8051 Embedded Micro-Controller with 32KB Flash ROM

GENERAL DESCRIPTION

CS8953 is a general purpose microcontroller with extensive peripherals that are suitable for wide range of application. The CPU is based on an enhanced 1-Cycle 8051 core that is equivalent to 10 times faster than a conventional 12-Cycle 8051. Total on-chip SRAM is 1KB for data storage. There is total 32KB of embedded flash memory that can be used as program memory and portion of this can be used as data flash. The 8051 core has built-in T0/T1/T2 timers and a 16-bit watchdog timer. Embedded in the CPU core are also two full-duplex UART ports, one I^{2}C master controller, and up to 28 GPIO, and up to 8 external interrupts.

Flexibility in clock includes an on-chip precision oscillator that have +/-2% accuracy or ultra low power real time clock, and regular crystal oscillator. Unused clock sources can be disabled or used as GPIO pins for best system optimization. The clock selections are also combined with flexible power management schemes, including PMM, IDLE, STOP modes to balance CPU speed and power consumption. The CPU also includes access protection for critical registers that improves the overall reliable system operations.

Other on-chip peripherals include one slave I^{2}C controller, a Programmable Counter Array (PCA) with 6 channels of Capture/Compare/PWM modules. There is a high performance Analog to Digital Converter with 3.5usec conversion time. The ADC can be configured in various modes to meet wide range of applications.

CS8953 also provides a flexible means of flash programming that supports ISP and IAP. The protections of loss of Flash contents are implemented in hardware. There is also access restriction on critical registers and low supply voltage detection that allow CS8953 reliable operations under harsh environment. The code security is extremely secure based on sophisticated writer command and ISP command.

Intended application fields of CS8953 include Automobile AV system, Home Appliance, and other embedded applications.

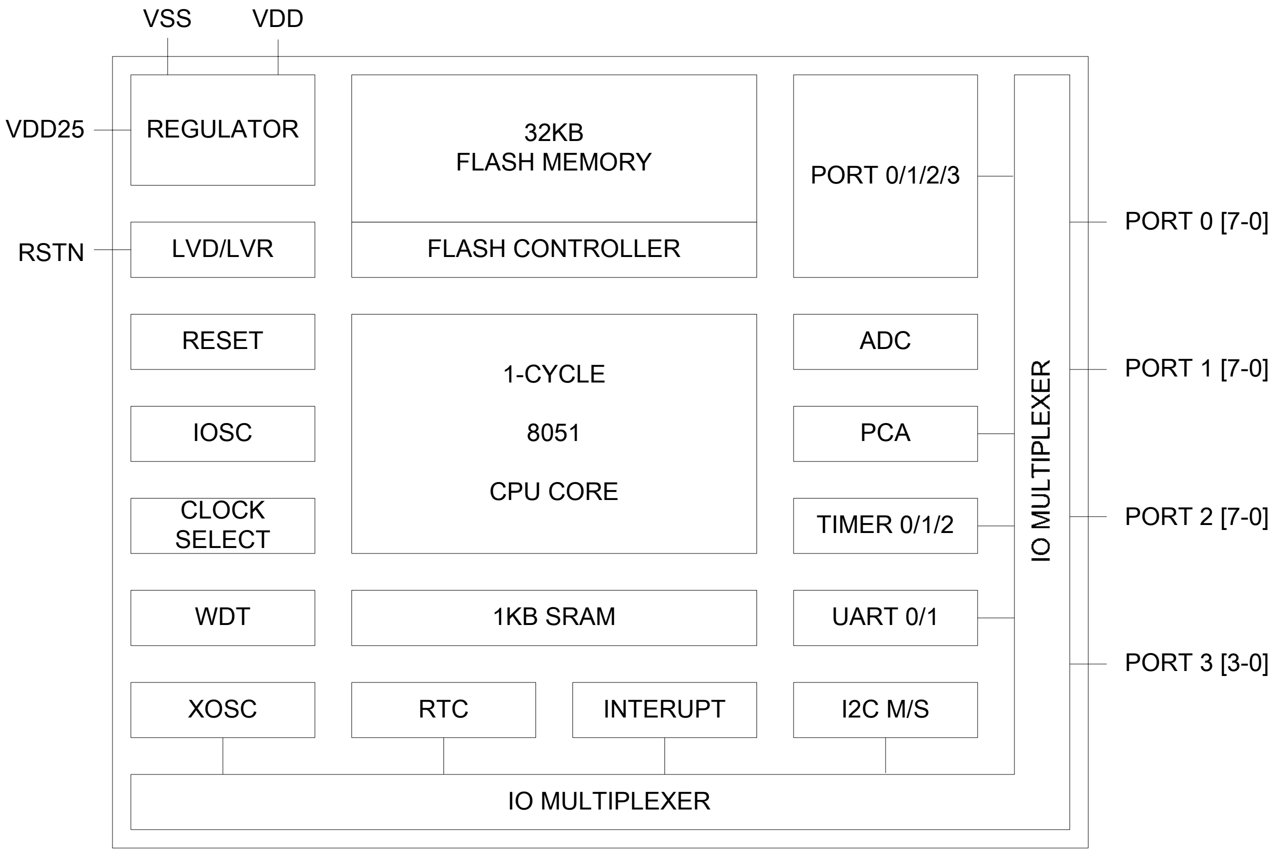
FEATURES

- 1-Cycle 8051 core up to 24MHz operation
  - Two 16-bit Timers T0/T1
  - One 16-bit ECT Timer T2
  - Programmable 16-bit Watch Dog Timer
  - Two full-duplex UART ports
  - Up to 8 external interrupts shared with GPIO pins
  - One I^{2}C Master Controller up to 400Kbit/sec
- One I2C Slave Controller shared with I2C Master pins
- 256B Internal SRAM and 768B XRAM
- 32KB Flash Memory
  - Configured to be shared by ISP code, program code, and data flash
  - Code security and content loss protection
- Programmable Counter Array (16-bit) with 6 CCP Modules
  - Capture Mode
  - Compare/Timer Mode
  - 8-Bit PWM Mode
  - 8-Bit Windowed PWM Mode
- 3.68MHz Internal Oscillator with typical +/- 2% deviation
- Real Time Clock (RTC) less than 2uA current consumption
- 10-bit monotonic SAR ADC
  - 3.5us conversion time
  - 0-VDD or 0-0.5VDD full scale
  - 4 Intrinsic Time Multiplexed Channels
  - 10 inputs multiplexed with GPIO pins
- 4 Analog Comparators
  - Two programmable 256 steps threshold
  - Input range 0V – 2.5V
  - Detection time less than 250nsec
- 3 2PST Analog Switches
- Low Voltage Detection on VDD
  - Programmable threshold 2.7V ~ 4.8V
  - Programmable for generation of Low Voltage Interrupt or Low Voltage Reset
- Up to 28 GPIO (CS8953AR or CS8953AN/ CS8953AG)
  - Configurable pin functions – OD/OS/LAT
  - 24/28 using IOSC
  - 22/26 using IOSC + RTC
  - 20/24 using RTC + XOSC
- Power saving modes
  - Normal operation: 5mA ~ 20mA
  - PMM mode: 0.2mA ~ 5mA
  - Idle mode: 0.2mA ~ 5mA
  - Stop mode (recovered by hardware reset, or wake up by WK condition): 300uA
- Single power supply
  - 3.0V ~ 5.5V
- Temperature range: -40°C ~ +85°C
- CS8953AN for Skinny DIP-28 or CS8953AR for SSOP-28 or CS8953AG for LQFP-32 RoHS

*** This datasheet is the confidential information of* MYSON CENTURY, INC. *and is subject to various privileges against unauthorized disclosure. Recipient shall not disclose this confidential information to any other person, nor shall one use the confidential information for the purpose of competing with* MYSON CENTURY, INC.

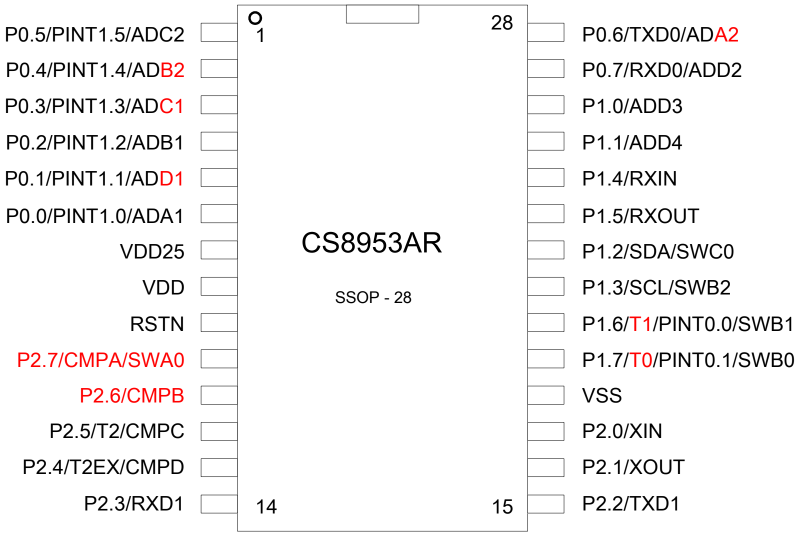
compliance packages

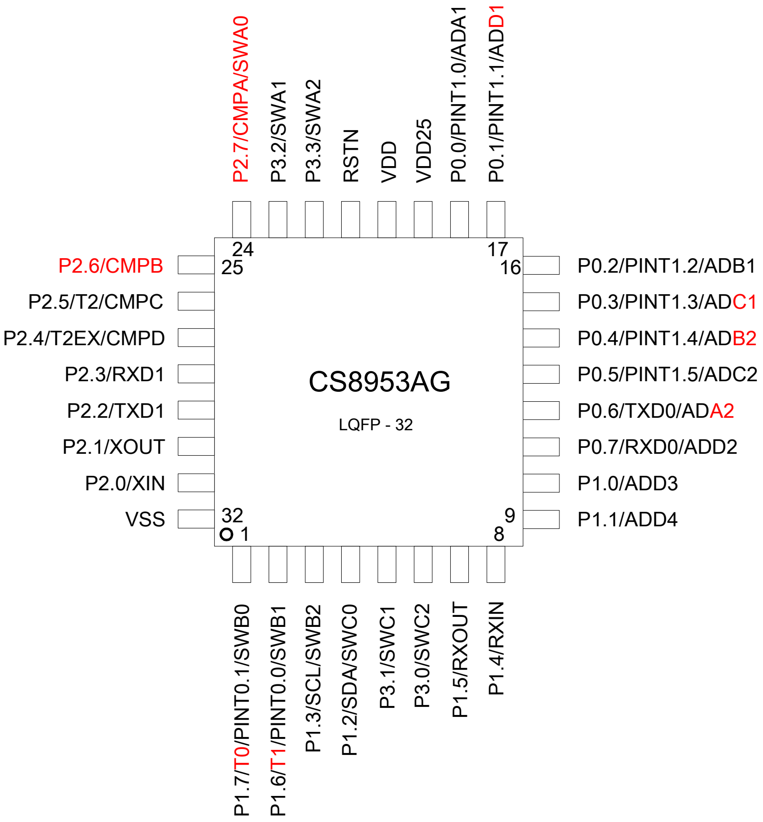
BLOCK DIAGRAM

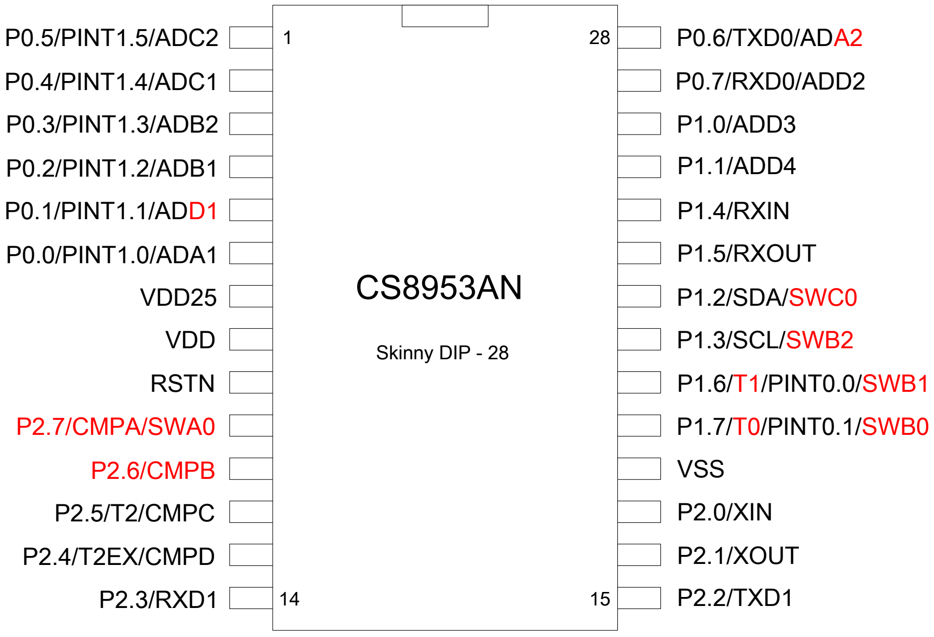


PIN CONNECTION

Note: The part number or description in this section is only for pin assignment and not the actual marking or logo.






