	0000_0000b	0000_0000b
TL1	0000_0000b	0000_0000b	0000_0000b
TH0	0000_0000b	0000_0000b	0000_0000b
TH1	0000_0000b	0000_0000b	0000_0000b
TMPRE	--00_0000b	--00_0000b	--00_0000b
P1	---1_1111b	---1_1111b	---1_1111b
P0WAKE	0000_0000b	0000_0000b	0000_0000b
DPS	----_---0 b	----_---0 b	----_---0 b
DPC	----_0000b	----_0000b	----_0000b
WDTCR	0101_0--0b	0101_0--0b	0101_0--0b
IEN2	----_--0-b	----_--0-b	----_--0-b
P0M0	0000_0000b	0000_0000b	0000_0000b
P0M1	0000_0000b	0000_0000b	0000_0000b
P2	----_---1 b	----_---1 b	----_---1 b
SRCCR	----_0000b	----_0000b	----_0000b
P1M0	---0_0000b	---0_0000b	---0_0000b
P1M1	---0_0000b	---0_0000b	---0_0000b
IEN0	000-_0000b	000-_0000b	000-_0000b
IEN1	00--_--00b	00--_--00b	00--_--00b
P3	1111_1111b	1111_1111b	1111_1111b
TBCR	0-00_-111b	0-00_-111b	0-00_-111b
P3M0	0000_0000b	0000_0000b	0000_0000b
P3M1	0000_0000b	0000_0000b	0000_0000b
IP0	-00-_0000b	-00-_0000b	-00-_0000b
IP0H	--0-_0000b	--0-_0000b	--0-_0000b
IRCON2	----_00-0b	----_00-0b	----_00-0b
IRCON	00--_00--b	00--_00--b	00--_00--b
CCEN	----_0000b	----_0000b	----_0000b
CCL1	0000_0000b	0000_0000b	0000_0000b
CCH1	0000_0000b	0000_0000b	0000_0000b
T2CON	-0-0_0000b	-0-0_0000b	-0-0_0000b
IEN3	----_00--b	----_00--b	----_00--b

Register Name	WDTCR Reset	LVR Reset	LVRCCR Reset
CRCL	0000_0000b	0000_0000b	0000_0000b
CRCH	0000_0000b	0000_0000b	0000_0000b
TL2	0000_0000b	0000_0000b	0000_0000b
TH2	0000_0000b	0000_0000b	0000_0000b
IP3	----_00--b	----_00--b	----_00--b
IP3H	----_00--b	----_00--b	----_00--b
PSW	0000_0000b	0000_0000b	0000_0000b
PRM	----_---0b	----_---0b	----_---0b
HFSCR	----_--00b	----_--00b	----_--00b
ACC	0000_0000b	0000_0000b	0000_0000b
FMSR	0---_0000b	0---_0000b	0---_0000b
IP1	----_--00b	----_--00b	----_--00b
IP1H	----_--00b	----_--00b	----_--00b
IP2	----_--0-b	----_--0-b	----_--0-b
IP2H	----_--0-b	----_--0-b	----_--0-b
LVRCCR	0011_0011b	uuuu_uuuub	0011_0011b
LVDCR	0---_000b	0---_000b	0---_000b
SCCR	----_--00b	----_--00b	----_--00b
LSOCR	---1_--0-b	---1_--0-b	---1_--0-b
HSOCR	--01_--01b	--01_--01b	--01_--01b
B	0000_0000b	0000_0000b	0000_0000b
ADCR0	0110_0000b	0110_0000b	0110_0000b
ADCR1	00-0_0000b	00-0_0000b	00-0_0000b
ADCR2	0000_0000b	0000_0000b	0000_0000b
ADPGA	----_000b	----_000b	----_000b
ADRL(ADRFS=0)	0000_----b	0000_----b	0000_----b
ADRH(ADRFS=0)	0000_0000b	0000_0000b	0000_0000b
SRST	----_---0b	----_---0b	----_---0b
FMCR	01--_000b	01--_000b	01--_000b
FMKEY	0000_0000b	0000_0000b	0000_0000b
FMAR0	0000_0000b	0000_0000b	0000_0000b
FMAR1	0000_0000b	0000_0000b	0000_0000b
FMAR2	0000_0000b	0000_0000b	0000_0000b
FMDR	0000_0000b	0000_0000b	0000_0000b
T2CON1	---u_10--b	---u_10--b	---u_10--b
RSTSRC	-010_-000b	-000_-100b	-100_-000b

Note: "-" not implement

"u" stands for "unchanged"

"x" stands for "unknown"

24 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, LVD 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	—	—
INTn Pin	EXn	IEn	n=0~1
A/D Converter	EADC	IADC	—
Time Base	ETB	TBF	—
LVD	ELVD	LVDF	—
Timer n	ETn	TFn	n=0~2
CCUn	ECCUn	CCUnF	n=0~1
Timer 2 External Reload	EXEN2	EXF2	—

Interrupt Register Contents

Register Name	Bit							
	7	6	5	4	3	2	1	0
IEN0	EAL	(WDT)	ET2	—	ET1	EX1	ET0	EX0
IEN1	EXEN2	(SWDT)	—	—	—	—	ECCU1	ECCU0
IEN2	—	—	—	—	—	—	ELVD	—
IEN3	—	—	—	—	ETB	EADC	—	—
IRCON	EXF2	TF2	—	—	CCU1F	CCU0F	—	—
IRCON2	—	—	—	—	LVDF	TBF	—	IADC
TCON	TF1	(TR1)	TF0	(TR0)	IE1	IT1	IE0	IT0
T2CON	—	I3FR	—	(T2R1)	(T2R0)	(T2CM)	(T2I1)	(T2I0)

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	—	ET1	EX1	ET0	EX0
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 **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 Unimplemented, read as “0”

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	—	—	—	—	ECCU1	ECCU0
R/W	R/W	R/W	—	—	—	—	R/W	R/W
POR	0	0	—	—	—	—	0	0

Bit 7 **EXEN2:** Timer 2 external reload interrupt enable

0: Disable

1: Enable

When EXEN2=0, Timer 2 external reload interrupt is disabled. When EXEN2=1, EAL=1 and ET2=1, Timer 2 external reload interrupt 2 is enabled.

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

Described elsewhere

Bit 5~2 Unimplemented, read as “0”

Bit 1 **ECCU1:** CCU1 interrupt enable

0: Disable

1: Enable

Bit 0 **ECCU0:** CCU0 interrupt enable

0: Disable

1: Enable

IEN2 Register

SFR Address: 9Ah

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

Bit 7~2 Unimplemented, read as “0”

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

Bit 0 Unimplemented, read as “0”

IEN3 Register

SFR Address: C9h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	ETB	EADC	—	—
R/W	—	—	—	—	R/W	R/W	—	—
POR	—	—	—	—	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 1~0 Unimplemented, read as “0”

IRCON Register SFR Address: C0h

Bit	7	6	5	4	3	2	1	0
Name	EXF2	TF2	—	—	CCU1F	CCU0F	—	—
R/W	R/W	R/W	—	—	R/W	R/W	—	—
POR	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 5~4** Unimplemented, read as “0”
- Bit 3** **CCU1F:** CCU1 interrupt request flag
 0: No request
 1: Interrupt request
 This bit 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). Additionally, this bit can be set high when Timer 2 Capture mode 0 is enabled and a rising edge is detected on CC1 input pin. Once the program into the interrupt subroutine, the CCU1F flag will be cleared by hardware automatically.
- Bit 2** **CCU0F:** CCU0 interrupt request flag
 0: No request
 1: Interrupt request
 This bit will be set high when Timer 2 compare mode is enabled and counter value (TH2, TL2) is equal to Compare/Reload/Capture register 3 (CRCH, CRCL). Additionally, this bit can be set high when Timer 2 Capture mode 0 is enabled and a rising or falling edge (depending on the I3FR bit) is detected on CC0 input pin. Once the program into the interrupt subroutine, the CCU0F flag will be cleared by hardware automatically.
- Bit 1~0** Unimplemented, read as “0”

T2CON Register

SFR Address: C8h

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

- Bit 7 Unimplemented, read as “0”
- Bit 6 **I3FR**: Timer 2 capture mode 0 edge selection for “CC0”
0: Falling edge
1: Rising edge
This bit is used as capture signal in CC0. When Timer 2 is selected as compare mode 0, for compare interrupt flag active high timing consideration, the I3FR bit is recommended to be set high by firmware.
- Bit 5 Unimplemented, read as “0”
- 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

IRCON2 Register

SFR Address: BFh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	LVDF	TBF	—	IADC
R/W	—	—	—	—	R/W	R/W	—	R/W
POR	—	—	—	—	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 Unimplemented, read as “0”
- Bit 0 **IADC**: ADC interrupt request flag
0: No request
1: Interrupt request
This bit will be cleared by hardware automatically.