P0.5/PINT1.5/ADC2
P0.4/PINT1.4/ADC1
P0.3/PINT1.3/ADB2
P0.2/PINT1.2/ADB1
P0.1/PINT1.1/ADD1
P0.0/PINT1.0/ADA1
VDD25
VDD
RSTN
P2.7/CMPA/SWA0
P2.6/CMPB
P2.5/T2/CMPC
P2.4/T2EX/CMPD
P2.3/RXD1
1
14
CS8953AN
Skinny DIP - 28
28
15
P0.6/TXD0/ADA2
P0.7/RXD0/ADD2
P1.0/ADD3
P1.1/ADD4
P1.4/RXIN
P1.5/RXOUT
P1.2/SDA/SWC0
P1.3/SCL/SWB2
P1.6/T1/PINT0.0/SWB1
P1.7/T0/PINT0.1/SWB0
VSS
P2.0/XIN
P2.1/XOUT
P2.2/TXD1

PIN DEFINITION AND DESCRIPTION

PIN NAME	PIN TYPE	PIN # CS8953AR/ CS8953AN	PIN# CS8953AG	PIN FUNCTION DESCRIPTION
VDD	Power	8	20	Supply Voltage. 3.0V – 5.5V. A good decoupling capacitor between VDD and VSS pins is critical for good performance
VSS	Power	18	32	Ground Voltage. 0V.
VDD25	Power Output	7	19	Internal Regulator Output. 2.25V – 2.75V. Typical decoupling capacitors of 0.1uF and 10uF should be connected between VDD25 and VSS.
RSTN	Input	9	21	Reset Low Active. Typically connect a resistor to VDD25 and a capacitor to VSS. Low asserted and threshold at 0.5*VDD25. When forced low, the chip enters into reset condition. This pin should not be connected to any level above VDD25..
P0.0	I/O, A	6	18	Port 0.0 GPIO 8051 P0.0 GPIO PINT1.0 This pin also can be configured as the expanded INT1 interrupt. ADA1 This pin also can be configured as the input to the ADC channel A by setting ANEN of IOCFGP0.0 to 1. Only one of ADA1 and ADA2 can be enabled at any one time.
P0.1	I/O, A	5	17	Port 0.1 GPIO 8051 P0.1 GPIO PINT1.1 This pin also can be configured as the expanded INT1 interrupt. ADD1 This pin also can be configured as the input to the ADC channel D by setting ANEN of IOCFGP0.1 to 1. Only one of ADD1, ADD2, ADD3 and ADD4 can be enabled at any one time
P0.2	I/O, A	4	16	Port 0.2 GPIO 8051 P0.2 GPIO PINT1.2 This pin also can be configured as the expanded INT1 interrupt. ADB1 This pin also can be configured as the input to the ADC channel B by setting ANEN of IOCFGP0.2 to 1. Only one of ADB1 and ADB2 can be enabled at any one time.
P0.3	I/O, A	3	15	Port 0.3 GPIO 8051 P0.3 GPIO PINT1.3 This pin also can be configured as the expanded INT1 interrupt. ADC1 This pin also can be configured as the input to the ADC channel C by setting ANEN of IOCFGP0.3 to 1. Only one of ADC1 and ADC2 can be enabled at any one time.
P0.4	I/O, A	2	14	Port 0.4 GPIO 8051 P0.4 GPIO PINT1.4 This pin also can be configured as the expanded INT1 interrupt. ADB2 This pin also can be configured as the input to the ADC channel B by setting ANEN of IOCFGP0.4 to 1. Only one of ADB1 and

				ADB2 can be enabled at any one time.
P0.5	I/O, A	1	13	Port 0.5 GPIO 8051 P0.2 GPIO PINT1.5 This pin also can be configured as the expanded INT1 interrupt. ADC2 This pin also can be configured as the input to the ADC channel C by setting ANEN of IOCFGP0.5 to 1. Only one of ADC1 and ADC2 can be enabled at any one time.
P0.6	I/O, A	28	12	Port 0.6 GPIO 8051 P0.6 GPIO TXD0 This pin also can be configured as TXD of UART 0. ADA2 This pin also can be configured as the input to the ADC channel A by setting ANEN of IOCFGP0.6 to 1. Only one of ADA1 and ADA2 can be enabled at any one time.
P0.7	I/O, A	27	11	Port 0.7 GPIO 8051 P0.6 GPIO RXD0 This pin also can be configured as RXD of UART 0. ADD2 This pin also can be configured as the input to the ADC channel D by setting ANEN of IOCFGP0.7 to 1. Only one of ADD1, ADD2, ADD3 and ADD4 can be enabled at any one time.
P1.0	I/O, A	26	10	Port 1.0 GPIO 8051 P1.0 GPIO CEX0 PCA CCAP Module 0 This pin also can be configured as CEX pin for PCA CCP module 0. CEX is an I/O interface signal for compare/capture input and PWM output. ADD3 This pin also can be configured as the input to the ADC channel D by setting ANEN of IOCFGP1.0 to 1. Only one of ADD1, ADD2, ADD3 and ADD4 can be enabled at any one time.
P1.1	I/O, A	25	9	Port 1.1 GPIO 8051 P1.0 GPIO CEX1 PCA CCAP Module 1 This pin also can be configured as CEX pin for PCA CCP module 1. CEX is an I/O interface signal for compare/capture input and PWM output. ADD4 This pin also can be configured as the input to the ADC channel D by setting ANEN of IOCFGP1.1 to 1. Only one of ADD1, ADD2, ADD3 and ADD4 can be enabled at any one time.
P1.2	I/O, A	22	4	Port 1.2 GPIO 8051 P1.2 GPIO SDA This pin also can be configured as the SDA signal of the I2C master or I2C slave controller. In this operation mode, this pin should also be configured as bi-directional I/O with open-drain output. SWC0 This pin also serves as one of the connection for analog switch C. The control of the analog switch is done by setting of ANEN of IOCFGP1.2.

P1.3	I/O, A	21	3	Port 1.3 GPIO 8051 P1.3 GPIO SCL This pin also can be configured as the SCL signal of the I2C master or I2C slave controller. In I2C master mode, this pin should be configured as open-drain output. In I2C slave, this pin should be configured as input only. SWB2 This pin also serves as one of the connection for analog switch B. The control of the analog switch is done by setting of ANEN of IOCFGP1.3.
P1.4	I/O, A	24	8	Port 1.4 GPIO 8051 P1.4 GPIO RXIN – RTC Crystal In This pin also can be configured as crystal IN for RTC oscillator. The connection between the pin and the oscillator is through the analog switch in the IO circuit. In this mode, ANEN of IOCFGP1.4 should be set and the reset of setting all cleared.
P1.5	I/O, A	23	7	Port 1.5 GPIO 8051 P1.5 GPIO RXOUT – RTC Crystal OUT This pin also can be configured as crystal OUT for RTC oscillator. The connection between the pin and the oscillator is through the analog switch in the IO circuit. In this mode, ANEN of IOCFGP1.5 should be set and the reset of setting all cleared.
P1.6	I/O, A	20	2	Port 1.6 GPIO 8051 P1.6 GPIO T1 Timer 1 Input This pin also can be configured as Timer 1 input PINT0.0 This pin also can be configured as the expanded INT0 interrupt. SWB1 This pin also serves as one of the connection for analog switch B. The control of the analog switch is done by setting of ANEN of IOCFGP1.6.
P1.7	I/O, A	19	1	Port 1.7 GPIO 8051 P1.7 GPIO T0 Timer 0 Input This pin also can be configured as Timer 0 input PINT0.1 This pin also can be configured as the expanded INT0 interrupt. SWB0 This pin also serves as one of the connection for analog switch B. The control of the analog switch is done by setting of ANEN of IOCFGP1.7.
P2.0	I/O	17	31	Port 2.0 GPIO 8051 P2.0 GPIO. To allow proper operation as GPIO P2.0 function, crystal oscillator must be disabled by setting XOSCCFG register to 0x00. XIN Crystal Oscillator Input This pin also can be configured as XIN for crystal oscillator. XIN is in parallel connection with the GPIO pin. To enable this pin as XIN, the IOCFGP2.0 must be cleared to 0x00.
P2.1	I/O	16	30	Port 2.1 GPIO 8051 P2.1 GPIO. To allow proper operation as GPIO P2.1 function, crystal oscillator must be disabled by setting XOSCCFG

				register to 0x00. XOUT Crystal Oscillator Output This pin also can be configured as XOUT for crystal oscillator. XOUT is in parallel connection with the GPIO pin. To enable this pin as XOUT, the IOCFGP2.1 must be cleared to 0x00.
P2.2	I/O	15	29	Port 2.2 GPIO 8051 P2.2 GPIO TXD1 This pin also can be configured as TXD of UART 1.
P2.3	I/O, A	14	28	Port 2.3 GPIO 8051 P2.3 GPIO RXD1 This pin also can be configured as RXD of UART 1.
P2.4	I/O, A	13	27	Port 2.4 GPIO 8051 P2.4 GPIO T2EX Timer 2 Trigger This pin also can be configured as T2EX signal for Timer 2. T2EX is the Timer 2 trigger input. Analog Comparator D Input This pin also can be configured as the positive input of the analog comparator D.
P2.5	I/O, A	12	26	Port 2.5 GPIO 8051 P2.5 GPIO T2 Timer 2 Input This pin also can be configured as Timer 2 input Analog Comparator C Input This pin also can be configured as the positive input of the analog comparator C.
P2.6	I/O, A	11	25	Port 2.6 GPIO 8051 P2.6 GPIO ~~T1 Timer 1 Input~~ ~~This pin also can be configured as Timer 1 input~~ Analog Comparator B Input This pin also can be configured as the positive input of the analog comparator B.
P2.7	I/O, A	10	24	Port 2.7 GPIO 8051 P2.7 GPIO ~~T0 Timer 0 Input~~ ~~This pin also can be configured as Timer 0 input~~ Analog Comparator A Input This pin also can be configured as the positive input of the analog comparator A. SWA0 This pin also serves as one of the connection for analog switch A. The control of the analog switch is done by setting of ANEN of IOCFGP2.7.
P3.0	I/O, A	-	6	Port 3.0 GPIO 8051 P3.0 GPIO CEX2 PCA CCAP Module 2 This pin also can be configured as CEX pin for PCA CCP module 2. CEX is an I/O interface signal for compare/capture input and PWM output.
P3.1	I/O, A	-	5	Port 3.1 GPIO 8051 P3.1 GPIO CEX3 PCA CCAP Module 3 This pin also can be configured as CEX pin for PCA CCP module

				3. CEX is an I/O interface signal for compare/capture input and PWM output.
P3.2	I/O, A	-	23	Port 3.2 GPIO 8051 P3.2 GPIO CEX4 PCA CCAP Module 4 This pin also can be configured as CEX pin for PCA CCP module 4. CEX is an I/O interface signal for compare/capture input and PWM output. SWA1 This pin also serves as one of the connection for analog switch A. The control of the analog switch is done by setting of ANEN of IOCFGP3.2.
P3.3	I/O, A	-	22	Port 3.3 GPIO 8051 P3.0 GPIO CEX5 PCA CCAP Module 5 This pin also can be configured as CEX pin for PCA CCP module 5. CEX is an I/O interface signal for compare/capture input and PWM output. SWA2 This pin also serves as one of the connection for analog switch A. The control of the analog switch is done by setting of ANEN of IOCFGP3.3.

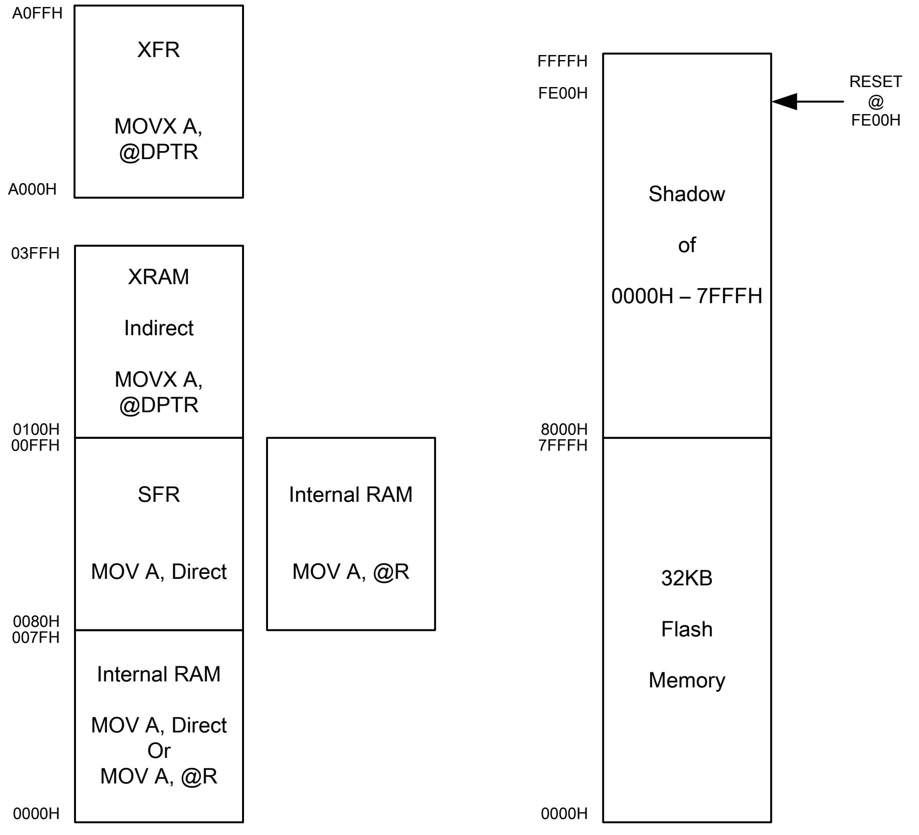
FUNCTIONAL DESCRIPTIONS

1. Introductions

The CPU core of CS8953 is compatible with the industry standard 8051, which includes 256 bytes RAM, Special Function Registers (SFR), two timers, 8 interrupt sources and 2 serial UART interface. The CPU core fetches its program code from the 32K bytes Flash in CS8953. The description of CPU Core is in Section 2, the descriptions of external peripherals are in Section3. Section 4 describes the Flash programming and ISP/IAP methodology.

1.1 Memory Map

There are total 256 bytes internal RAM in CS8953, the same as standard 8052. There are total 768 bytes auxiliary RAM allocated in the 8051 extended RAM area 100h – 03FFh. Programs can use "MOVX" instruction to access the AUXRAM. The 32KB embedded flash occupies the program address space from 0000h – 7FFFh, the 32KB is shadowed at 8000H – FFFFH. The CPU reset to address FE00H equivalent to 7E00H. This preserves the compatibility for 32KB or 64KB version of Myson's MCU products. The memory map is shown in the following figure.



DATA Memory MAP PROGRAM Memory MAP

1.2 SFR Map (0080h – 00FFh) and XFR Map (A000h – A07Fh)

The SFR address map maintains maximum compatibilities to most commonly used 8051 like MCU. The following table shows the SFR address map. Since SFR can be accessed by direct addressing mode, registers of built-in peripherals that requires fast access are mostly located in SFR. XFR is mainly used for on-chip peripherals configurations and control.

	0	1	2	3	4	5	6	7
0XF0	B				I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0XE0	ACC	CCON	CCAP3L	CCAP3H	CCAP4L	CCAP4H	CCAP5L	CCAP5H
0XD0	PSW	CMOD	CCAP0L	CCAP0H	CCAP1L	CCAP1H	CCAP2L	CCAP2H
0XC0	SCON1	SBUF1	SCON2	SBAUD2	PMR	STATUS	MCON	TA
0XB0	P3		CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4	CCAPM5
0XA0	P2	SPICR	SPIMR	SPIST	SPIDAT	SFIFO2	SBUF2	SINT2
0X90	P1	EXIF	WTST	DPX	CMPST	DPX1	PINT0EN	PINT0FG
0X80	P0	SP	DPL	DPH	DPL1	DPH1	DPS	PCON
	8	9	A	B	C	D	E	F
0XF0	EXIP	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
0XE0	EXIE	CH	MXAX	I2CSCON1	I2CSST1	I2CSADR1	I2CSDAT1	-
0XD0	WDCON	CL	DPXR	I2CSCON2	I2CSST2	I2CSADR2	I2CSDAT2	-
0XC0	T2CON	TB	RLDL	RLDH	TL2	TH2		
0XB0	IP	ADCCHSL	ADCAL	ADCAH	ADCBL	ADCBH	ADCCL	ADCCH
0XA0	IE	ADCCFG	ADCDL	ADCDH			PINT1EN	PINT1FG
0X90	SCON0	SBUF0		ESP		ACON		WKMASK
0X80	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CKSEL

	0	1	2	3	4	5	6	7
A000	REGTRM	IOSCITRM	IOSCVTRM	PLLM	PLLN	PLLCFG		XOSCCFG
A010	LVDCFG	LVDTHD						
A020	FLSHCMD1	FLSHDAT1	FLSHADH1	FLSHADL1		CNTPCTL1	CNTPCTH1	-
A030	CMPCFGAB	CMPCFGCD	CMPVTH0	CMPVTH1	PGACFG	PGAGAIN		VMIDCFG
A040	IOCFGP0.0	IOCFGP0.1	IOCFGP0.2	IOCFGP0.3	IOCFGP0.4	IOCFGP0.5	IOCFGP0.6	IOCFGP0.7
A050	MFCFGP0.0	MFCFGP0.1	MFCFGP0.2	MFCFGP0.3	MFCFGP0.4	MFCFGP0.5	MFCFGP0.6	MFCFGP0.7
A060	IOCFGP2.0	IOCFGP2.1	IOCFGP2.2	IOCFGP2.3	IOCFGP2.4	IOCFGP2.5	IOCFGP2.6	IOCFGP2.7
A070	MFCFGP2.0	MFCFGP2.1	MFCFGP2.2	MFCFGP2.3	MFCFGP2.4	MFCFGP2.5	MFCFGP2.6	MFCFGP2.7
	8	9	A	B	C	D	E	F
A000							RTCCMD	RTCDATA
A010	-	-			AMXCFGA	AMXCFGB		
A020	FLSHCMD2	FLSHDAT2	FLSHADH2	FLSHADL2		CNTPCTL2	CNTPCTH2	
A030	OPCFGA	OPCFGB	OPCFGC	OPCFGD	DACCFG	DACDAT1	DACDAT2	DACDAT3
A040	IOCFGP1.0	IOCFGP1.1	IOCFGP1.2	IOCFGP1.3	IOCFGP1.4	IOCFGP1.5	IOCFGP1.6	IOCFGP1.7

A050	MFCFGP1.0	MFCFGP1.1	MFCFGP1.2	MFCFGP1.3	MFCFGP1.4	MFCFGP1.5	MFCFGP1.6	MFCFGP1.7
A060	IOCFGP3.0	IOCFGP3.1	IOCFGP3.2	IOCFGP3.3	IOCFGP3.4	IOCFGP3.5	IOCFGP3.6	IOCFGP3.7
A070	MFCFGP3.0	MFCFGP3.1	MFCFGP3.2	MFCFGP3.3	MFCFGP3.4	MFCFGP3.5	MFCFGP3.6	MFCFGP3.7

The register names in gray are for those external peripherals not included in CS8953 and are reserved.
The register names in green are for those registers not standard in 8051/8052/80390.

2.0 Enhanced 8051 CPU Core

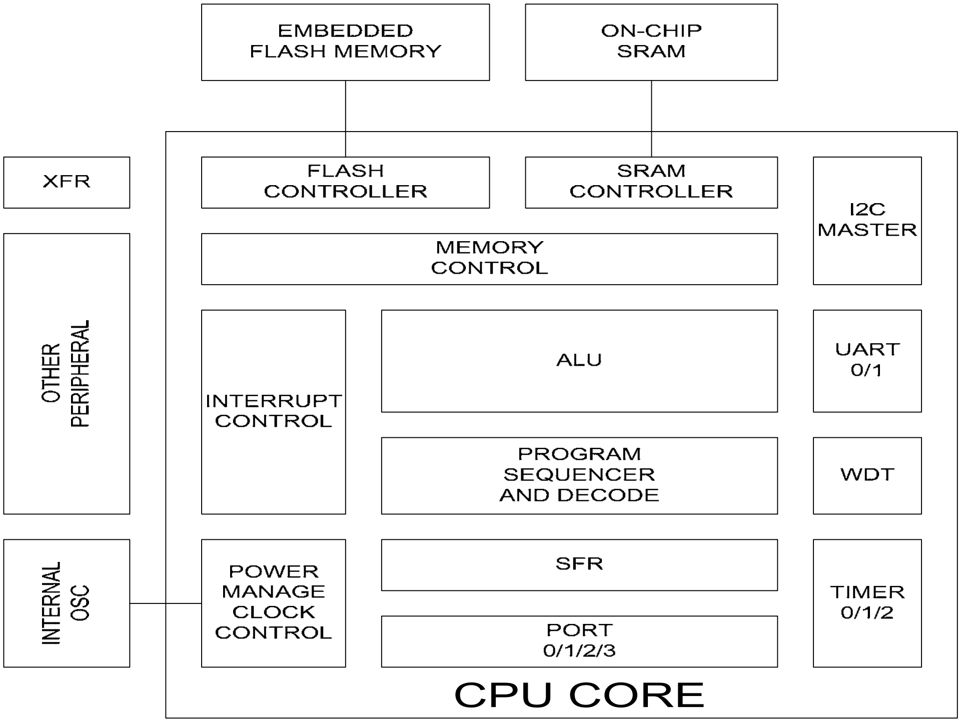
The CPU core is an enhanced version of standard 8051 used by series of Myson-Century MCU products. The CPU core is in RISC architecture and maintains binary instruction set compatible with the industry standard 8051. There is average 10 times performance enhancement in typical applications. The CPU operates at 20-bit addressing space that allows up to 1M bytes of program and data space for expansion. The CPU includes the following enhanced features compared with standard 8051.

- 16-bit LARGE addressing mode and 20-bit FLAT addressing mode control register ACON
- Two data pointers DPTR and DPTR1, and additional DPS, DPX, DPC1, MXAX registers for MOVX instruction
- 8-bit stack pointer for LARGE mode and 16-bit extended stack pointer for FLAT mode control register ESP
- Hardware Multiplication and Division Unit (MDU) provides 12 times faster performance using MD[5-0] and ARCON
- Programmable wait state for program space for on-chip flash memory using WTST register
- 256 Bytes of Direct Data Memory
- Enhanced Interrupt Controller allows 15 interrupt sources and 2 priority levels.
- Power Saving modes include IDLE mode, Power Management mode (PMM), and STOP mode. The PMM mode also support switchback features.
- Access Control of critical registers - TA, and TB registers.

In addition to standard 8052 peripherals, the CPU core also integrates the following peripherals. These peripherals are in same CPU clock domain. Thus they are described in this section.

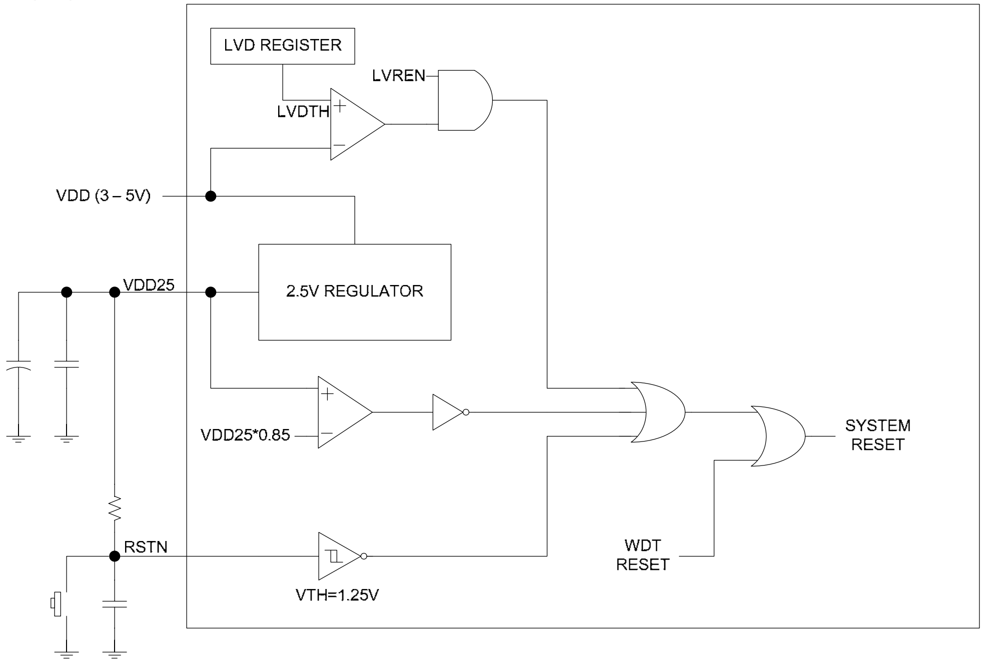
- Four 8-Bit I/O ports
- 16-bit Watch Dog Timer. WDT, WDCON, and CKCON registers
- Three 16-bit Timers, T0/T1 and T2. TCON, RLDL, RLDH, TL2, TH2, and T2CON registers
- UART0 and UART1.
- I^{2}C Master Controller. I2CMSA, I2CMCR, I2CMBUF, and I2CMTP registers.

The following sections describe in details of these enhanced features and peripherals. Assuming readers are familiar with 8051 standard operations and peripherals, the compatible functions will not be covered. The following shows the block diagram of the CPU and the relationship of other peripherals. The internal oscillator (IOSC) is tightly coupled to CPU as described later because it plays an important role as default clock after reset and is never disabled to maintain a constant accurate time base except in the STOP mode.



2.1 System Reset

After system reset, all registers have their default value. The default value is shown in the register description. The reset conditions include power on/off reset, external RSTN ping being pulled low, low supply voltage detection reset, and WDT reset. The block diagram illustrating these reset conditions is show in the following figure.



The power on/off reset is based on the detection of output level of the internal regulator. This detection also serves as the low-level detection of the core supply voltage level. The internal regulator output is 2.5V. And the output of the regulator should have external capacitors of a 0.1uF in parallel with 10uF for decoupling purpose. The larger the decoupling capacitor the better the decoupling effect to filter out high and low frequency noise. This is very critical for good analog peripheral performance and it also improve the EMI performance and enhance the noise immunity from EMC interference. The power on/off reset is asserted when the output of the 2.5V regulator has not reached or fallen below the 85% of its target value. In case of interference that the output of the regulator is disturbed and fall below its 85% level, the power on/off reset will also be asserted. This is different from the low-voltage detect reset as the low-voltage detection detects the level of the external supply of 3.0V to 5.5V. When this reset condition occurs, the RSTN pin is also forced low through internal strong pull-down device. To recover from reset state, the regulator output must be restored to within 85% of target value and then RSTN pin is pulled high by the external pull-up resistor charging the reset capacitor. This ensures a solid and extended reset condition when the voltage supply to the internal logic and flash memory is lower than rated level.

The external RSTN pin can also generate reset to the device. In typical applications, the RSTN should have a resister connected to the internal regulator output, and a capacitor to ground. For system containing hardware reset control, there is usually a button switch connecting RSTN pin to ground. When the switch is pressed causing RSTN to short to ground, and the device will enter into reset state. The RSTN logic has a built-in filter that will ignore any RSTN low duration less than 5usec. It is therefore recommended that RSTN needs to be actively pulled low for at least 50usec to guarantee a solid reset.

The low supply voltage detection circuit can also be configured to generate reset to the device. The detection circuit detects the level of the supply voltage (VDD) against a register controlled reference level. The output of the detection is controlled by LVREN register and if LVREN is set then when the supply voltage is lower than the register set threshold, a system reset will be generated. Please note LVREN is cleared after any reset so the default state of Low Voltage Reset is disabled. The software need to enable LVREN if low-voltage reset is desired.

The last reset source is from the watch dog counter. The WDT reset function is disabled after any reset condition. Software must enable the WDT function and its reset function to utilize WDT reset.

The program counter is loaded with 0xFE00 after reset. This differs from standard 8051. In typical cases, 0xFE00 will start Calibration and ISP boot codes then jump to 0x0000. The clock selection after reset is set to using internal oscillator automatically. The IOSC is disabled only in STOP mode.

2.2 Addressing Mode, Memory Operations and related registers

The LARGE mode addressing mode is compatible with standard 8051 in 16-bit address. FLAT mode extends the program address to 20-bit and expands the stack space to 16-bit data space. The data space is always 16-bit regardless of LARGE or FLAT mode.

ACON (0x9D) R/W (0x00) TA Protected

	7	6	5	4	3	2	1	0
RD	-	-	-	-	**DPXREN**	**SA**	**AM1**	**AM0**
WR	-	-	-	-	**DPXREN**	**SA**	**AM1**	**AM0**

ACON is address mode control register.

DPXREN DPXR register control bit. If DPXREN is 0, "MOVX, @Ri" instruction will use P2 (0xA0) register as XRAM Address [15-8]. If DPXREN is 1, XRAM Address [15-8] will be from DPXR (0xDA) register.

SA Extended Stack Address Mode Indicator. This bit is read only.
0 – 8051 standard stack mode where stack reside in internal 256 byte memory
1 – Extended stack mode. Stack pointer is ESP:SP in 16-bit addressing to data space.

AM1 and AM0 Address Mode Control Bits
00 – LARGE address mode in 16-bit
1x – FLAT address mode with 20-bit program address

ESP (0x9B) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**ESP.7**	**ESP.6**	**ESP.5**	**ESP.4**	**ESP.3**	**ESP.2**	**ESP.1**	**ESP.0**
WR	**ESP.7**	**ESP.6**	**ESP.5**	**ESP.4**	**ESP.3**	**ESP.2**	**ESP.1**	**ESP.0**

ESP is Extended Stack Pointer. It holds the high byte of the stack pointer in FLAT 20-bit address mode. The effective stack pointer is ESP:SP. SP is the standard 8051 stack pointer located at 0x81.

The clock speed of an MCU with embedded flash memory is usually limited by the access time of on-chip flash memory. While in modern process technology, the CPU can operate up to 100MHz to 200MHz, but the access time of flash memory is usually around 20 nanoseconds and thus limiting the clock rate up to less than 50MHz. To alleviate this problem, a programmable wait state function is incorporated to allow faster CPU clock rate and slow embedded flash memory. The wait state is controlled by WTST register as shown in the following.

WTST (0x92) R/W (0x07)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	**WTST3**	**WTST2**	**WTST1**	**WTST0**
WR	-	-	-	-	**WTST3**	**WTST2**	**WTST1**	**WTST0**

WTST[3:0] determines the program memory access wait state time. It is in CPU clock period unit.

WTST3	WTST2	WTST1	WTST0	Wait State Cycle
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

The default setting of the program wait state register after reset is 0x07 and the software must initialize the setting to change the wait state setting. For typical embedded flash, the read access time is specified as 30 nsec. Therefore the user program should set the WTST register according to the SYSCLK frequency. For example, using a SYSCLK of 4MHz, the WTST can be set to minimum because one clock period is 250 nsec which is longer than the embedded flash access time. If SYSCLK is above 33MHz, then WTST should be set higher than 1 to allow enough read access time.

2.3 Dual Data Pointers and MOVX operations

In standard 8051/8052, there is only one data pointers DPH:DPL to perform MOVX. The enhanced CPU provides 2nd data pointer DPH1:DPL1 to speed up the movement, or copying of data block. The active DPTR is selected and operation of DPTR is controlled by setting DPS (Data Pointer Select) register. Through the control DPS, efficient programming can be achieved

DPS (0x86) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**ID1**	**ID0**	**TSL**	**-**	**-**	**-**	**-**	**SEL**
WR	**ID1**	**ID0**	**TSL**	**-**	**-**	**-**	**-**	**SEL**

ID[1:0] Define the operation of Increment/Decrement functions of selected DPTR for INC DPTR instruction is executed.

ID1	ID0	SEL=0	SEL=1
0	0	INC DPTR	INC DPTR1
0	1	DEC DPTR	INC DPTR1
1	0	INC DPTR	DEC DPTR1
1	1	DEC DPTR	DEC DPTR1

TSL Enable toggling selection of DPTR selection. When this bit is set, the selection of DPTR is toggled when DPTR is used in an instruction and is executed.

SEL DPTR selection bit. Set to select DPTR1, and clear to select DPTR. SEL is also affected by the state of ID[1:0] and TSL after DPTR is used in an instruction. When read, SEL reflect the current selection state.

DPL1 (0x84) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**DPL1**							
WR	**DPL1**							

DPL1 register holds the low byte of data pointer 1.