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

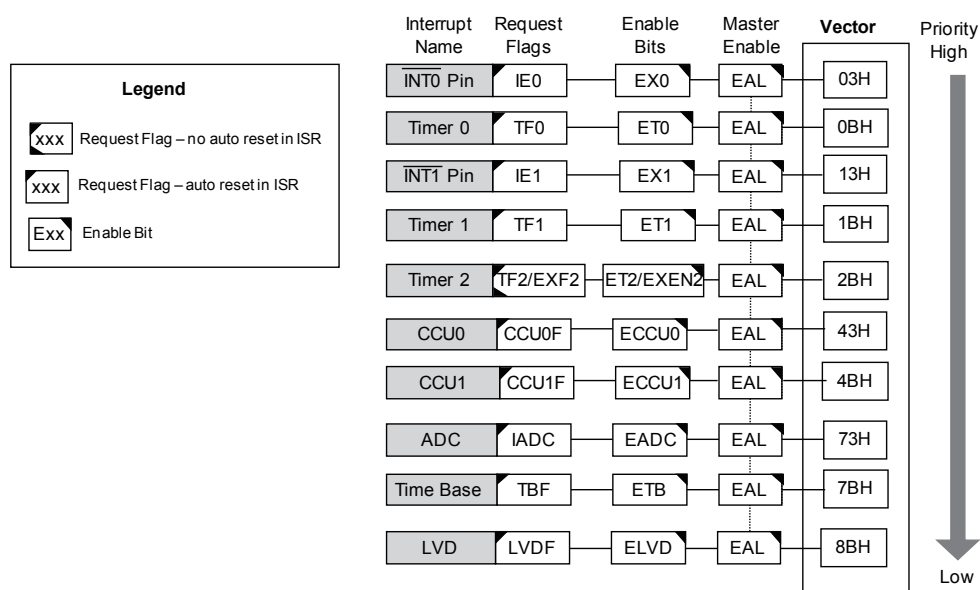
- Bit 7 **TF1:** Timer 1 interrupt request flag
 0: No request
 1: Interrupt request
 This bit will be cleared by hardware automatically when the interrupt is processed.
- Bit 6 **TR1:** Timer 1 Run control
 Described elsewhere
- Bit 5 **TF0:** Timer 0 interrupt request flag
 0: No request
 1: Interrupt request
 This bit will be cleared by hardware automatically when the interrupt is processed.
- 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 is set by hardware when the external interrupt $\overline{\text{INTI}}$ (edge, depending on settings) is observed. This bit will be cleared by hardware when the interrupt is processed.
 When the external interrupt $\overline{\text{INTI}}$ (level, depending on settings) is observed, IE1 is the inverse level of the $\overline{\text{INTI}}$.
- Bit 2 **IT1:** External interrupt 1 type control
 0: Low Level
 1: Falling Edge
- Bit 1 **IE0:** External interrupt 0 request flag
 0: No request
 1: Interrupt request
 This bit is set by hardware when the external interrupt $\overline{\text{INT0}}$ (edge, depending on settings) is observed. This bit will be cleared by hardware when the interrupt is processed.
 When the external interrupt $\overline{\text{INT0}}$ (level, depending on settings) is observed, IE0 is the inverse level of the $\overline{\text{INT0}}$.
- Bit 0 **IT0:** External interrupt 0 type control
 0: Low Level
 1: Falling Edge

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 EAL is used to control the whole interrupt function enable or disable and it has to be cleared using the application program.

The various interrupt enable bits, together with their associated request flags, are shown in the following diagram with their order of priority. Interrupts with higher priority can stop lower priority ones. All interrupts are categorised into four priority levels.



Interrupt Structure

Note that if one interrupt requests occur during the other interrupt interval, although the interrupt will not be immediately serviced, the request flag will still be recorded and the next interrupt request will take place according to the interrupt priority. 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 Priority

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. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority levels are pending at the end of an instruction, the request of higher priority level is serviced. If requests of the same priority level are pending at the end of an instruction, an internal polling sequence determines which 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 pending 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

With regard to interrupts which are assigned at the same priority level by the IPnH.x and IPn.x bits, their actual interrupt priority should be followed by the accompanying table.

Interrupt Source	Interrupt Vector	Priority Control	Service Priority
Reset	0000H	Always Highest	Highest Priority
External Interrupt 0 (INT0)	0003H	PX0H(IP0H.0) / PX0(IP0.0)	↓
Timer 0 Overflow Interrupt	000BH	PT0H(IP0H.1) / PT0(IP0.1)	↓
External Interrupt 1 (INT1)	0013H	PX1H(IP0H.2) / PX1(IP0.2)	↓
Timer 1 Overflow Interrupt	001BH	PT1H(IP0H.3) / PT1(IP0.3)	↓
Timer 2 Overflow Interrupt or Timer 2 External Reload Interrupt	002BH	PT2H(IP0H.5) / PT2(IP0.5)	↓
CCU0 Comapre Interrupt	0043H	PCCU0H(IP1H.0) / PCCU0(IP1.0)	↓
CCU1 Comapre Interrupt	004BH	PCCU1H(IP1H.1) / PCCU1(IP1.1)	↓
ADC End of Conversion Interrupt	0073H	PADCH(IP3H.2) / PADC(IP3.2)	↓
Time Base Overflow Interrupt	007BH	PTBH(IP3H.3) / PTB(IP3.3)	↓
LVD Interrupt	008BH	PLVDH(IP2H.1) / PLVD(IP2.1)	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	—	PT1	PX1	PT0	PX0
R/W	—	R/W	R/W	—	R/W	R/W	R/W	R/W
POR	—	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 Unimplemented, read as “0”
- 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 Interrupt 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 Interrupt 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	—	PT1H	PX1H	PT0H	PX0H
R/W	—	—	R/W	—	R/W	R/W	R/W	R/W
POR	—	—	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 Unimplemented, read as “0”
- 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 Interrupt 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 Interrupt 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	—	—	—	—	—	—	PCCU1	PCCU0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1 **PCCU1**: CCU1 Interrupt priority low
Low order bit for CCU1 interrupt priority level.

Bit 0 **PCCU0**: CCU0 Interrupt priority low
Low order bit for CCU0 interrupt priority level.

High byte of Interrupt Priority Register 1: IP1H

SFR Address: E5h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PCCU1H	PCCU0H
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1 **PCCU1H**: CCU1 Interrupt priority high
High order bit for CCU1 interrupt priority level.

Bit 0 **PCCU0H**: CCU0 Interrupt priority high
High order bit for CCU0 interrupt priority level.

Low byte of Interrupt Priority Register 2: IP2

SFR Address: E6h

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

Bit 7~2 Unimplemented, read as “0”

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

Bit 0 Unimplemented, read as “0”

High byte of Interrupt Priority Register 2: IP2H

SFR Address: E7h

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

Bit 7~2 Unimplemented, read as “0”

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

Bit 0 Unimplemented, read as “0”

Low byte of Interrupt Priority Register 3: IP3

SFR Address: CEh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PTB	PADC	—	—
R/W	—	—	—	—	R/W	R/W	—	—
POR	—	—	—	—	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 1~0 Unimplemented, read as “0”

How byte of Interrupt Priority Register 3: IP3H

SFR Address: CFh

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PTBH	PADCH	—	—
R/W	—	—	—	—	R/W	R/W	—	—
POR	—	—	—	—	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 1~0 Unimplemented, read as “0”

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 low level or falling edge 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 but the EAL bit must be cleared by the application program to disable other interrupts.

The IT0 and IT1 bits are used to select the active edge or level that will trigger the external interrupt for $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ respectively.

External Interrupt Trigger Type

Ext Int	Trigger Type	Register	Bit
$\overline{\text{INT0}}$	Low Level or Falling Edge	TCON	IT0
$\overline{\text{INT1}}$	Low Level or Falling Edge	TCON	IT1

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

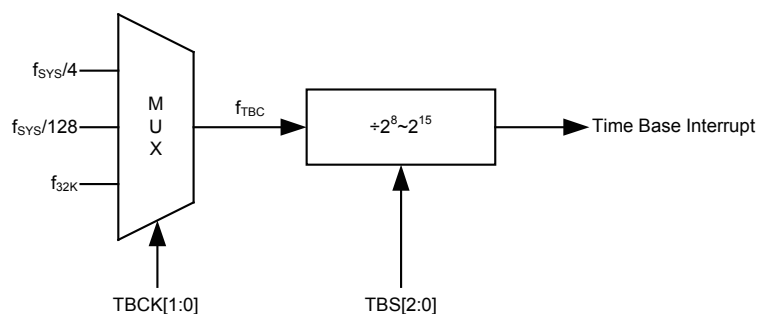
For a Timer Counter interrupt to occur, the global interrupt enable bit, EAL, and the corresponding timer interrupt enable bit, ETn, must first be set. An actual Timer Counter interrupt will take place when the Timer Counter request flag, TFn, is set, a situation that will occur when the relevant Timer 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), 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.