DPH1 (0x85) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**DPH1**							
WR	**DPH1**							

DPH1 register holds the high byte of data pointer 1.

When accessing XRAM using "MOVX, @DPTR" instruction, the address of XRAM access is formed by DPHi:DPLi depending on which data pointer is selected. Another form of MOVX instruction is "MOVX, @Ri". This instruction provides an efficient programming method to move content within a 256 byte data block. In "@RI" instruction, the XRAM address [15-7] can be derived from two sources. If ACON.DPXREN = 0, the high order address [15-8] is from P2 (0xA0), if ACON.DPXREN = 1, the high order address is from DPXR (0xDA) register.

The maximum addressing space of XRAM is up to 16MB thus requiring 24 bit address. For "MOVX, @DPTR", the XRAMADDR [23-16] is from either DPX (0x93) or DPX1 (0x95) depending on which data pointer is selected. For "MOVX, @Ri", the XRAMUADDR [23-16] is from MXAX (0xEA) register.

2.4 Interrupt System

The CPU implements an enhanced Interrupt Control that allows total 15 interrupt sources and each with two programmable priority levels. The interrupts are sampled at rising edge of SYSCLK. If interrupts are present and enabled, the CPU will enter interrupt service routine by vectoring to the highest priority interrupt. Of the 15 interrupt sources, 7 of them are from CPU internal integrated peripherals, and 6 of them are for on-chip external peripherals, and 2 of them are used for external pin interrupt expansion. When an interrupt is shared, the interrupt service routine must determine which source is requesting the interrupt by examining the corresponding interrupt flag of sharing peripherals.

The following table shows the interrupt sources and corresponding interrupt vectors. The Flag Reset column shows whether the corresponding interrupt flag is cleared by hardware (self cleared), or needs to be cleared by software. Please note the software can only clear the interrupt flag not set the interrupt flag. The Natural Priority column shows the inherent priority if more than one interrupts are assigned to the same priority level. Please note that the interrupts assigned with high priority level will always get serviced first compared with interrupts assigned with low priority regardless of the natural priority sequence.

Interrupt	Peripheral Source Description	Vectors	FLAG RESET	Natural Priority
PINT0	Expanded Pin INT0.x	0x03	Software	1
TF0	Timer 0	0x0B	Hardware	2
PINT1	Expanded Pin INT1.x	0x13	Software	3
TF1	Timer 1	0x1B	Hardware	4
TI0/RI0	UART0	0x23	Software	5
TF2	Timer 2	0x2B	Software	6
TI1/RI1	UART1	0x33	Software	7
I2CM	I2C Master	0x3B	Software	8
INT2	LVT	0x43	Software	9
INT3	Comparator (A-D)	0x4B	Software	10
INT4	ADC (A-D)	0x53	Software	11
WDIF	Watchdog	0x5B	Software	12
INT6	PCA	0x63	Software	13
INT7	SPI/I2CS/EUART	0x6B	Software	14
INT8	RTC	0x73	Software	15

The interrupt related registers are listed in the following. Each interrupt can individually enabled or disabled by setting or clearing corresponding bit in IE and EXIE and integrated peripherals' control registers.

IE (0xA8) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**EA**	**ES1**	**ET2**	**ES0**	**ET1**	**INT1EN**	**ET0**	**INT0EN**
WR	**EA**	**ES1**	**ET2**	**ES0**	**ET1**	**INT1EN**	**ET0**	**INT0EN**

EA	Global Interrupt Enable bit.
ES1	UART1 Interrupt Enable bit.
ET2	Timer 2 Interrupt Enable bit.
ES0	UART0 Interrupt Enable bit.
ET1	Timer 1 Interrupt Enable bit.
INT1EN	INT1 interrupt enable. INT1 is used for expanded pin interrupt PINT1.x.
ET0	Timer 0 Interrupt Enable bit.
INT0EN	INT0 interrupt enable. INT0 is used for expanded pin interrupt PINT0.x

EXIE (0xE8) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**EINT8**	**EINT7**	**EINT6**	**EWDI**	**EINT4**	**EINT3**	**EINT2**	**EI2CM**
WR	**EINT8**	**EINT7**	**EINT6**	**EWDI**	**EINT4**	**EINT3**	**EINT2**	**EI2CM**

EINT8	RTC Interrupt Enable bit.
EINT7	SPI, I2C Slave, and EUART Interrupt Enable bit

EINT6 PCA interrupt Enable bit
EWD1 Watchdog Timer Interrupt Enable bit.
EINT4 ADC Interrupt Enable bit.
EINT3 Analog Comparator Interrupt Enable bit.
EINT2 Low Voltage Detection Interrupt Enable bit.
EI2CM I2C Master Interrupt Enable bit.

And each interrupt can be individually assigned to one of two priorities either high or low. When the corresponding bit is set to 1, it indicates it is at high priority.

IP (0xB8) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**-**	**PS1**	**PT2**	**PS0**	**PT1**	**PX1**	**PT0**	**PX0**
WR	**-**	**PS1**	**PT2**	**PS0**	**PT1**	**PX1**	**PT0**	**PX0**

PS1 UART 1 Priority bit.
PT2 Timer 2 Priority bit.
PS0 UART 0 Priority bit.
PT1 Timer 1 Priority bit.
PX1 Pin Interrupt INT1 Priority bit.
PT0 Timer 0 Priority bit.
PX0 Pin Interrupt INT0 Priority bit.

EXIP (0xF8) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**EINT8-**	**EINT7**	**EINT6**	**EWDI**	**EINT4**	**EINT3**	**EINT2**	**EI2CM**
WR	**EINT8-**	**EINT7**	**EINT6**	**EWDI**	**EINT4**	**EINT3**	**EINT2**	**EI2CM**

EINT8 INT8 RTC Priority bit.
EINT7 INT7 SPI, I2C Slave, and EUART Priority bit.
EINT6 INT6 PCA Priority bit.
EWDI Watchdog Priority bit.
EINT4 INT4 ADC Priority bit.
EINT3 INT3 Analog Comparator Priority bit.
EINT2 INT2 Low Voltage Detection Priority bit.
EI2CM I2C Master Priority bit.

EXIF (0x91) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**INT8F**	**INT7F**	**INT6F**	**-**	**INT4F**	**INT3F**	**INT2F**	**I2CMIF**
WR	**INT8F**	**INT7F**	**INT6F**	**-**	**INT4F**	**INT3F**	**INT2F**	**I2CMIF**

INT8F INT8 RTC Interrupt flag.
INT7F INT7 SPI, I2C Slave and EUART interrupt flag.
INT6F INT6 PCA Interrupt flag.
INT4F INT4 ADC Interrupt flag.
INT3F INT3 Analog Comparator Interrupt flag.
INT2F INT2 Low Voltage Detection Interrupt flag.
I2CMIF I2C Master Interrupt flag. Must be cleared by software
Writing to INT2F to INT8F has no effect.

The interrupt flag of internal peripherals are stored in the corresponding flag registers in the peripheral and IXIF registers. These peripherals include T0, T1, T2, and WDT. Therefore to clear the interrupt flag, the software just need to clear the corresponding flag located in the peripheral (for T0, T1, and T2, and WDT). For I2CM, the interrupt flag is located in the EXIF register bit I2CMIF. This also needs to be cleared by software.

INT2 to INT8 are used to connect to the external peripherals. INT2F to INT8F are direct equivalent of the interrupt flag from the corresponding peripherals. These peripherals include RTC, I2CS, PCA, ADC, etc. For RTC example, there is an interrupt flags (SECINT, ALMINT, and 4HZINT) in RTCCMD register. Because RTC interrupt is connected to INT8, INT8F is set to one when one or more than one of SECINT, ALMINT, and

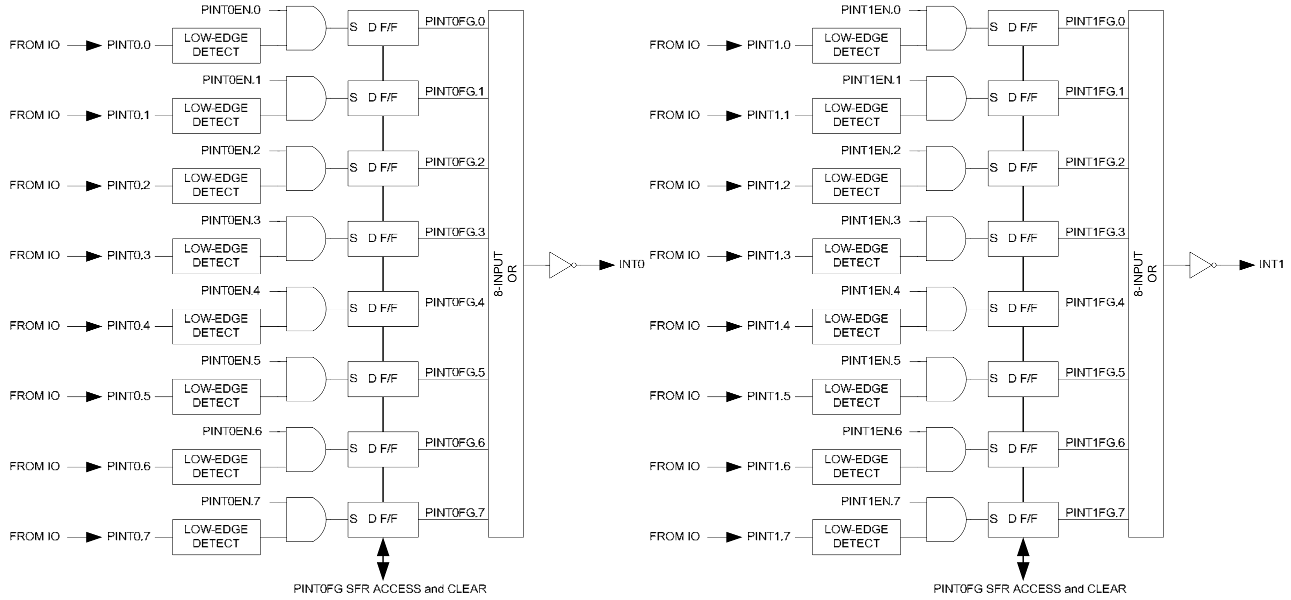
4HZINT of RTC is set, i.e., INT8F = (SECINT + ALMINT + 4HZINT). The software need to clear the origin of the interrupt flag bit in the RTC before exiting the service routine. In RTC's example, if the service routine only clear one interrupt flag, for example, SECINT, and if ALMINT is also set, then after exiting, ALMINT will still set INT2F and result an re-entry of the interrupt service routine. And then the service routine can take care of ALMINT.

TCON (0x88) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**TF1**	**TR1**	**TF0**	**TR0**	**INT1F**	**-**	**INT0F**	**-**
WR	**TF1**	**TR1**	**TF0**	**TR0**	**INT1F**	**-**	**INT0F**	**-**

TF1	Timer 1 overflow Interrupt Flag. TF1 is cleared by hardware when entering ISR.
TR1	Timer 1 Run Control. Set to enable Timer 1 and clear to disable Timer 1.
TF0	Timer 0 overflow Interrupt Flag. TF0 is cleared by hardware when entering ISR.
TR0	Timer 0 Run Control. Set to enable Timer 0 and clear to disable Timer 0.
INT1F	Expanded INT1 Interrupt Flag. INT1 is expanded to PIN1.x. This flag reflect the combined PINT1FG.x OR-ed together and latched result. To clear this bit, the software must clear the corresponding PINT1FG.x bit causing the interrupt first. After this is done, this bit must be cleared by the software.
INT0F	Expanded INT0 Interrupt Flag. INT0 is expanded to PIN0.x. This flag reflect the combined PINT0FG.x OR-ed together and latched result. To clear this bit, the software must clear the corresponding PINT0FG.x bit causing the interrupt first. After this is done, this bit must be cleared by the software.

PINT0 and PINT1 are used for expanded Pin Interrupt, PINT0.x and PINT1.x, and need special attention. The expansion is shown in the following diagram.



PINT0EN (0x96) RW (0x00)

	7	6	5	4	3	2	1	0
RD	**PINT0EN.7**	**PINT0EN.6**	**PINT0EN.5**	**PINT0EN.4**	**PINT0EN.3**	**PINT0EN.2**	**PINT0EN.1**	**PINT0EN.0**
WR	**PINT0EN.7**	**PINT0EN.6**	**PINT0EN.5**	**PINT0EN.4**	**PINT0EN.3**	**PINT0EN.2**	**PINT0EN.1**	**PINT0EN.0**

PINT0EN register control the enable and disable setting of the expanded INT0 from PINT0.x interrupt functions.

PINT0FG (0x97) W (0x00)

	7	6	5	4	3	2	1	0
RD	**PINT0FG.7**	**PINT0FG.6**	**PINT0FG.5**	**PINT0FG.4**	**PINT0FG.3**	**PINT0FG.2**	**PINT0FG.1**	**PINT0FG.0**
WR	**PINT0FG.7**	**PINT0FG.6**	**PINT0FG.5**	**PINT0FG.4**	**PINT0FG.3**	**PINT0FG.2**	**PINT0FG.1**	**PINT0FG.0**

PINT0FG register holds the external pin interrupt status flag. The individual bit is set at the falling or rising edge of the external pin signal. The flag must be cleared by the software.

PINT1EN (0xAE) W (0x00)

	7	6	5	4	3	2	1	0
RD	PINT1EN.7	PINT1EN.6	PINT1EN.5	PINT1EN.4	PINT1EN.3	PINT1EN.2	PINT1EN.1	PINT1EN.0
WR	PINT1EN.7	PINT1EN.6	PINT1EN.5	PINT1EN.4	PINT1EN.3	PINT1EN.2	PINT1EN.1	PINT1EN.0

PINT1EN register control the enable and disable setting of the expanded INT1 from PINT1.x interrupt functions.

PINT1FG (0xAF) W (0x00)

	7	6	5	4	3	2	1	0
RD	PINT1FG.7	PINT1FG.6	PINT1FG.5	PINT1FG.4	PINT1FG.3	PINT1FG.2	PINT1FG.1	PINT1FG.0
WR	PINT1FG.7	PINT1FG.6	PINT1FG.5	PINT1FG.4	PINT1FG.3	PINT1FG.2	PINT1FG.1	PINT1FG.0

PINT1FG register holds the external pin interrupt status flag. The individual bit is set at the falling edge of the external pin signal. The flag must be cleared by the software.

PINT0FG and PINT1FG are the interrupt flag register while the PINT0EN and PINT1EN are the interrupt enable control for the expansion interrupt INT0 and INT1. PINT0.[7-0] and PINT1.[7-0] use edge detect to latch the interrupt condition and the outputs of the DFF are OR together and sent to INT0 and INT1. The combined results are further latched into INT0F and INT1F in TCON register. Therefore to clear the interrupt flag, the corresponding PINTxFG bit needs to be cleared by the software first, and INT0F or INT1F bit should be cleared by the software too. Reversing the sequence of flag clearing may recursive entry of ISR.

PINT0.[7-0] and PINT1.[7-0] are connected to individual pin's input buffer and this is defined according to the pin arrangement of the packages and the multi-function pin definitions. The selection of positive or falling edges of PINT0.x and PINT1.x is defined in the corresponding PINCFG registers.

When a satisfying edge condition occurs on an external pin, the corresponding PINTxFG.x in PINTxFG register is set. This will cause INT0F or INT1F be set and latched in TCON register and causes the CPU to branch to the interrupt service routine. The service routine can read PINTxFG to determine which external pin caused the interrupt and take appropriate action, and then clear the corresponding PINTxFG.x flag bit.

If more than one PINTxF.x flag is set, then the un-cleared flag bits will cause the reentry of the interrupt and the service routine should take service action one by one to clear each pin interrupt request.

2.5 Register Access Control

One important aspect of the embedded MCU is its reliable operations under harsh environment. Many system failures results from the accidental loss or changes of critical registers or program and data contents and this may lead to catastrophic effects. The CPU provides several mechanisms of protections which are described in this section.

TA (0xC7) Time Access A Control Register2 WO xxxxxxx0

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	TA Register							

TA access control emulates a ticket that must be purchased before modifying a critical register. To modify or write into a TA protected register, TA must be accessed in a predefined sequence to obtain the ticket. The ticket is used when an intended modification operation is done to the TA protected register. To obtain a next access a new ticket must be obtained again by performing the same predefined sequence on TA. TA does not limit the read access of the TA protected registers. The TA protected register include WDCON (0xD8), MCON (0xC6), and ACON (0x9D) registers. The following predefined sequence is required, for example, to modify the content of MCON.

```
MOV TA, #0xAA;
MOV TA, #0x55;
MOV MCON, #0x01;
```

Once the access is granted, it is valid until used. There is no time limitation of the access. The access is voided if any operation is performed on TA address.

TB (0xC9) Time Access B Control Register2 RW (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	TBSTAT
WR	TB Register							

TB access control functions similar to TA control except the ticket is a multiple use ticket with time limit. Once access is granted, the access is open for 256 clock periods and then expires. The software can also read TB address to obtain the current TB status. To modify registers with TB protection, the following procedure must be performed.

```
    MOV TB, #0xAA
    MOV TB, #0x55
```

This action creates a timed window of 256 SYSCLK periods to allow write access of these TB protected registers. If another above operation sequence is repeated before the 128 cycles expires, a new 128 cycles will be extended.

The current 256 cycles can be terminated immediately by writing #0x00 to TB registers, such as

```
MOV TB, #0x00
```

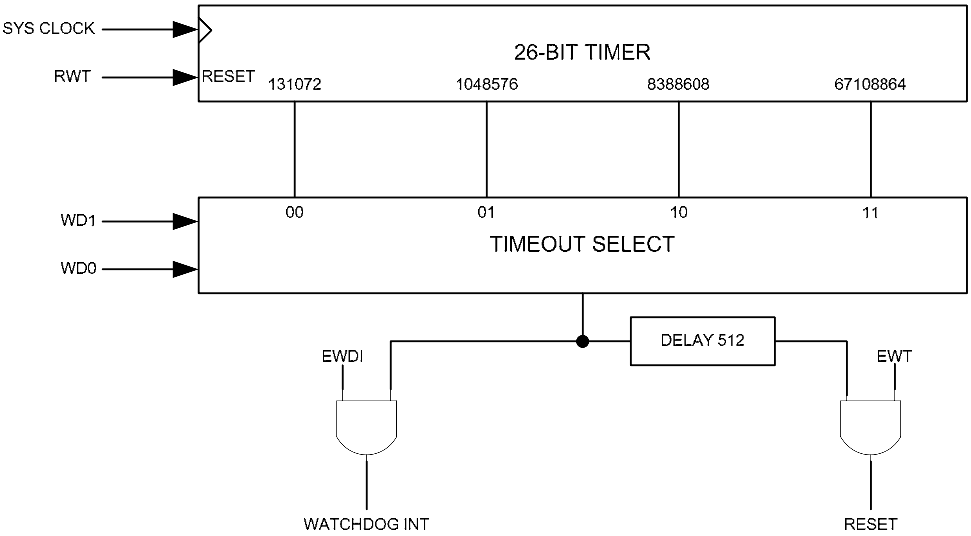
It is recommended to terminate the TB access window once user program finished the modifications of TB protected registers.

Because TA and TB are critical protection of the reliable operation of the MCU and prevent accidental hazardous uncontrollable modification of critical register, the operation of these two register should bear extreme cautions. It is strongly advised that these two register should not be turned-on unnecessarily. Because both register uses synchronous CPU clock, it is also important that TA and TB access is terminated before entering IDLE or STOP modes. Both of these two modes will turn off CPU clock, and if TA and TB are enabled, they will stay enabled until CPU clock is resumed thus creating vulnerabilities for those critical registers.

Another reliability concern of embedded Flash MCU is the important content on the Flash can be accidentally erased. This concern is addressed by the content protection in the Flash controller.

2.6 Watchdog Timer

The Watchdog Timer is a 26-bit timer can be used for system supervisor or an event timer. The time out of Watchdog timer can be used to generate an interrupt or to issue a system reset depending on the control settings. This section describes the register related to the operation of Watchdog Timer, and the use of the Watchdog Timer. The following diagram shows the structure of the Watchdog Timer. Note WDT uses the same clock as CPU, thus WDT is also stopped in IDLE or STOP modes, and in reduced rate at PMM mode.



WDCON (0xD8) R/W (0x00) TA Protected

	7	6	5	4	3	2	1	0
RD	**-**	**-**	**-**	**-**	**WDIF**	**WTRF**	**EWT**	**RWT**
WR	**-**	**-**	**-**	**-**	**WDIF**	**WTRF**	**EWT**	**RWT**

WDIF WDT Interrupt Flag. This bit is set when time-out expires regardless whether WDT interrupt is enabled or not. Note the WDT interrupt enable control is located in EIE (0xE8).4 EWDI bit. Must be cleared by software

WTRF WDT Reset Flag. When set by hardware indicates that a WDT reset event has occurred. Set by software does not generate a reset. Cleared by software or by external reset generated by RSTN pin

EWT Enable Watchdog Reset. Set this bit to enable the watchdog reset function.

RWT Reset the Watchdog timer. Setting RWT will reset the timer count.

CKCON (0x86) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**WD1**	**WD0**	**-**	**-**	**-**	**-**	**-**	**-**
WR	**WD1**	**WD0**	**-**	**-**	**-**	**-**	**-**	**-**

This register controls the time out value of WDT as the following table. The value is in number of CPU clock.

WD1	WD0	Time Out Value
0	0	131072
0	1	1048576
1	0	8388608
1	1	67108864

2.7 Clock Control and Power Management Modes

This section describes the clock control and power saving modes of the CPU and its integrated peripherals. The settings are controlled by PCON (0x87) and PMR (0xC4) registers. The register description is defined as following.

PCON (0x87) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**SMOD0**	**SMOD1**	**-**	**-**	**-**	**-**	**-**	**-**
WR	**SMOD0**	**SMOD1**	**-**	**-**	**-**	**-**	**STOP**	**IDLE**

SMOD0 UART 0 Baud Rate Control. This is used to select double baud rate in mode 1, 2 or 3 for UART0 using Timer 1 overflow. This definition is the same as standard 8051.

SMOD1 UART 1 Baud Rate Control. This is used to select double baud rate in mode 1, 2 or 3 for UART1 using Timer 1 overflow. This definition is the same as standard 8051.

STOP Stop bit. When set the clock goes to the CPU and all peripherals is stopped and enter into STOP mode. The STOP mode can only be waked up by non-clocked interrupt or reset. Upon exiting STOP mode, STOP bit in PCON is automatically cleared.

IDLE Idle bit. If the IDLE bit is set, the system goes into IDLE mode. In Idle mode, the CPU clock is stopped, and the CPU and all CPU integrated peripherals are stopped, and these integrated peripherals include WDT, T0/T1/T2, and UART0/UART1. But the clock goes to peripherals external to CPU are still active, such as PCA, ADC, UART2, SPI, I2C slave and others. This allows the interrupts generated by these peripherals and external interrupts to wake up the CPU. The exit mechanism of IDLE mode is the same as STOP mode. IDLE bit is automatically cleared at the exit of the IDLE mode..

PMR (0xC4) R/W (010xxxxx)

	7	6	5	4	3	2	1	0
RD	**CD1**	**CD0**	**SWB**	**-**	**-**	**-**	**-**	**-**
WR	**CD1**	**CD0**	**SWB**	**-**	**-**	**-**	**-**	**-**

CD1 and CD0 Clock divide bits. These two bits control the entry of PMM mode. When CD0=1, and CD1=0, full speed operation is in effect. When CD0=1, and CD1=1, the CPU enters into PMM mode where CPU and its integrated peripherals operate at a clock rate divided by 256. Note that in PMM mode, all integrated peripherals such as UART0, UART1, WDT, and T0/T1/T2 will run at this reduce rate, thus may not function properly. All peripherals external to CPU still operate at full speed in PMM mode.

SWB Switch Back Control bit. Setting this bit allows the action occurs in integrated peripherals to cause automatically switching back to normal operation mode.

STATUS (0xC5) RO (0x00)

	7	6	5	4	3	2	1	0
RD	**-**	**HIP**	**LIP**	**-**	**SPTA1**	**SPRA1**	**SPTA0**	**SPRA0**
WR	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

STATUS register can be accessed by program to determine the status of critical events occurring in the integrated peripherals. Program should check these status conditions before entering into STOP, IDLE, or PMM modes to prevent loss of intended functions by delaying the entry until these events are finished.

HIP High Priority Interrupt Status. This bit reads 1 when there is high priority interrupt processing.

LIP Low Priority Interrupt Status. This bit reads 1 when there is low priority interrupt processing.

SPTA1 UART1 Transmit Activity Status. This bit reads 1 when UART1 transmission is active.

SPRA1 UART1 Receive Activity Status. This bit reads 1 when UART1 receive is active.

SPTA0 UART Transmit Activity Status. This bit reads 1 when UART0 transmission is active.

SPRA0 UART0 Receive Activity Status. This bit reads 1 when UART0 receive is active.

CKSEL (0x8F) R/W (0x00) System Clock Selection Register TB Protected

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	**CLKSEL1**	**CLKSEL0**
WR	-	-	-	-	-	-	**CLKSEL1**	**CLKSEL0**

CLKSEL1 and CLKSEL0 define the clock source of the SYSTEM, SYCLK. There are possible four selections depending on the MCU implementations and the default is from IOSC. There are two CLOCK domains, CPUCLOCK and PPHCLOK. They are always derived from the same SYSCLK. The CPUCLK is used to clock for the CPU core, and PPHCLOCK is used to clock external peripherals. The clock source selection of SYSCLK is determined by the following table.

CLKSEL1	CLKSEL0	SYSCLK
0	0	IOSC
0	1	XOSC
1	0	RTC
1	1	PLL

Because CS8953 does not include on-chip PLL clock synthesizer, the setting of CLKSEL=11 is a reserved setting. Setting clock selection to be PLL clock will cause unpredictable behavior.

WKMASK (0x9F) R/W (0xFF) Wake Up Mask Register TB Protected

	7	6	5	4	3	2	1	0
RD	**WEINT8-**	**WEINT7-**	**WEINT6**	**WEINT4**	**WEINT3**	**WEINT2**	**WEINT1**	**WEINT0**
WR	**WEINT8-**	**WEINT7-**	**WEINT6**	**WEINT4**	**WEINT3**	**WEINT2**	**WEINT1**	**WEINT0**

This register defines the wake up control of the interrupt signals from the STOP mode. The wake up is performed by these interrupts and if enabled will first turn on the internal oscillator and resume CPUCLK. Although the interrupt can be set as level trigger or edge trigger, the wake up is always based on the edge. Please note the wake up control is implemented separately from the interrupt logic, therefore, after wake up CPU does not necessary enter the interrupt service routine if the corresponding interrupt is not enabled. In this case, the CPU just continues onto the next instruction after the instruction that initiates the STOP modes. The software need to either maintain the consistency.

WEINT8	Set this bit to allow INT8 to trigger the wake up of CPU from STOP modes.
WEINT7	Set this bit to allow INT7 to trigger the wake up of CPU from STOP modes.
WEINT6	Set this bit to allow INT6 to trigger the wake up of CPU from STOP modes.
WEINT4	Set this bit to allow INT4 to trigger the wake up of CPU from STOP modes.
WEINT3	Set this bit to allow INT3 to trigger the wake up of CPU from STOP modes.
WEINT2	Set this bit to allow INT2 to trigger the wake up of CPU from STOP modes.
WEINT1	Set this bit to allow INT1 to trigger the wake up of CPU from STOP modes.
WEINT0	Set this bit to allow INT0 to trigger the wake up of CPU from STOP modes.

Please also note that all clocks are stopped in STOP mode, therefore peripherals requiring clock such as I2C slave, UART, ADC, LVD can not perform wake up function. Only external pins and peripherals that do not require clock can be used for wake up purpose. Such peripheral in CS8953 are analog comparator and RTC.

PMM mode

PMM mode is enabled by setting CD[1:0] bits in PMR register to both 1. In PMM mode, the CPU and its integrated peripheral such as WDT, UART0, UART1, T0/T1/T2, and I2C Master will operate at 256 times slower than SYSCLK. All other external peripherals such as PCA, ADC, etc. are still operating under normal clock. The PMM mode saves power from CPU and internal Flash memory and SRAM by operating at much slower frequency. Since CPU is still clocked, the program continues to operate and CPU is at reduced performance. To further save power, unused external peripherals can be turned off or disabled.

Normal mode operation can be recovered from PMM mode by program itself that set CD[1:0] = 01. Another way of recovery is to enable the SWITCHBACK function by setting SWB bit to high in PMR register. When switchback is enabled, the following conditions will trigger the CPU to exit PMM mode and resume normal operations.

External Interrupt INT0/1/2/3/4/6/7/8 and any external peripherals interrupt ORed with these interrupts.

UART0/1 receive start bit detection

UART0/1 transmit buffer loaded

When external interrupt is intended to be used to perform switchback, the corresponding interrupt must be enabled and not blocked by higher priority interrupts. In the case of UART triggered switchback, the triggering is not generated by the UART associated interrupt. This is because UART operating under PMM mode will not necessarily operate correctly to receive a byte or transmit a back. The switchback is thus initiated by the reception of the falling edge of the start bit. The UART receive switchback is enabled only if the associated receive enable bit (SCON0.4 or SCON1.4) is set. The UART transmit initiated switchback is triggered when UART transmit buffer is loaded. Thus CPU operating under PMM mode will recover to normal mode automatically when it writes into the transmit buffer. Once recovered, the UART will operate under normal frequency to correctly transmit the data.

The return of PMM mode after switchback must be performed by software manually. The PMM mode is exited also after WDT reset or external RSTN reset.

Since the purpose of the PMM mode is to save power consumption, the internal oscillator clock IOSC is recommended to be used as system clock as IOSC consumes significant less power than the crystal oscillator.

IDLE MODE

IDLE mode provide a further power saving than PMM mode by stopping the clock for CPU and its integrated peripherals but keeps the external peripherals at normal operating conditions. Because the external peripherals still functions correctly and thus can generate interrupts that wake up the CPU from IDLE mode. The IDLE mode is entered by setting IDLE bits 1.

In IDLE mode, because CPU is stopped, no processing is possible. All integrated internal peripherals such T0/T1/T2, UART0 and UART1, and I2C Master are all stopped. The IDLE mode can be exited by hardware reset through RSTN pin or by external pin interrupts as well as the interrupts from external peripherals. The triggering external interrupts have to be in an enabled state to be effective. Upon exiting from IDLE mode, the CPU will resume operation as the clock is turned back on. CPU will immediately vector to the interrupt service routine of the corresponding interrupt source that wake up the CPU. When the interrupt service routine completes, RETI returns the program to the instruction immediately following the one that invoked the IDLE mode. Upon return from IDLE mode to normal mode, IDLE bit in PCON is automatically cleared by hardware.