In addition, when the Timer 2 works in relode mode 1, a falling edge at the T2EX input pin will reload the data from CRCH/CRCL registers to TH2/TL2 registers, if the external reload interrupt control bit, EXEN2, the Timer 2 interrupt control bit, ET2, and the global interrupt enable bit, EAL, are all set high, a Timer 2 external reload interrupt will generated. When the interrupt is serviced, the interrupt flag, EXF2, must be cleared using the application program.

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 happens 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 or the LIRC oscillator.



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**: Time Base 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})
- 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)

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 must be cleared by the application program 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 among these interrupts only $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ 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, or a low power supply voltage 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 $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ 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.

25 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 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, P1 and P3. 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, P1 and P3 can be setup using Port Mode Registers to operate in a series of different modes. The Port P0 provides register controlled wake up function as well. Bit manipulation instructions can be used to control Ports P0, P1 and P3, and all pins for these ports are equipped with slew rate control.

I/O Port Function Summary

Function.	Port Number			Notes
	P0	P1	P3	
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	—	—	—	P1.4	P1.3	P1.2	P1.1	P1.0
P1M0	—	—	—	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0
P1M1	—	—	—	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
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
SRCR	—	—	—	—	SRCR.3	SRCR.2	SRCR.1	SRCR.0

Each Port has its own data register, known as P0, P1 and P3, 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 write 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

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

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

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.

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	—	—	—	—	—	—	—	P2M0.0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

P2M1 Register

SFR Address: AFh

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

Note: 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.3	SRCR.2	SRCR.1	SRCR.0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as “0”

Bit 3 **SRCR.3:** Port 3 Slew Control Enable

0: Fast

1: Slow

Bit 2 **SRCR.2:** Port 2 Slew Control Enable

0: Fast

1: Slow

Bit 1 **SRCR.1:** Port 1 Slew Control Enable

0: Fast

1: Slow

Bit 0 **SRCR.0:** Port 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.

Pin-remapping Function

The limited number of supplied pins in a package can impose restrictions on the amount of functions a certain device can contain. However by allowing the same pins to share several different functions and providing a means of function selection, a wide range of different functions can be incorporated into even relatively small package sizes.

PRM Register – Pin-remapping Control Register

PRM Register

SFR Address: D5h

Bit	7	6	5	4	3	2	1	0
Label	—	—	—	—	—	—	—	PRM0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 Unimplemented, read as “0”

Bit 0 **PRM0:** CC3~CC0 pin-remapping selection bit

0: CC1 on P3.1, CC0 on P3.0

1: CC1 on P1.0, CC0 on P0.7

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

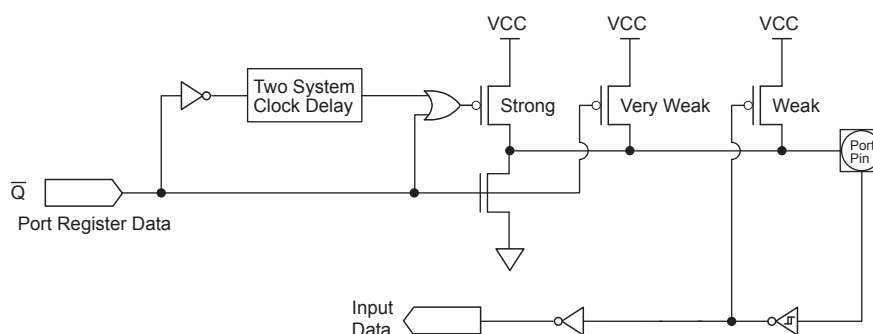
This is the traditional 8051 type I/O port type, constructed from an NMOS FET transistor and three pull high resistors, so called Strong, Weak, Very Weak pull high resistors. This structure can be used to reduce the power consumption and the output switching state respond time.

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 register bits have 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 resistors 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.

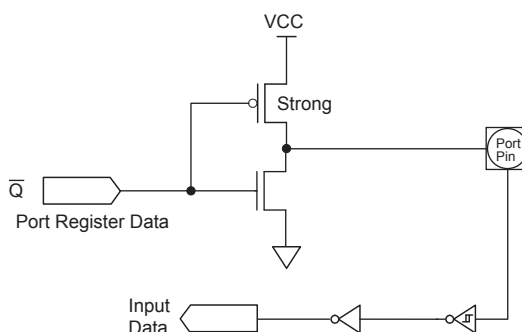
A Quasi-bidirectional pin also provides a Schmitt Trigger input.



Quasi-bidirectional I/O Structure

Push-pull Output

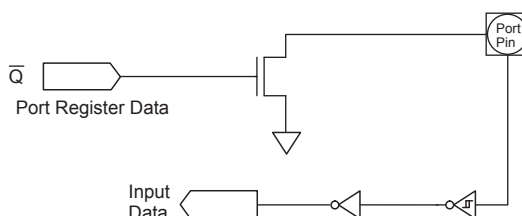
This I/O structure is a standard CMOS type structure with a single NMOS and PMOS complimentary transistor pair. The input is a Schmitt Trigger type input.



Push-pull Output Structure

Open-drain Output

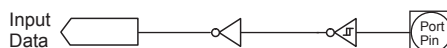
This I/O structure is an open drain type structure with a Schmitt Trigger input. Usually, an external pull high resistor is needed for such applications.



Open-drain Output Structure

Input Only

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, P1 and P3 provide four I/O structure modes option. 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, P1 and P3, 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 SETB 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 P1, A”, while others can operate directly onto the external port input, such as “MOV A, P1”.

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	√
CLR Px.y	Clear Bit y of Port x	CLR P3.0	√
SETB Px.y	Set Bit y of Port x	SETB P3.0	√

26 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 three Timers, namely Timer 0, Timer 1 and Timer 2. Each individual Timer is 16-bit wide which are composed of two 8-bit registers, TLn and THn. Timers 0 and Timer 1 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 as 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 and 1 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
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	—	—	√

27 Timer/Event Counters 0, 1

These two timers provide have a similar type and structure and operate with a choice of three modes for Timer 1 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.

Timer Mode	TnM1, TnM0 bits	Mode Name	Application Timer
0	00	13-bit Timer-Counter	Timer 0, 1
1	01	16-bit Counter	Timer 0, 1
2	10	8-bit Counter Auto Reload	Timer 0, 1
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 Register Description

Overall operation of the Timer 0 and Timer 1 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 Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
IEN0	EAL	WDT	ET2	—	ET1	EX1	ET0	EX0
IEN1	EXEN2	SWDT	—	—	—	—	ECCU1	ECCU0
IRCON	EXF2	TF2	—	—	CCU1F	CCU0F	—	—
TMOD	GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TLn	D7	D6	D5	D4	D3	D2	D1	D0
THn	D15	D14	D13	D12	D11	D10	D9	D8
TMPRE	—	—	T2PRE1	T2PRE0	T1PRE1	T1PRE0	T0PRE1	T0PRE0

Note: n=0, 1

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

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 $\overline{\text{INT1}}$ 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 $\overline{\text{INT0}}$ 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 when the interrupt is processed.
- 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 when the interrupt is processed.
- 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

TMPRE Register SFR Address: 8Fh

Bit	7	6	5	4	3	2	1	0
Name	—	—	T2PRE1	T2PRE0	T1PRE1	T1PRE0	T0PRE1	T0PRE0
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 5~4 **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 $\overline{INTn}$ 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

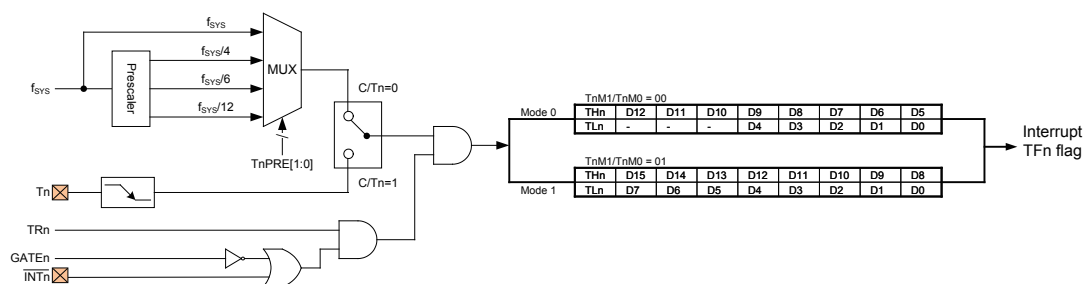
Register Name	Bit							
	7	6	5	4	3	2	1	0
THn	D12	D11	D10	D9	D8	D7	D6	D5
TLn	—	—	—	D4	D3	D2	D1	D0

Note: n=0, 1

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 $\overline{\text{INTn}}$ 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.