As the purpose of the IDLE mode is to save power, it is strongly recommended that the system to switch using IOSC clock for SYSCLK before entering IDLE mode. IOSC clock generator consumes significantly less power than the crystal oscillator or other clock sources.

STOP MODE

STOP mode provides the lowest power consumption by stopping clocks to all components in the system. STOP mode is entered by setting STOP=1. To achieve minimum power, before entering the STOP mode, it is essential to turn off all peripherals and the current consuming clock oscillators such as crystal oscillator and PLL. It is also important that the software switches into IOSC clock and disable all other clock generator such as crystal oscillator or PLL clock generator before entering the STOP mode. This is critical to ensure a smooth transition during resuming normal operations when the CPU is wake up later. Selecting other clock sources, such as XTAL oscillator or PLL clock as CPU system clock can put the system in chaotic condition as during wake up process these clock sources take significant amount of time to stabilize as they are turned on. When entering STOP mode, the system will use the last edge of IOSC clock to shut down IOSC clock generator by itself. Thus minimum power consumption state is achieved.

STOP mode is exited by hardware reset through RSTN pin or by interrupts generated by external pins (INT0 and INT1), and interrupt generated by INT2 to INT8. Because all clocks are stopped, all peripherals that requires clock will not cause the exit of STOP mode, these peripherals include UART, Timers, I2C master and slave, ADC, or LVD. Analog comparator and RTC interrupt can be used to exit STOP mode as they are implemented asynchronously or has its own clock source.

The triggering interrupt source must be enabled and its wake up bit set in the WKMASK register. For external pin triggering, the triggering event is LOW level triggered. For those on-chip external peripherals, INT flag needs to be HIGH to trigger. Upon the triggering event, the IOSC circuit is activated and the clock generated will revive the CPU at the first IOSC clock edge. Please note that the clock selection is forced to IOSC when STOP mode is exiting. As CPU resumes the normal operation using IOSC clock, the CPU will immediately vector to the interrupting service routine of the corresponding interrupt source, if the interrupt is enabled. When the interrupt service routine completes, RETI returns the program to the instruction immediately following the one that invoked the STOP mode. The STOP bit in PCON is automatically cleared by hardware during the wake up transition.

Please note the wake up control WKMASK register and interrupt enable register IE and EXIE are separately controlling the wake up and interrupt. The software should pay attention on these setting to get coherent setting for application purpose. Please also note that in STOP mode, all clocks to CPU and peripherals are stopped (except RTC), therefore only external pins and peripherals such as analog comparator and RTC that do not require clock can be used to cause the wake up. Peripherals such as UART, Timers, I2C master and slave, or ADC, and LVD can not generate wake up interrupt.

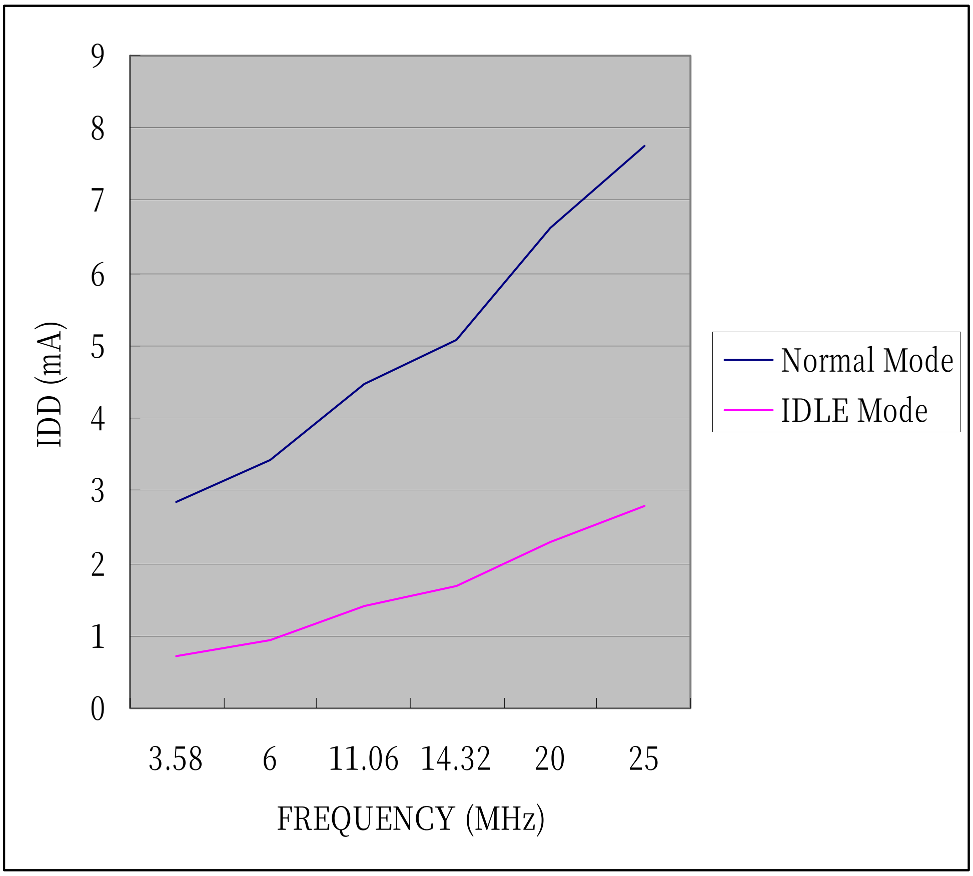
CLOCK CONTROL

The clock selection is defined by CKSEL register (0x8F). IOSC is a critical component in MCU although not integrated in the CPU core. It is never stopped and disabled except in STOP mode. IOSC also handles critical timing conformance for flash programming, the default manufactured calibrated IOSC is set at 3.68MHz. Although user can arbitrarily set IOSC frequency but it should not be varied more than 50% away from its typical value to avoid flash performance.

It is recommended that IOSC should be used for transition between changing clock sources to ensure a smooth and glitch-less clock switching. This is also true for switching among different power saving modes. Please note that when waking up from the STOP mode, the clock selection will be switched automatically to IOSC. The software need to change the clock selection if other clock sources are preferred.

When switching clock sources, it is also important to note the oscillators for crystal, real time clock and the phase lock loop, takes significant amount of time to stabilize. Therefore software needs to turn on the corresponding clock source first and wait for the stabilization time before actually change CKSEL settings.

The typical power dissipation relationship to the CPU frequency is shown in the following graph.



The IDD does not include the power dissipation of the clock oscillator. The graph shows that for normal operation, the power dissipation increases approximately at 0.6mA/MHz, and in idle mode (the power dissipation still increases as the frequency increases due to presence of peripheral clock) increases at about 0.1mA/MHz.

The last and most important note to the software is that when switching to a clock source that is not available will cause the system to hang. There is no hardware protection against this happening. Therefore for any application, the programmer must be very careful and exert extreme precautions for doing clock source change.

2.8 FLASH CONTROLLER

The flash controller connects the CPU to the on-chip embedded FLASH memory. The FLASH memory can function as the program storage as well as non-volatile data storage. The program access of the FLASH does not require any special attention. When the FLASH is used as data storage, the software must go through commands issued to the FLASH controller through the XFR registers. And when the FLASH controller is processing these commands, CPU is held idle until the command is completed. Depending on the embedded FLASH memory configurations, up to two FLASH memories is possible. In this case, CPU is possible to continue to have program access on one flash while the other flash controller is executing its command.

The commands can be performed by a Flash Controller are defined in FLSHCMD registers. These operations defined allow the user program to use on-chip flash as a program memory, and non-volatile data memory, as well as In-System-Programming and In-Application-Programming. And maximum flexible use of the on-chip flash memory is possible depending on user program. The manufacture provides a default ISP boot program located on the top sectors of the flash. The user program can inherit the ISP boot program or modify or replace it according to application requirements.

FLSHCMD1 (A020h) FLSHCMD2 (A028h) Flash Controller Command Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	**RDYEN**	**BUSY**	**FAIL**	**CMD4**	**CMD3**	**CMD2**	**CMD1**	**CMD0**
WR	**RDYEN**	**-**	**-**	**CMD4**	**CMD3**	**CMD2**	**CMD1**	**CMD0**

RDYEN RDYEN=1 will use RDY to stall CPU. RDYEN=0 will not use RDY to stall CPU. This bit is not used in single Flash configuration. For single flash, this bit should be tied to low by HW.

BUSY Busy Status. Read Only.
This bit indicates the busy status of Flash Controller. It is set to 1 when Flash Controller is performing commands and can not respond. This bit is not useful for single Flash configuration.

FAIL Command Execution Result. It is set if the previous Flash command execution failed due to any reasons. It is recommended program should verify the command success after issuing command to the Flash controller. It is not cleared by reading and is refreshed when a new command is issued.

CMD4 Byte Read Command
Set CMD4 will ask Flash Controller to execute a Byte Read of the Flash addressed by FLSHADH and FLSHADL. The read data is put on the FLASHDAT register. ~~This command is not implemented in this version.~~ To read a byte from Flash, MOVC command can also be used.

CMD3 Sector Erase Command
Set CMD3 will ask Flash Controller to execute Sector Erase to the Flash. The Sector address is stored in FLSHADH.

CMD2 Byte Write Command
Set CMD2 will ask Flash Controller to execute Byte Write to the Flash. The Byte Address is stored in FLSHADH and FLSHADL registers. The Byte data is stored in FLSHDAT register.

CMD1 IFB Byte Read Command
Set CMD1 will ask Flash Controller to execute Byte read to the Flash IFB. The IFB Byte Address is stored in FLSHADL. And the data read is put on FLSHDAT after the command executed successfully.

CMD0 IFB Byte Write Command
Set CMD0 will ask Flash Controller to execute Byte Write to the Flash IFB. The IFB Byte Address is stored in FLSHADL. And write data is pre-stored in FLSHDAT register. The IFB write can only operate on the IFB User-data locations (00-3F) range.

Only one CMD bit can be set at any one time. If more than one CMD bits are set, the Flash Controller will not do anything and return with FAIL result bit.
For Erase and Write command, the Flash Controller will check if the destination address falls within the protection zone defined by CNTPCTL and CNTPCTH registers. If it is protected, the Flash Controller will not execute the command and return with FAIL result bit. For IFB Byte write, the Flash Controller will not execute the command and return with Fail result bit if the byte address falls into manufacturer data range.

FLSHDAT1 (A021h) FLSHDAT2 (A029h) Flash Controller Data Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	**Flash Read Data Register**							
WR	**Flash Write Data Register**							

FLSHADL1 (A023h) FLSHADL2 (A02Bh) Flash Controller Low Address Data Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	**Flash Address Low Byte Register**							
WR	**Flash Address Low Byte Register**							

FLSHADH1 (A022h) FLSHADH2 (A02Ah) Flash Controller High Address Data Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	**Flash Address Low Byte Register**							
WR	**Flash Address Low Byte Register**							

A very common problem of embedded flash memory used as for both data and program storage is the content loss due to improper software or accidents caused by program flow or noise induced random executions of modifying the contents. The implementations of Flash controller take into considerations of these events and provide further protections to avoid accidental reassure or modifications of critical information or software code.

When Flash Controller is issued with a command through FLSHCMD register, it checks whether the destination of the command falls in the content protection zones. If it falls within the protection zones, the flash control aborts its operations and returns with command failure status. Two protections zones are defined by CNTPCTL and CNTPCTH.

CNTPCTL1 (A025h) CNTPCTL2 (A02Dh) Flash Content Protection Low Register R/W 11111111 TB Protected

	7	6	5	4	3	2	1	0
RD	**Content Protection Low Register**							
WR	**Content Protection Low Register**							

This register defines the high bound address from 0000h of the flash to be protected against erasure or modifications. The increment is in 256 Byte. The protected region is defined from 00 to (CNTPCTL1 -1) For example, if CNTPCTL is 49h, then the protection is effective from 0000h to 48FFh. Or if CNTPCTL is A7h, then the protection zone is 0000h to A6FFh. Note CNTPCTL defaults to FFh that protect the whole flash memory. User program needs to write the appropriate data into CNTPCTL to enable erase and write access.

CNTPCTH1 (A026h) CNTPCT2 (A02Eh) Flash Content Protection High Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	**Content Protection High Register**							
WR	**Content Protection High Register**							

This register defines the low bound address from FFFFh of the flash to be protected against erasure or modifications. The increment is in 256 Byte. The protected region is defined from (CNTPCTH1+1) to FF. For example, if CNTPCTH is EAh, then the protection is effective from EB00h to FFFFh. Or if CNTPCT is A7h, then the protection zone is A800 to FFFFh. Note CNTPCTH defaults to 00h and protection of the whole flash is on. User program needs to write the write appropriate data into CNTPCT by reading IFB protect information to protect boot code and increase the protection zone under application considerations.

2.9 System Timers - Timer 0, Timer 1 and Timer 2

The CPU contains three 16-bit timer/counters, Timer 0, Timer 1 and Timer 2. In timer mode, Timer 0, Timer 1 registers are incremented every 12 CPUCLK periods when appropriate timer is enabled. In the timer mode, Timer 2 registers are incremented every 12 or 2 CPUCLK periods (depending on working mode) when it is enabled. In the counter mode, the timer registers are incremented every falling edge on their corresponding inputs, T0, T1, and T2. These inputs are sampled every CPUCLK period.

Timer 0 and Timer 1

Timer 0 and Timer 1 are fully compatible with the standard 8051. Timer 0 and 1 are controlled by TCON (0x88) and TMOD (0x89) registers while each timer consists of two 8-bit registers TH0 (0x8C), TL0(0x8A), TH1 (0x8D), TL1 (0x8B).

TCON (0x88h) Timer 0 and 1 Configuration Register

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

TF1	Timer 1 overflow interrupt flag. TF1 is cleared by hardware when entering ISR.
TR1	Timer 1 run control bit. Set to enable Timer 1, and clear to disable Timer 1.
TF0	Timer 0 overflow interrupt flag. TF0 is cleared by hardware when entering ISR.
TR0	Timer 0 run control bit. Set to enable Timer 0, and clear to disable Timer 0.

IE1, IT1, IE0, IT0 are related to configurations of expanded interrupt INT1 and INT0. These are described in interrupt system section.