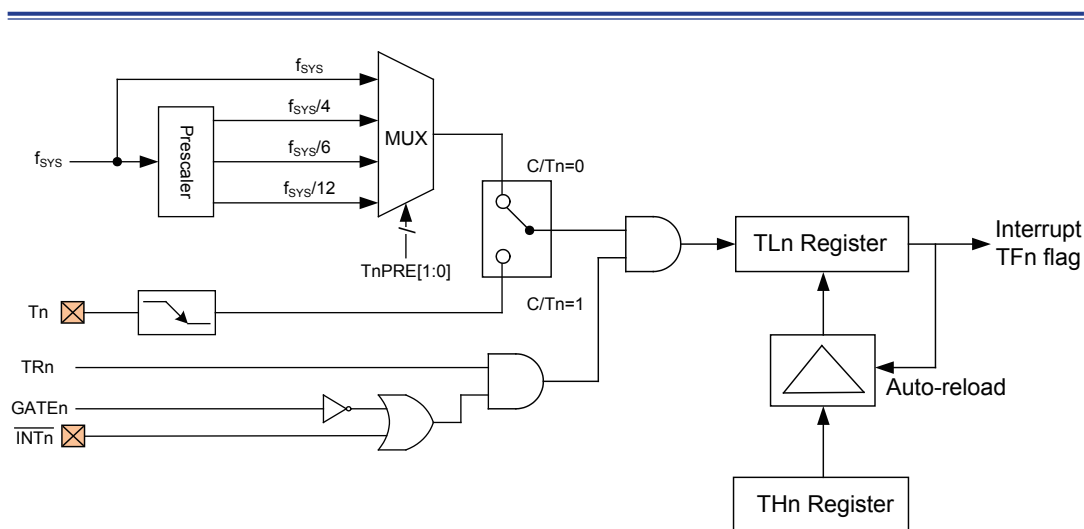
The following block illustrates the 13-bit and 16-bit Timer/Counter basic operational blocks.



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

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 $\overline{\text{INTn}}$ 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.

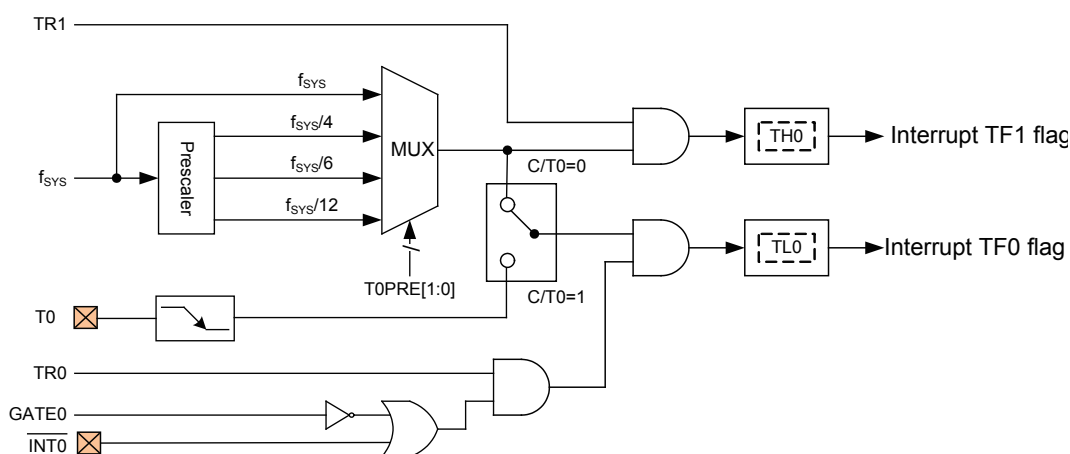


Mode 2 Block Diagram – Timer 0, 1

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, 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 $\overline{\text{INT0}}$ 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 TOPRE0 and TOPRE1 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 timer overflows, an interrupt will be generated and the interrupt request flag, TF1, will be set high. The timer clock source is decided by the TOPRE0 and TOPRE1 bits in the TMPRE register. The following block illustrates the two 8-bit Timer/Counters basic operational blocks.



Mode 3 Block Diagram – Timer 0

28 Timer 2 with Additional 2-channel PCA

The structure of Timer 2 is very different from that of Timers 0 and 1 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 2-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 1. 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.

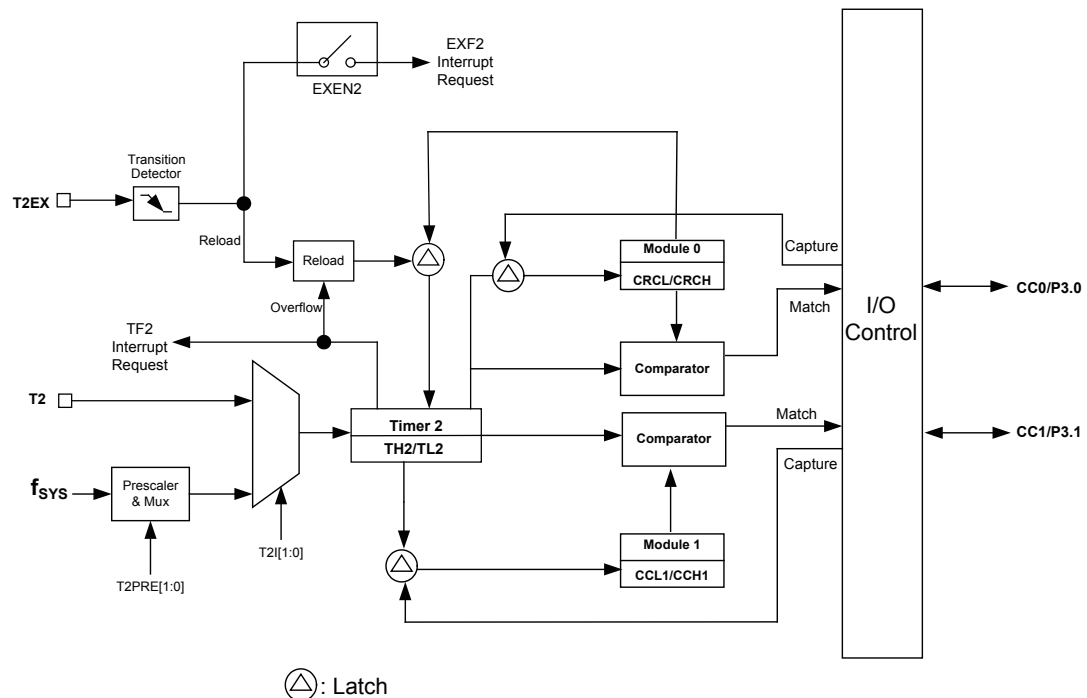
Timer 2 with PCA Modules Operating Modes Summary

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

Note: Module 0 only provides the reload value from the Timer 2 capture registers CRCH and CRCL for 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
Reload trigger	T2EX	—
Capture	CC0, CC1	—
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 T2 pin 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 two Timer 2 Modules contains a pair of registers, CRCL/CRCH for Module 0 and CCL1/CCH1 for Modules 1. 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 CCL1 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 CCL1/CCH1 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, T2. 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 CRCL/CRCH registers 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 CCL1/CCH1 for module 1. 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

Register Name	Bit							
	7	6	5	4	3	2	1	0
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
CCL1	D7	D6	D5	D4	D3	D2	D1	D0
CCH1	D15	D14	D13	D12	D11	D10	D9	D8
T2CON	—	I3FR	—	T2R1	T2R0	T2CM	T2I1	T2I0
T2CON1	—	—	—	—	T2OI	T2OE	—	—
CCEN	—	—	—	—	COCAH1	COCAL1	COCAH0	COCAL0

Note: n = 1, 2, 3

CCEN Register

SFR Address: C1h

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	COCAH1	COCAL1	COCAH0	COCAL0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as “0”

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 writing 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 writing data into register CRCL

T2CON Register

SFR Address: C8h

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

- Bit 7 Unimplemented, read as “0”
- Bit 6 **I3FR**: Timer 2 Capture Mode 0 edge selection for “CC0”
0: Falling edge
1: Rising edge
This bit is used as capture signal in CC0. When Timer 2 is selected as compare mode 0, the I3FR bit is recommended to be set high by application program.
- Bit 5 Unimplemented, read as “0”
- 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

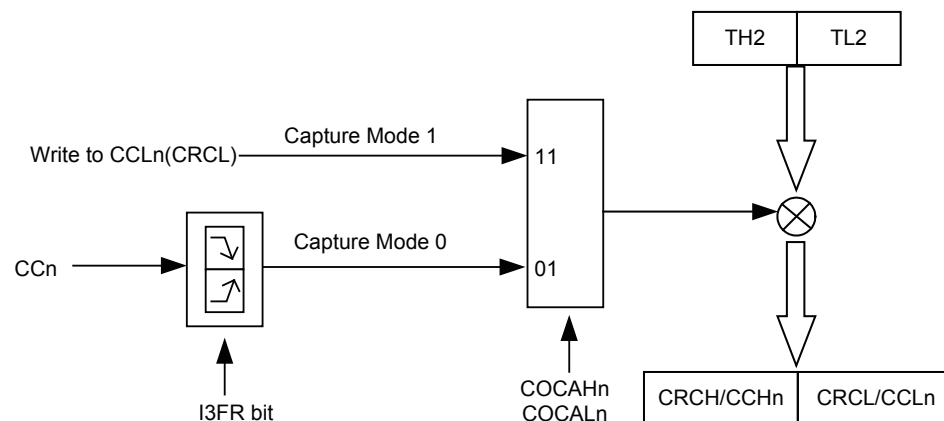
SFR Address: FEh

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 clock output enable bit
0: Disable
1: Enable
The Timer 2 clock output is enabled by setting the T2OE bit high. When the Timer 2 clock output is disabled, the T2 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. $n=0\sim 1$

2. CC1 capture input by rising edge
3. CC0 capture input by rising or falling edge selection by the I3FR bit
4. Write to CCLn is for CC1 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 will capture the Timer 2 counter on the rising edge of an external signal applied on the CC1 pin. 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. In this mode, when the CCUn interrupt is enabled and a trigger edge is detected on CCn input pin, the CCUnF interrupt flag will be set high to generate an interrupt.

Capture On Write Mode

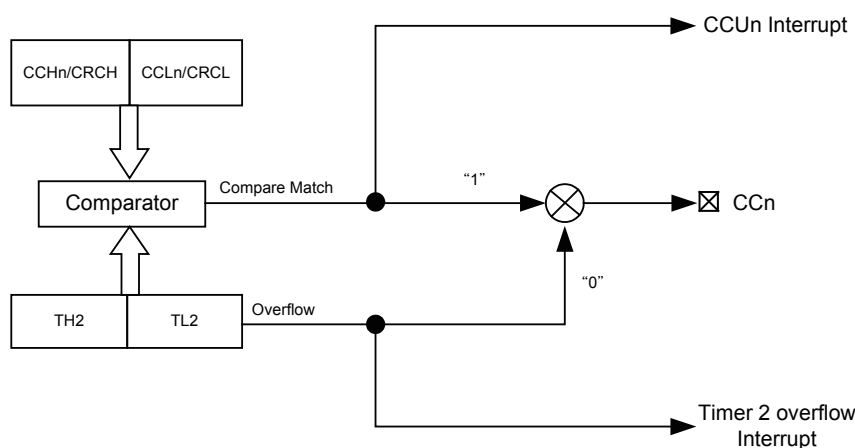
To select this mode, bits COCAHn and COCALn in the CCEN register, should be set to “11” respectively. In this mode a Timer 2 Capture is generated by any write operation into the capture register low byte. Note that the capture action occurs right after the write operation and the value written to capture register is irrelevant for this function. The Timer 2 contents will be latched into the appropriate capture registers. In this mode, no interrupt request will be generated.

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. The CCUnF interrupt flag will be set high when Timer 2 compare mode is enabled and counter value (TH2, TL2) is equal to Compare/Capture register, CCH1 and CCL1, or Compare/Reload/Capture register, CRCH and CRCL.

Compare Mode 0

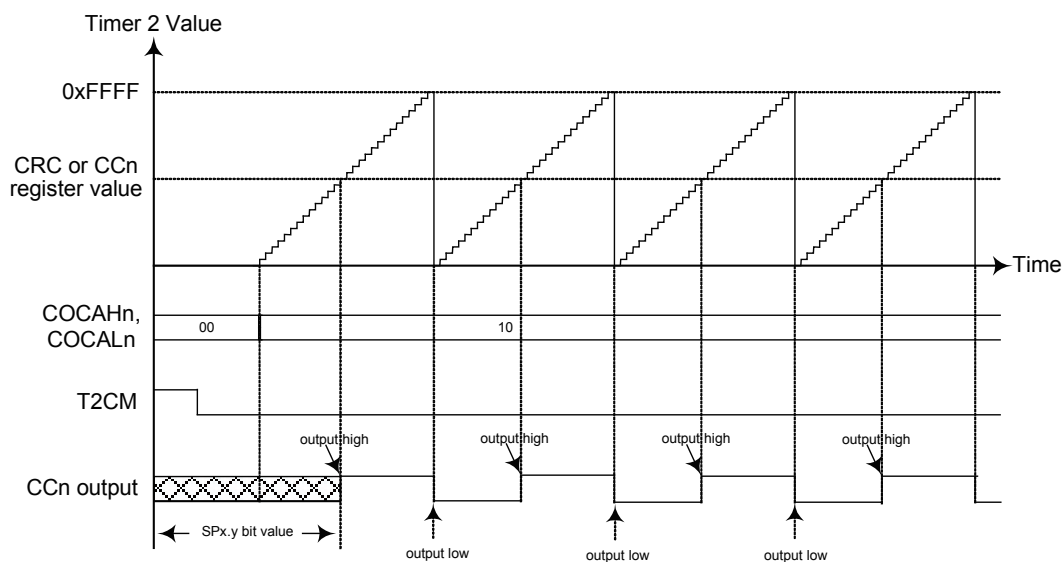
In Mode 0, if the Timer 2 counter data is the same as the Compare registers, the compare output will be set from low to high and the Timer 2 counter overflow will clear the respective output pins, CCn, to low. The accompanying diagrams illustrate the Basic application blocks.



Note: CCHn&CCLn for CC1, CRCH&CRCL for CC0

Compare Mode 0 – Module 0~1

Figure below illustrates the operation of compare mode 0.

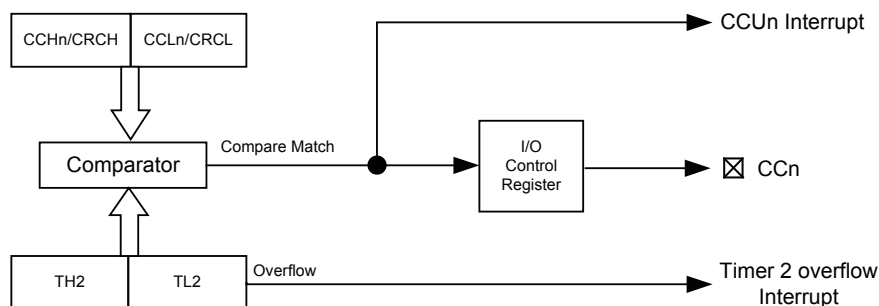


- Note:**
1. Px.y is the corresponding data bit of the I/O pin which is pin-shared with CCn.
 2. SPx.y is the corresponding shadow bit of Px.y. When the Px.y is selected as CCn pin function, its data bit becomes the shadow bit.
 3. The CCn pin output changes to low when timer 2 overflows.

Compare Match Mode 0 Timing Diagram – T2CM=0

Compare Mode 1

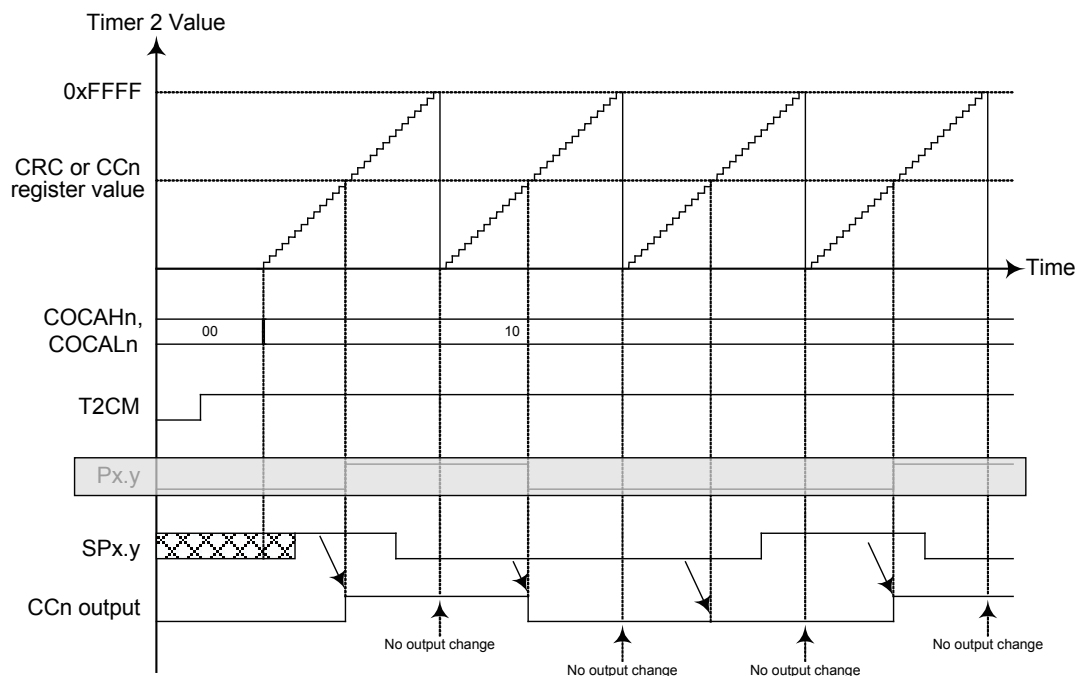
In Mode 1, the compare output can be decided by the software setting of the related I/O pins register. When the compare match takes place, the control register value will be output to I/O pins, CCn, and the Timer 2 counter overflow will not affect the Compare output. The accompanying diagrams illustrate the Basic application blocks.