TMOD (0x89h) Timer 0 and 1 Mode Control Register

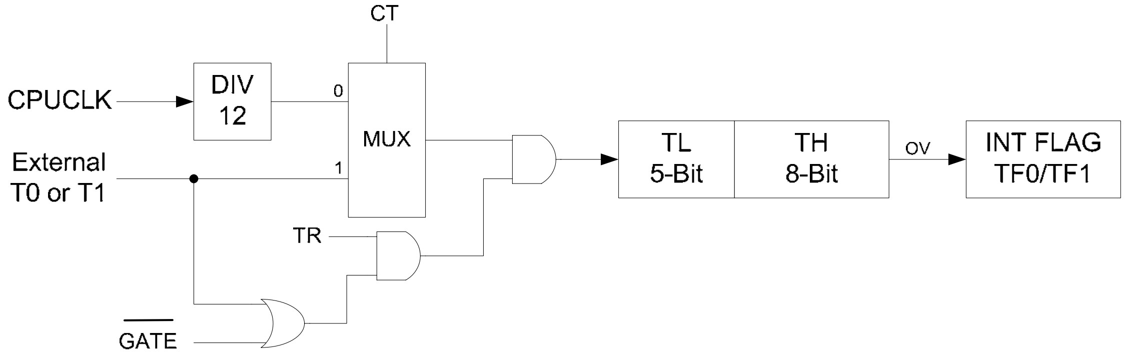
	7	6	5	4	3	2	1	0
RD	**GATE1**	**CT1**	**T1M1**	**T1M0**	**GATE0**	**CT0**	**T0M1**	**T0M0**
WR	**GATE1**	**CT1**	**T1M1**	**T1M0**	**GATE0**	**CT0**	**T0M1**	**T0M0**

GATE1	Timer 1 Gate Control bit. Set to enable external T1 to function as gating control of the counter.
CT1	Counter or Timer mode select bit. Set CT1 to use external T1 as the clock source. Clear CT1 to use internal clock.
T1M1	Timer 1 Mode Select bit.
T1M0	Timer 1 Mode Select bit.
GATE0	Timer 0 Gate Control bit. Set to enable external T1 to function as gating control of the counter.
CT0	Counter or Timer mode select bit. Set CT0 to use external T0 as the clock source. Clear CT0 to use internal clock.
T0M1	Timer 0 Mode Select bit.
T0M0	Timer 0 Mode Select bit.

M1	M0	Mode	Mode Descriptions
0	0	0	TL serves as a 5-bit pre-scaler and TH functions as 8-bit counter/timer. Together they form a 13-bit operations
0	1	1	TH and TL are cascaded to form 16-bit counter/timer
1	0	2	TL functions as 8-bit counter/timer and auto-reload from TH
1	1	3	TL functions as 8-bit counter/timer. TH functions as 8-bit timer which is controlled by GATE1. Only Timer 0 can be configured in Mode 3. When this happens, Timer 1 can only be used where its interrupt is not required.

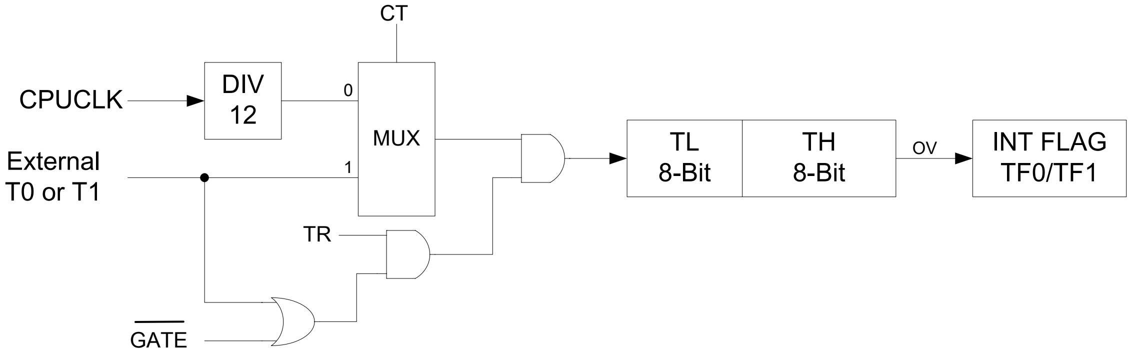
MODE 0

In this mode, TL serves as a 5-bit pre-scaler and TH functions as an 8-bit counter/timer, together it is a 13-bit counter/timer. The Mode 0 operation is shown in the following diagram.



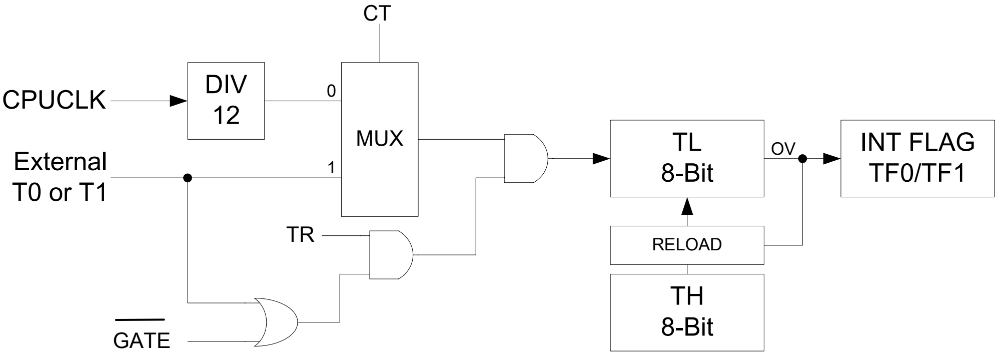
MODE 1

Mode 1 is the same as Mode 0 operation, except TL is configured as 8-bit and thus forming total of 16-bit counter/timer. This is shown as the following diagram.



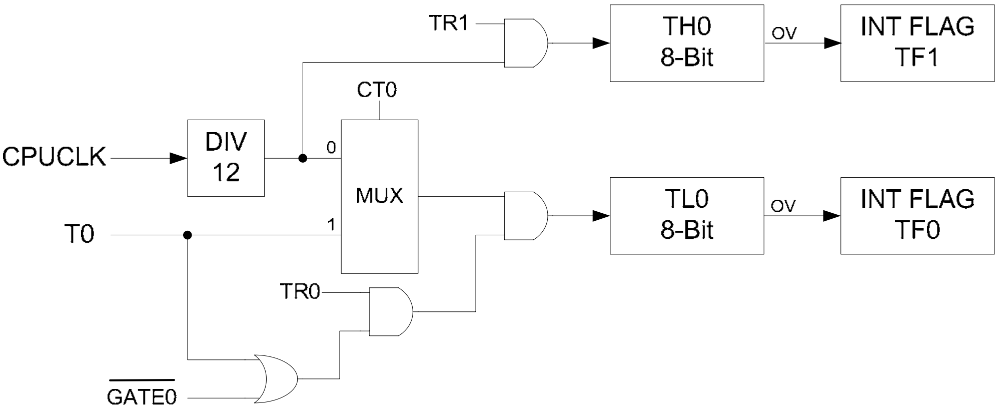
MODE 2

Mode 2 configures the timer as an 8-bit reload-able counter. The counter is TL while TH holds the reload data. The reload occurs when TL overflows. The operation is shown in the following diagram.



MODE 3

Mode 3 is a special mode for Timer 0 only. In this mode, timer 0 is configured as two separate 8-bit counters. TL0 uses Timer 0's control and interrupt flag, and TH0 uses Timer 1's control and interrupt flag. Since Timer 1's control and flag is used, Timer 2 can only be used for counting purpose such as Baud rate generator when Timer 0 is in Mode 3. The operation of Mode 3 is shown in the following diagram.



Timer 2

Timer 2 is fully compatible with the standard 8052 timer 2. Timer 2 can be used for reload-able counter, capture timer, or baud rate generator. Timer 2 uses five SFR for counter registers, capture registers and control register.

T2CON (0xC8h) Timer 2 Control and Configuration Register

	7	6	5	4	3	2	1	0
RD	**TF2**	**EXF2**	**RCLK**	**TCLK**	**EXEN2**	**TR2**	**CT2**	**CPRL2**
WR	**TF2**	**EXF2**	**RCLK**	**TCLK**	**EXEN2**	**TR2**	**CT2**	**CPRL2**

TF2 Timer 2 interrupt flag. TF2 must be cleared by software. TF2 will no be set when RCLK or TCLK is set (that means Timer 2 is used for UART0 Baud rate generator).

EXF2 T2EX falling edge flag. This bit is set when T2EX has a falling edge when EXEN2=1. EXF2 must be cleared by software.

RCLK Receive Clock enable bit
1 – UART0 receiver is clocked by Timer 2 overflow pulses
0 – UART0 receiver is clocked by Timer 1 overflow pulses

TCLK Transmit Clock enable bit
1 – UART0 transmitter is clocked by Timer 2 overflow pulses
0 – UART0 transmitter is clocked by Timer 1 overflow pulses

EXEN2 Enable T2EX function.
1 – Allows capture or reload as a result of T2EX falling edge
0 – Ignore T2EX events

TR2 Start/Stop Timer 2
1 – Start
0 – Stop

CT2 Timer 2 Timer/Counter Mode Select
1 – External event counter uses T2 pin as the clock source
0 – Internal clocked timer mode

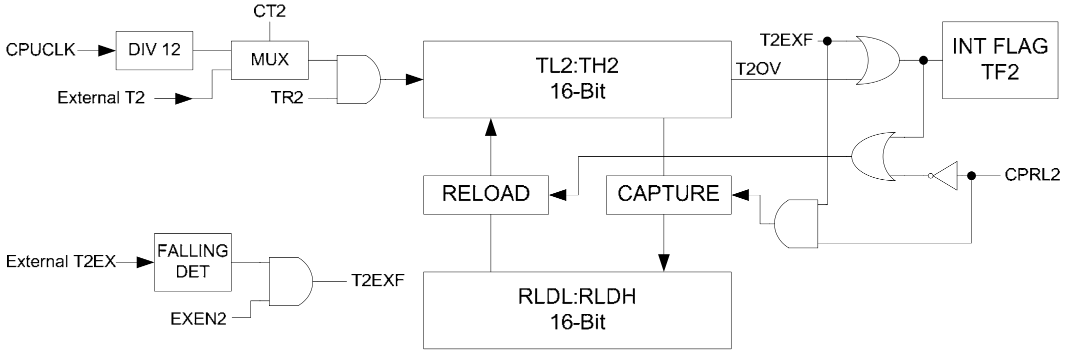
CPRL2 Capture/Reload Select
1 – Use T2EX pin falling edge for capture
0 – Automatic reload on Timer 2 overflow or falling edge of T2EX (when EXEN2=1). If RCLK or TCLK is set (Timer 2 is used as baud rate generator), this bit is ignored and automatic reload is forced on Timer 2 overflow.

Timer 2 can be configured in three modes of operations – auto-reload counter, or capture timer, or baud rate generator. These modes are defined by RCLK, TCLK, CPRL2 and TR2 bits of T2CON registers. The definition is illustrated in the following table.

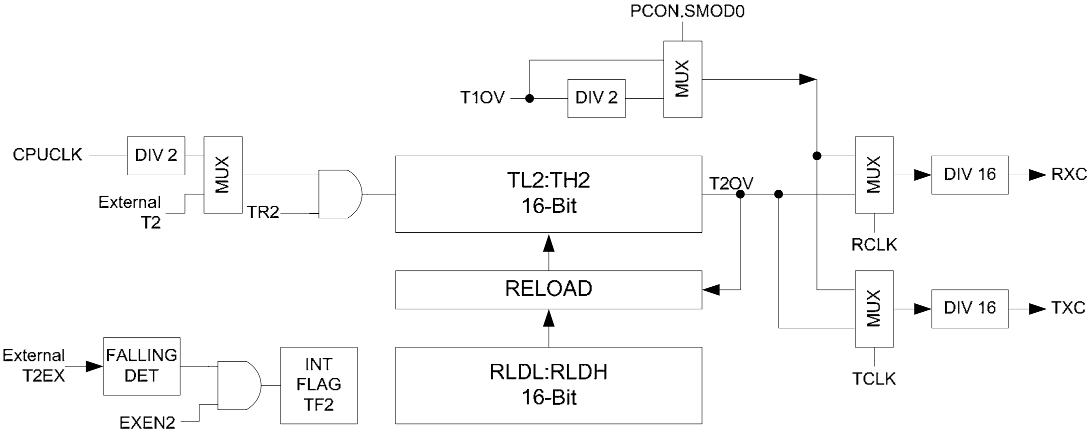
RCLK or TCLK	CPRL2	TR2	Mode Descriptions
0	0	1	16-Bit Auto Reload Counter mode. Timer 2 overflow will set the TF2 interrupt flag and TH2/TL2 is reloaded with RLDH/RLHL register.
0	1	1	16-Bit Capture Timer mode. Timer 2's overflow will set TF2 interrupt flag. When EXEN2=1, TH2/TL2

			content will be captured into RLDH/RLDL when T2EX falling edge occurs.
1	X	1	Baud Rate Generator mode. Timer 2 overflow is used for clocking UART0.
X	X	0	Timer 2 is stopped.

The block diagram of the Timer 2 operating in Auto-Reload Counter and Capture Timer modes are shown in the following diagram.



The block diagram of the Timer 2 operating in Baud Rate Generator is shown in the following diagram.



2.10 Serial Ports - UART0 and UART1

Both UART0 and UART1 are fully compatible with the standard 8052 UART and are full duplex. The receive path of the UART is double-buffered that can commence reception of second byte before previously received byte is read from the receive register. Writing to SBUF loads the transmit register and reading SBUF reads a physically separate receive register. The UART can operate in four modes: one synchronous (Mode 0) and three asynchronous modes (Mode 1, 2, and 3). Mode 2 and Mode 3 have a special provision for multi-processor communications. This feature is enabled by setting SMx2 bit in SCON register. The master processor first sends out an address byte which identifies the slave. An address byte differs from a data byte in the 9th bit is 1 in an address byte and 0 in a data byte. With SMx2 = 1, no slave will be interrupted by a data byte. An address byte will interrupt all slaves. The addressed slave will clear its SMx2 bit and prepare to receive the following incoming data bytes. The slaves that were not addressed leave their SMx2 set and ignore the incoming data.

The UART related registers are SBUF, SCON, PCON, IE, and IP.

SCON0 (0x98h) and SCON1 (0xC0h) UART Configuration Register

	7	6	5	4	3	2	1	0
RD	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TIF**	**RIF**
WR	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TIF**	**RIF**

SM0 and SM1 define the UART operation modes.

MODE	SM0	SM1	Description
0	0	0	Synchronous Shift Register Mode Baud rate = CPUCLK/12
1	0	1	8-Bit UART Mode Baud rate = Timer 0 or Timer 1 overflow rate. This is selected in TCON registers.
2	1	0	9-Bit UART Mode, fix baud rate Baud rate = CPUCLK/64 (PCON.SMOD0 = 0) or CPUCLK/32 (PCON.SMOD = 1)
3	1	1	9-Bit UART Mode, variable baud rate Baud rate = Timer 0 or Timer 1 overflow rate. This is selected in TCON registers.

SM2 Set to enable a multiprocessor communication as a slave device.

REN Set to enable the serial reception. Clear to disable reception

TB8 The 9th bit transmit value in 9-bit UART Mode (mode 2 and mode 3). Set or cleared by CPU depending on the function of the 9th bit whether a parity check bit or a multi-processor token.

RB8 The 9th bit receive value in 9-bit UART Mode (mode 2 and mode 3). Set or cleared by hardware.
In mode 1 of UART1, if SM2 is 0, RB8 is the stop bit, and if SM2 is 1, RB8 is the multi-processor token bit.

TIF Transmit Interrupt Flag. Set by hardware after completion of a serial transmission and must be cleared by software. The interrupt enable bit is located in IE (0xA8) and the interrupt priority is located in IP (0xB8).

RIF Receive Interrupt Flag. Set by hardware after completion of a serial reception and must be cleared by software. The interrupt enable bit is located in IE (0xA8) and the interrupt priority is located in IP (0xB8).