Note: CCHn&CCLn for CC1, CRCH&CRCL for CC0

Compare Mode 1 – Module0~1

Figure below illustrates the operation of compare mode 1.

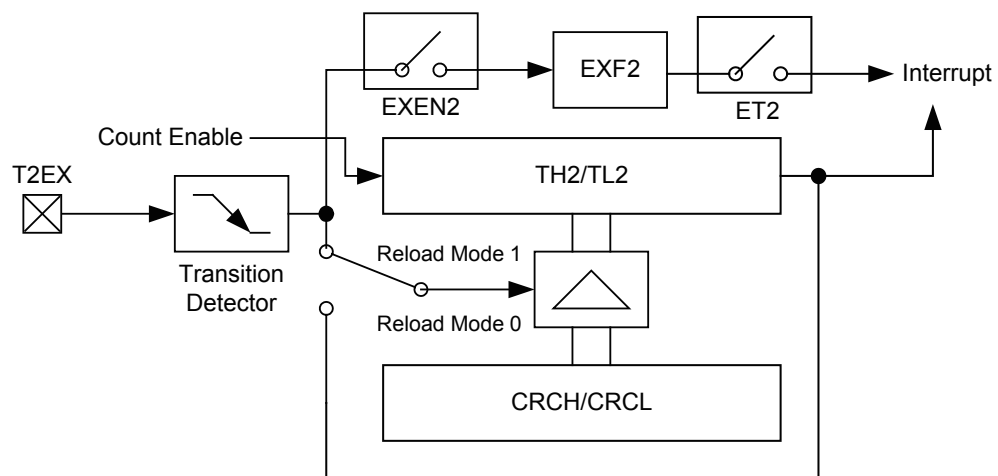


- Note:**
1. Px.y is the corresponding data bit of the I/O pin which is pin-shared with CCn.
 2. SPx.y is the corresponding shadow bit of Px.y and is used to control the CCn output when compare match occurs. When the Px.y is selected as CCn pin function, its data bit becomes the shadow bit.
 3. There will no output change on CCn pin when timer 2 overflows.

Compare Match Mode 1 Timing Diagram – T2CM=1

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 action 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 external reload interrupt will be generated. The following diagram illustrates the basic operation.



Reload Mode – Module 0

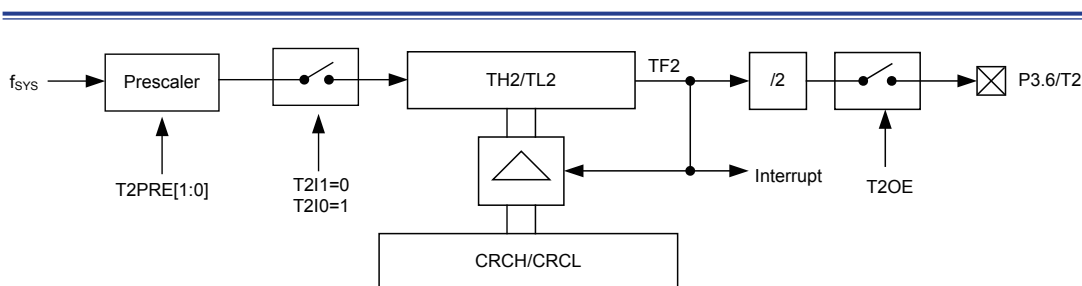
Programmable Clock Output Mode

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. Besides being a regular I/O pin, the P3.6 has two alternate functions. One function 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 Timer 2 clock frequency is 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 output frequency depends on the Timer 2 clock frequency and the reload value of Timer 2 capture registers (CRCH, CRCL) as shown in this equation:

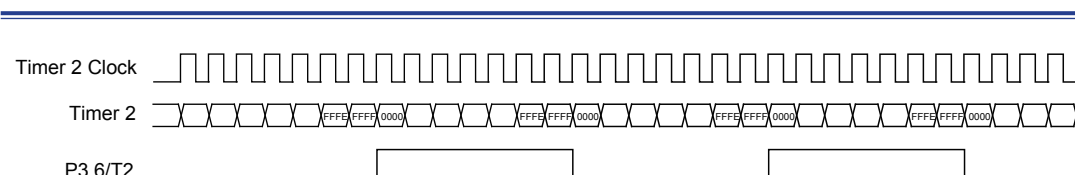
$$\text{Clock Output Frequency} = \frac{\text{Timer 2 Clock Frequency}}{2 * (65536 - [\text{CRCH}, \text{CRCL}])}$$

While the Timer 2 clock frequency is furtherly determined by 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 3 control registers, P3M1 and P3M0, to setup the P3.6 pin as an output. The accompanying diagram illustrates the Timer 2 programmable clock output timing diagram.



Programmable Clock Output Timing Diagram – Module 0

29 Analog to Digital Converter – ADC

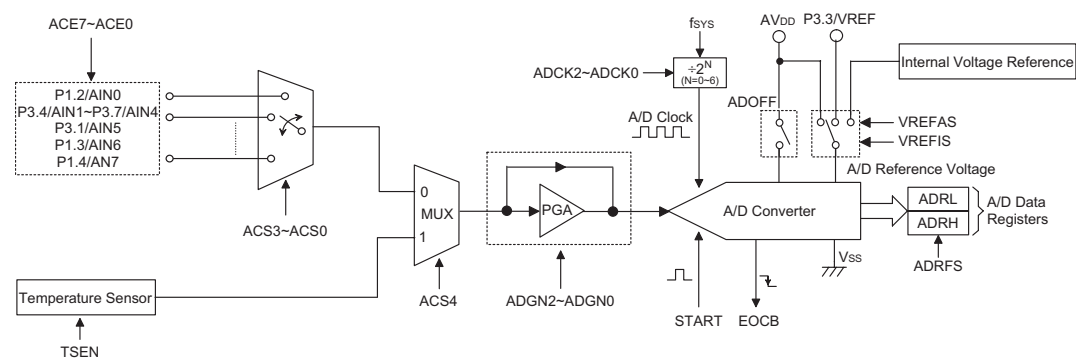
The devices include a multi-channel 12-bit fully integrated Analog to Digital Converter or ADC. A range of programmable features allow flexible and fast analog to digital conversion for a wide range of input signals.

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

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

Register 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 (SFR Address: F6h)								ADRL (SFR Address: F5h)							
	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 which pins on Port 1 and Port 3 are used as A/D converter input and which pins are not used as A/D converter input. Setting the corresponding bit high will select the A/D input function, clearing the bit to zero will select the I/O function or other pin-shared function. When the pin is selected to be an A/D input, its original function whether it is an I/O or other pin-shared 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: AIN0
 0001: AIN1
 0010: AIN2
 0011: AIN3
 0100: AIN4
 0101: AIN5
 0110: AIN6
 0111: AIN7
 1xxx: Undefined, can 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: AV_{DD}
1: VREFI
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 provided by VREFI, which 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 AV_{DD} .
- Bit 3 VREFIS:** VREFI source select
0: Externally supplied on VREF pin
1: Internal Voltage Reference Generator
- 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:** P1.4 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN7
- Bit 6 **ACE6:** P1.3 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN6
- Bit 5 **ACE5:** P3.1 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN5
- Bit 4 **ACE4:** P3.7 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN4
- Bit 3 **ACE3:** P3.6 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN3
- Bit 2 **ACE2:** P3.5 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN2
- Bit 1 **ACE1:** P3.4 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN1
- Bit 0 **ACE0:** P1.2 A/D input select
 0: I/O line or pin-shared function
 1: A/D input, AIN0

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 selection

000: PGA Bypass (PGA off)

001: PGA Bypass (PGA off)

010: Gain=1 (PGA on)

011: Gain=2

100: Gain=4

101: Gain=8

110: Gain=12

111: Gain=16

These three bits are used to select the PGA internal gain setting to allow greater A/D Converter input voltage dynamic range.

When the Programmable Gain Amplifier is disabled by Setting ADGN2~ ADGN0 as 000 or 001, the 12-bit A/D converter input signal will bypass the PGA. Otherwise the Programmable Gain Amplifier is enabled and the 12-bit A/D converter input signal will go through the PGA.

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 A/D power supply AV_{DD} , internal voltage reference or from an external reference source 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 from 0.5 μ s to 10 μ s, care must be taken for different system clock frequencies. For example, if the system clock operates at a frequency of 4MHz, the ADCK2~ADCK0 bits should not be set to “000”, “110” or “111”. 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 (f_{SYS})
1MHz	1 μ s	2 μ s	4 μ s	8 μ s	16 μ s*	32 μ s*	64 μ s*	1 μ s
2MHz	500ns	1 μ s	2 μ s	4 μ s	8 μ s	16 μ s*	32 μ s*	500ns
4MHz	250ns*	500ns	1 μ s	2 μ s	4 μ s	8 μ s	16 μ s*	250ns*
8MHz	125ns*	250ns*	500ns	1 μ s	2 μ s	4 μ s	8 μ s	125ns*
12MHz	83ns*	167ns*	333ns*	667ns	1.33 μ s	2.67 μ s	5.33 μ s	83ns*
16MHz	62.5ns*	125ns*	250ns*	500ns	1 μ s	2 μ s	4 μ s	62.5ns*
32MHz	31.25ns*	62.5ns*	125ns*	250ns*	500ns	1 μ s	2 μ s	31.25ns*

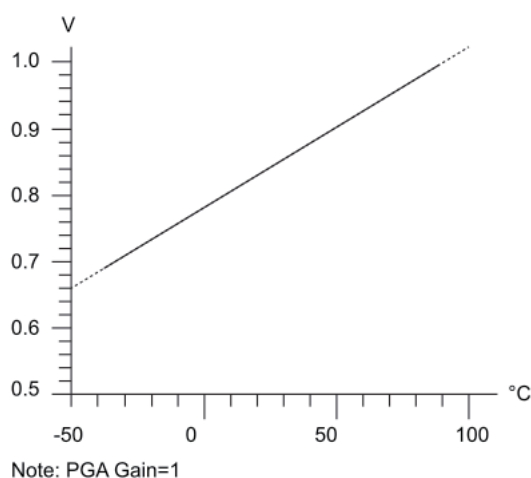
A/D Input Pins

All of the A/D analog input pins are pin-shared with the I/O pins on P1 or P3 function. The ACE7~ACE0 bits in the ADCR2 registers, determine whether the input pins are setup as A/D converter analog inputs or other functions. 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 other functions. 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 P1 or P3 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 A/D power supply AV_{DD} 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.



Temperature vs Voltage Diagram

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 AV_{DD} power supply, 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	AV_{DD}
0	1	Externally supplied on VREF pin
1	0	AV_{DD}
1	1	Internal Voltage Reference Generator

Summary of A/D Conversion Steps

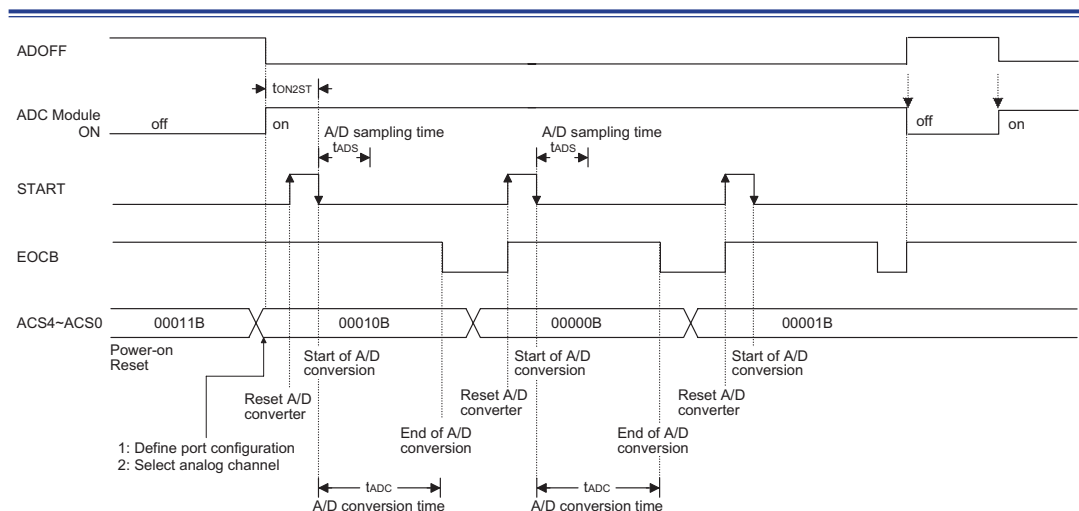
The following summarises the individual steps that should be executed in order to implement an A/D conversion process.

- 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 completed, 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

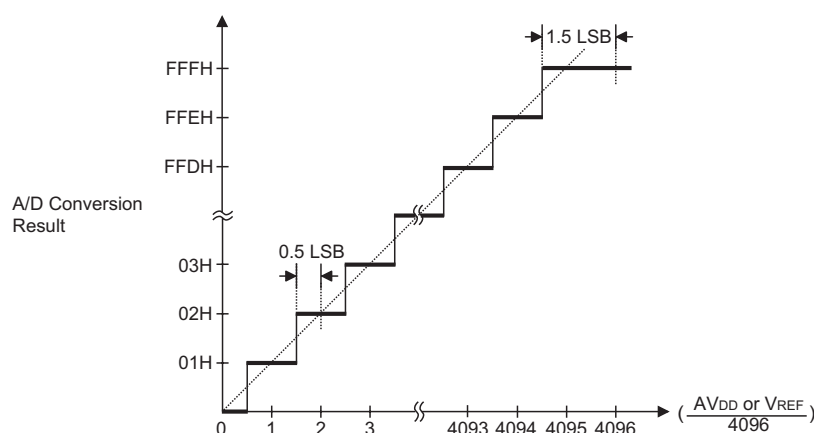
As the converted data is 12-bit wide, its full-scale converted digitised value is equal to FFFH. Since the full-scale analog input value is equal to the AV_{DD} or V_{REF} voltage, this gives a single bit analog input value of AV_{DD} or V_{REF} divided by 4096.

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

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

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

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 AV_{DD} or V_{REF} level.



Ideal A/D Transfer Function (PGA Gain=1)

30 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.

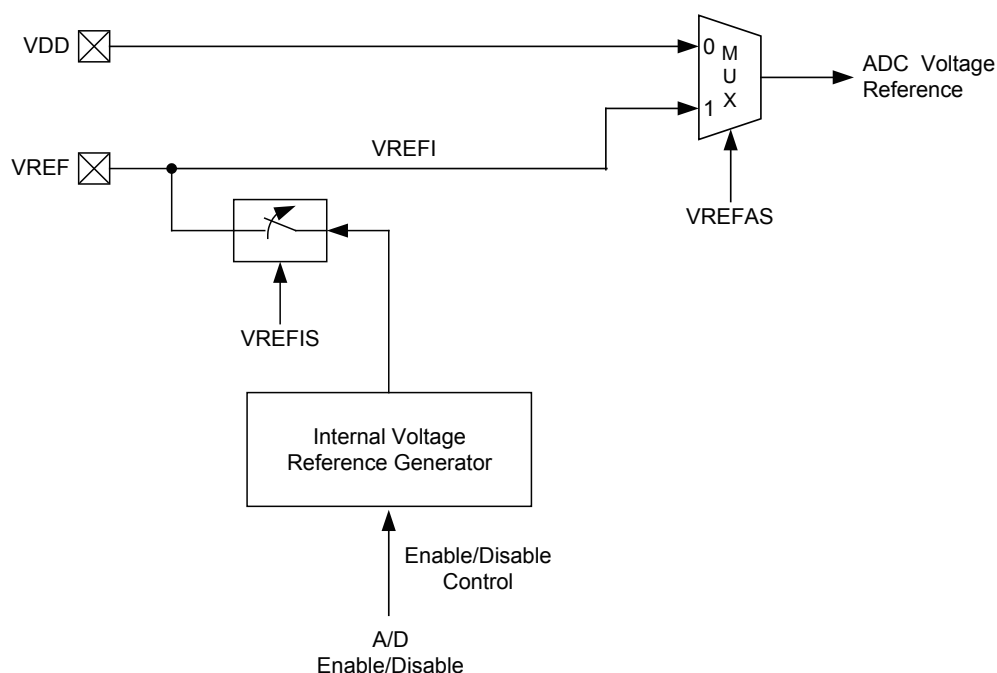
Voltage Reference Generator Operation

The voltage reference circuit will be automatically enabled when the A/D converter is enabled. If the A/D converter is disabled then the generator will be disabled thus conserving power.

Internal Voltage Reference Enable/Disable Control

ADC	Voltage Reference
Disabled	Disabled
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 reference voltage is selected by the VREFIS and VREFAS control bits. When the VREFIS bit is set the internal voltage reference will be routed to pin VREF and can be selected for use by the A/D converter, using the VREFAS bit.

31 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.

32 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](http://www.holtek.com) 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 Materials Information](#)
- [Carton information](#)

20-pin DIP (300mil) Outline Dimensions

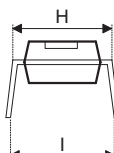
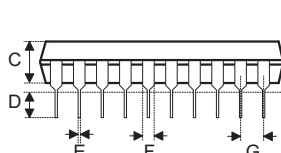
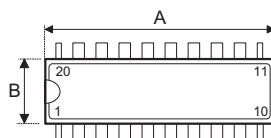
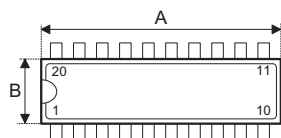


Fig 1. Full Lead Packages

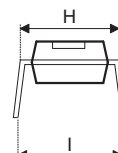
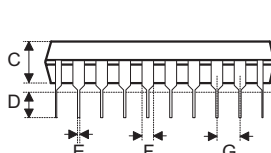


Fig 2. 1/2 Lead Packages

See fig 1

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.980	1.030	1.060
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.1BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

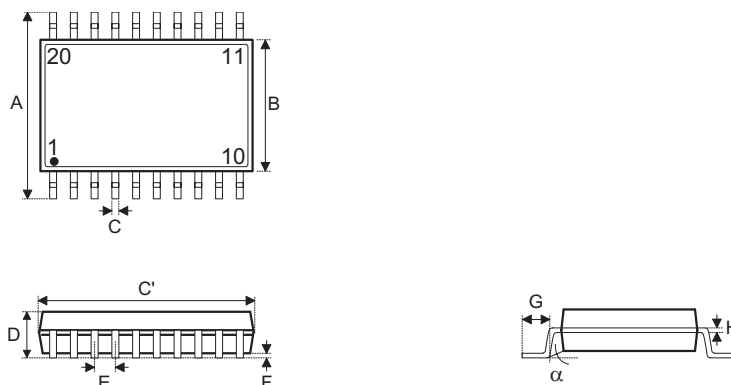
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	24.89	26.16	26.92
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

See fig2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.945	0.965	0.985
B	0.275	0.285	0.295
C	0.120	0.135	0.150
D	0.110	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.050	0.060
G	—	0.1BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	24.00	24.51	25.02
B	6.99	7.24	7.49
C	3.05	3.43	3.81
D	2.79	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.27	1.52
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

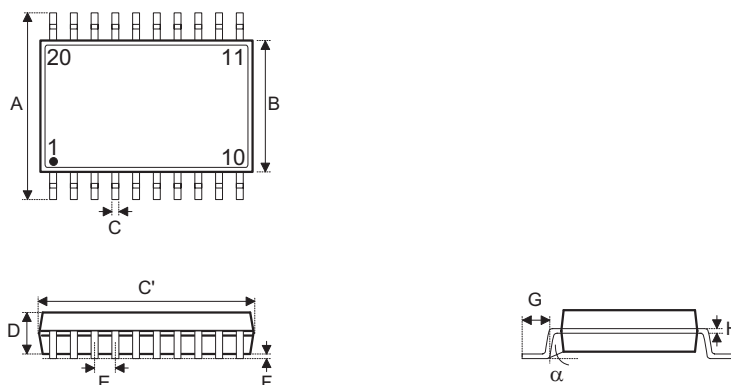
20-pin SOP (300mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.406 BSC	—
B	—	0.406 BSC	—
C	0.012	—	0.020
C'	—	0.504 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	10.30 BSC	—
B	—	7.5 BSC	—
C	—	7.5 BSC	—
C'	—	12.8 BSC	—
D	—	12.8 BSC	—
E	—	1.27 BSC	—
F	0.10	—	0.30
G	0.40	—	1.27
H	0.40	—	1.27
α	0°	—	8°

20-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.155 BSC	—
C	0.008	—	0.012
C'	—	0.341 BSC	—
D	—	—	0.069
E	—	0.025 BSC	—
F	0.004	—	0.0098
G	0.016	—	0.05
H	0.004	—	0.01
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6 BSC	—
B	—	3.9 BSC	—
C	0.20	—	0.30
C'	—	8.66 BSC	—
D	—	—	1.75
E	—	0.635 BSC	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

24-pin SKDIP (300mil) Outline Dimensions

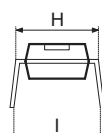
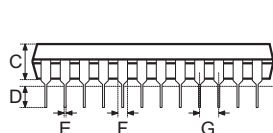
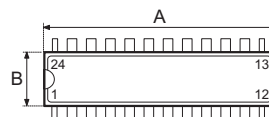
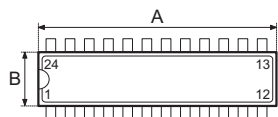


Fig1. Full Lead Packages

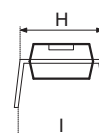
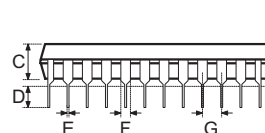


Fig2. 1/2 Lead Packages

See fig1

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.230	1.250	1.280
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.1 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	31.24	31.75	32.51
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

See fig2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.160	1.185	1.195
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.1 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

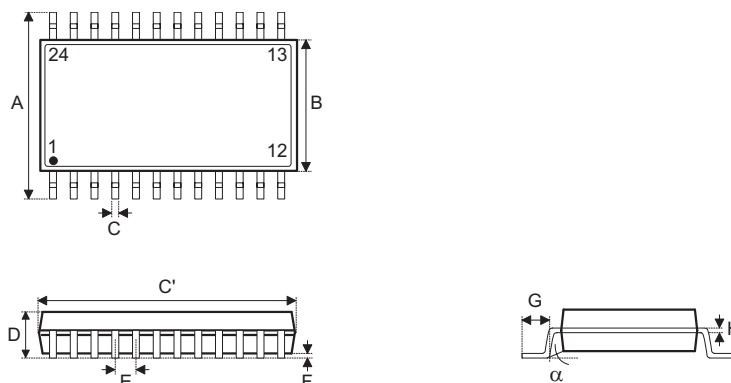
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.46	30.10	30.35
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

See fig2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.145	1.165	1.185
B	0.275	0.285	0.295
C	0.120	0.135	0.150
D	0.110	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.050	0.060
G	—	0.1 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.08	29.59	30.10
B	6.99	7.24	7.49
C	3.05	3.43	3.81
D	2.79	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.27	1.52
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

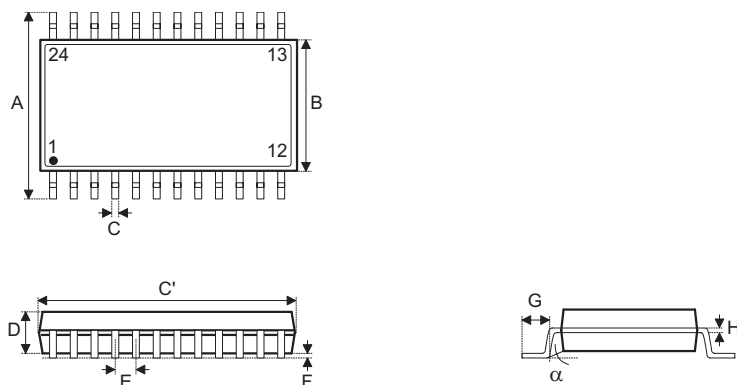
24-pin SOP (300mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.406 BSC	—
B	—	0.295 BSC	—
C	0.012	—	0.020
C'	—	0.606 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	10.30 BSC	—
B	—	7.5 BSC	—
C	0.31	—	0.51
C'	—	15.4 BSC	—
D	—	—	2.65
E	—	1.27 BSC	—
F	0.10	—	0.30
G	0.40	—	1.27
H	0.20	—	0.33
α	0°	—	8°

24-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.008	—	0.012
C'	—	0.341 BSC	—
D	—	—	0.069
E	—	0.025 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.0 BSC	—
B	—	3.9 BSC	—
C	0.20	—	0.30
C'	0.20	—	0.30
D	—	—	1.75
E	—	0.635 BSC	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

Copyright© 2015 by HOLTEK SEMICONDUCTOR INC.

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>.