SBUF0 (0x99h) and SBUF1 (0xC0h) UART Configuration Register

	7	6	5	4	3	2	1	0
RD	**RB7**	**RB6**	**RB5**	**RB4**	**RB3**	**RB2**	**RB1**	**RB0**
WR	**TB7**	**TB6**	**TB5**	**TB4**	**TB3**	**TB2**	**TB1**	**TB0**

SBUF is used for both transmit and receive. Writing a data byte into SBUF put this data in UART's transmit buffer and starts a transmission. Reading a byte from SBUF reads data from the UART's receive buffer.

Mode 0

Mode 0 is a simple synchronous shift register mode. TXD is an output that outputs the shift clock which is fixed at CPUCLK/12. RXD is a bidirectional I/O that serves as the shift data. To utilize this mode, TXD pin must be enabled as an output pin, while RXD needs to be configured as an open-drain type I/O.

The shift data changes at the rising edge of the shift clock and is valid at the falling edge of the shift clock. The transmit starts when the SBUF is written with a new byte and TI is cleared to 0. When the byte is transmitted, TI is set and the UART wait for next byte to be transmitted. The reception is initiated by setting REN=1 and RI cleared to 0. When a byte is received, RI is set by UART.

Mode 1

Mode 1 is 8-bit UART mode. RXD is the serial input and TXD is the serial output. To utilize this mode, the corresponding RXD and TXD pin configuration should also be set correctly. 10-bit data (including a Start bit, 8 data bit, and a stop bit) are transferred.

For UART0, the baud rate is set by Timer 1 or Timer 2 overflow rate. The control is determined by SMOD0.PCON, and RCLK.TCON2, TCLK.TCON2. When SMOD0.PCON is 1, Timer 1 overflow is selected, and SMOD0.PCON is 0, Timer 1 overflow rate divided by 2 is selected. And if RCLK.TCON2, or TCLK.TCON2 is set, the Timer 2 overflow rate is selected and will overwrite the SMOD0 setting.

For UART1, only Timer 1 overflow rate can be used. The rate can be controlled by SMOD1.PCON. When SMOD1.PCON is set, the Timer 1 overflow is used as the baud rate, when SMOD1.PCON is 0, Timer 1 overflow divided by 2 is used as the baud rate.

Mode 2

Mode 2 is 9-bit UART mode. RXD is the serial input and TXD is the serial output. To utilize this mode, the corresponding RXD and TXD pin configuration should also be set correctly. 11-bit data including a Start bit (always 0), 8 data bit, a programmable 9th bit, and a stop bit (always 1) are transferred. The 9th bit can be used to act as the parity bit and be set by the software through TB8 in SCON. The received 9th bit can be read from TB8. The software must calculate the correctness of the parity check.

The baud rate in Mode 2 is fixed at 1/32 or 1/64 of CPU clock. This is controlled by SMOD0 or SMOD1 in PCON register.

Mode 3

Mode 3 is similar to Mode 2 and is 9-bit UART mode. RXD is the serial input and TXD is the serial output. To utilize this mode, the corresponding RXD and TXD pin configuration should also be set correctly. 11-bit data including a Start bit (always 0), 8 data bit, a programmable 9th bit, and a stop bit (always 1) are transferred. The 9th bit can be used to act as the parity bit and be set by the software through TB8 in SCON. The received 9th bit can be read from TB8. The software must calculate the correctness of the parity check.

The baud rate in Mode 3 is controlled similar to Mode 1 that is derived from Timer 1 or Timer 2 overflow, and is set by SMOD, and TCON

2.11 I2C Master Controller

The I2C master controller provides the interface to I2C slave devices. It can be programmed to operate with arbitration and clock synchronization to allow it to operate in multi-master configurations. The master uses SCL and SDA pins. The controller contains a built-in 8-bit timer to allow various I2C bus speed.

I2CMTP (0xF7h) I2C Master Time Period R/W 00000000

	7	6	5	4	3	2	1	0
RD	P.7	P.6	P.5	P.4	P.3	P.2	P.1	P.0
WR	P.7	P.6	P.5	P.4	P.3	P.2	P.1	P.0

This register set the period time of I2C bus clock – SCL. The SCL period time is set according to
SCLPERIOD = 8 * (1 + I2CMTP) * CPUCLK_PERIORD

I2CMSA (0xF4) I2C Master Slave Address R/W 00000000

	7	6	5	4	3	2	1	0
RD	SA6	SA5	SA4	SA3	SA2	SA1	SA0	RS
WR	SA6	SA5	SA4	SA3	SA2	SA1	SA0	RS

SA6 – SA0 defines the slave address the I2C master uses to communicate.
RS bit determines if the next operation will be a RECEIVE (1) or SEND (0).

I2CMBUF (0xF6) I2C Master Data Buffer Register R/W 00000000

	7	6	5	4	3	2	1	0
RD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
WR	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0

I2CMBUF functions as a transmit data register when written and as a receive data register when read. When written, TDn will be sent on the bus by the next SEND or BURST SEND operations. TD7 is sent first. When read, RDn contains the 8-bit data that has been received the bus due to the last RECEIVE or BURST RECEIVE operations.

I2CMCR (0xF5) I2C Master Control and Status Register R/W 00000000

	7	6	5	4	3	2	1	0
RD	-	BUSBUSY	IDLE	ARBLOST	DATAACK	ADDRACK	ERROR	BUSY
WR	INFILEN	-	-	HS	ACK	STOP	START	RUN

The I2CMCR register is used for setting control when it is written, and serve as the status when read.

INFILEN is input pulse filtering control. When IFILEN is set, pulses shorter than 50 nsec on inputs of SDA and SCL are filtered out.

The START bit is used to generate START, or REPEAT START protocol. The STOP bit determines if the cycle will stop at the end of the data cycle or continue on to a burst. To generate a single read cycle, SA is written with the desired address, RS is set to 1, and I2CMCR is written with ACK=0, STOP=1, START=1, RUN=1 to perform the operation and then STOP. When the operation is completed (or aborted due to errors), I2C master generates an interrupt. The ACK bit must be set normally 1. This causes the controller to send an ACK automatically after each byte transaction. The ACK bit must be reset when set to 0 when the master is operated in receive mode and intend not to receive further data from the slave device.

The following table lists the permitted control bits combinations in master idle mode.

HS	RS	ACK	STOP	START	RUN	OPERATIONS
0	0	-	0	1	1	START condition followed by SEND. Master remains in TRANSMITTER mode
0	0	-	1	1	1	START condition followed by SEND and STOP
0	1	0	0	1	1	START condition followed by RECEICE operation with negative ACK. Master remains in RECEICER mode
0	1	0	1	1	1	START condition followed by RECEIVE and STOP

0	1	1	0	1	1	START condition followed by RECEIVE. Master remains in RECEIVER mode
0	1	1	1	1	1	Illegal command
1	0	0	0	0	1	Master Code sending and switching to HS mode

The following table lists the permitted control bits combinations in master TRANSMITTER mode.

HS	RS	ACK	STOP	START	RUN	OPERATIONS
0	-	-	0	0	1	SEND operation. Mater remains in TRANSMITTER mode
0	-	-	1	0	0	STOP condition
0	-	-	1	0	1	SEND followed by STOP condition
0	0	-	0	1	1	Repeated START condition followed by SEND. Master remains in TRANSMITTER mode
0	1	-	1	1	1	Repeated START condition followed by SEND and STOP condition
0	1	0	0	1	1	Repeated START condition followed by RECEIVE operation with negative ACK. Master remains in TRANSMITTER mode
0	1	0	1	1	1	Repeated START condition followed by SEND and STOP condition.
0	1	1	0	1	1	Repeated START condition followed by RECEIVE. Master remains in RECEIVER mode.
0	1	1	1	1	1	Illegal command

The following table lists the permitted control bits combinations in master RECEIVER mode.

HS	RS	ACK	STOP	START	RUN	OPERATIONS
0	-	0	0	0	1	RECEIVE operation with negative ACK. Mater remains in RECEIVE mode
0	-	-	1	0	0	STOP condition
0	-	0	1	0	1	RECEIVE followed by STOP condition
0	-	1	0	0	1	RECEIVE operation. Mater remains in RECEIVER mode
0	-	1	1	0	1	Illegal command
0	1	0	0	1	1	Repeated START condition followed by RECEIVE operation with negative ACK. Master remains in RECEIVER mode
0	1	0	1	1	1	Repeated START condition followed by RECEIVE and STOP conditions
0	1	0	1	1	1	Repeated START condition followed by RECEIVE. Master remains in RECEIVER mode
0	0	-	0	1	1	Repeated START condition followed by SEND. Master remains in TRANSMITTER mode.
0	0	-	1	1	1	Repeated START condition followed by SEND and STOP conditions.

All other control bits combinations not mentioned in the above three tables are NOP. In Master RECEIVER mode, STOP condition should be generated only after data negative ACK executed by Master or address negative ACK executed by slave. Negative ACK here means SDA is pulled low during the acknowledge clock pulse.

I2CMCR when read reflect the status of the I2C master controller.
IDLE this bit indicates that I2C master is in the IDLE mode.
BUSY this bit indicates that I2C master is receiving or transmitting data, and other status bits

are not valid.

BUSBUSY this bit indicates that the external I2C bus is busy and access to the bus is not possible. This bit is set/reset by START and STOP conditions.

ERROR this bit indicates that error occurred in the last operations. The errors include slave address was not acknowledged, or transmitted data was not acknowledged, or the master controller lost arbitration.

ADDRERR this bit indicates that the last operation slave address transmitted was not acknowledged.

DATAERR this bit indicates that the last operation transmitted data was not acknowledged.

ARBLOST this bit indicates that the last operation I2C master controller lost the bus arbitration.

2.12 Multiplication and Division Unit (MDU)

MDU proves accelerations on unsigned integer operations of 16-bit multiplications 32-bit division, and shift and normalize operations. The following table shows the execution characteristics of these operations. The MDU do not contains the operation completion status flag. Therefore the most efficient utilization of the MDU is just use NOP delay for the required clock time of the MDU operation types. The number of the clock time required is counted from the last write of the write sequence.

Operations	Result	Reminder	Clock Time
32 bit division by 16 bit	32 bit	16 bit	17
16 bit division by 16 bit	16 bit	16 bit	9
16 bit multiplication by 16 bit	32 bit	-	10
32 bit normalization	-	-	3 – 20
32 bit shift left/right	-	-	3 – 18

The MDU is accessed through MD0 to MD5 that contains the operands and the results, and ARCON register.

ARCON (0xFF) MDU Control R/W 00000000

	7	6	5	4	3	2	1	0
RD	**MDEF**	**MDOV**	**SLR**	**SC4**	**SC3**	**SC2**	**SC1**	**SC0**
WR	**MDEF**	**MDOV**	**SLR**	**SC4**	**SC3**	**SC2**	**SC1**	**SC0**

MDEF MDU Error Flag Bit. Set by hardware to indicate improper operations that MDx is written before the previous operation has completed. MDEF is automatically cleared after read of ARCON.

MDOV MDU Overflow Flag Bit. MDOV is set by hardware if division by zero, or multiplications result is greater than 0x0000FFFFh

SLR Shift Direction Control Bit. SLR = 1 for shift right and SLR =0 for shift left.

SC4-0 Shift Count Control and Result Bit. If SC0-4 is written with 00000, the normalization operation will be performed by MDU. When the normalization is completed, SC4-0 contains the number of shift performed in the normalization. If SC4-0 is written with non-zero value, then shift operation will be performed by MDU with the number of shift specified by SC4-0 value.

MD0 – MD5 (0xF9 to 0xFE) MDU Data Register R/W 00000000

	7	6	5	4	3	2	1	0
RD	**MD0 to MD5 Data register**							
WR	**MD0 to MD5 Data register**							

MDU operation is consisted of three phases.

1. Loading MDx data registers in an appropriate order depending on the operation.
2. Execution of the operation.
3. Reading result from MDx registers.

The following table shows the MDU read and write sequence. Each operation has its unique writing and read sequence of MDx registers therefore the precise access sequence is required.

Division – 32bit divide by 16bit or 16bit divide by 16bit

Follow the following write sequence. The first write of MD0 will reset the MDU and start the MDU error flag mechanism. The last write starts MDU with the calculation.

1. Write MD0 with Dividend LSB byte
2. Write MD1 with Dividend LSB+1 byte
3. Write MD2 with Dividend LSB+2 byte (ignore this step for 16-bit divide by 16-bit)
4. Write MD3 with Dividend MSB byte (ignore this step for 16-bit divide by 16-bit)
5. Write MD4 with Divisor LSB byte
6. Write MD5 with Divisor MSB byte

Then follow the following read sequence. The last read release the MDU for next operations.

1. Read MD0 with Quotient LSB byte
2. Read MD1 with Quotient LSB+1 byte
3. Read MD2 with Quotient LSB+2 byte (ignore this step for 16-bit divide by 16-bit)

4. Read MD3 with Quotient MSB byte (ignore this step for 16-bit divide by 16-bit)
5. Read MD4 with Reminder LSB byte
6. Read MD5 with Reminder MSB byte
7. Read ARCON to determine error or overflow condition

Please note if the sequence is violated, the calculation can be interrupted and the result may contain errors.

Multiplication – 16bit multiply by 16-bit

Follow the following write sequence.

1. Write MD0 with Multiplicand LSB byte
2. Write MD4 with Multiplicator LSB byte
3. Write MD1 with Multiplicand MSB byte
4. Write MD5 with Multiplicator MSB byte

Then follow the following read sequence.

1. Read MD0 with Product LSB byte
2. Read MD1 with Product LSB+1 byte
3. Read MD2 with Product LSB+2 byte
4. Read MD3 with Product MSB byte
5. Read ARCON to determine error or overflow condition

Normalization – 32bit

Normalization is done on integer variable stored in MD0 to MD3. After normalization, all leading zeroes are removed by shift left operations. To start the normalization operation, SC4-0 in ARCON is first written with 00000. After completion of the normalization, SC4-0 is updated with the number of leading zeroes and the normalized result is stored back to MD0 to MD3. The number of the shift of the normalization can be used as exponent.

The following write sequence should be followed. The last write to ARCON start the normalization operations by MDU.

1. Write MD0 with Operand LSB byte
2. Write MD1 with Operand LSB+1 byte
3. Write MD2 with Operand LSB+2 byte
4. Write MD3 with Operand MSB byte
5. Write ARCON with SC4-0 = 00000

Then follow the following read sequence.

1. Read MD0 with Result LSB byte
2. Read MD1 with Result LSB+1 byte
3. Read MD2 with Result LSB+2 byte
4. Read MD3 with Result MSB byte
5. Read ARCON's SC4-0 for normalization count or error flag

Shift – 32bit

Shift is done on integer variable stored in MD0 to MD3. To start the shift operation, SC4-0 in ARCON is first written with shift count and SLR with shift direction. After completion of the Shift, the result is stored back to MD0 to MD3.

The following write sequence should be followed. The last write to ARCON start the normalization operations by MDU.

1. Write MD0 with Operand LSB byte
2. Write MD1 with Operand LSB+1 byte
3. Write MD2 with Operand LSB+2 byte
4. Write MD3 with Operand MSB byte
5. Write ARCON with SC4-0 = Shift count and SLR with shift direction

Then follow the following read sequence.

1. Read MD0 with Result LSB byte
2. Read MD1 with Result LSB+1 byte
3. Read MD2 with Result LSB+2 byte
4. Read MD3 with Result MSB byte
5. Read ARCON's for error flag

MDU Flag

The error flag (MDEF) of MDU indicate improperly performed operations. The error mechanism is started at the first MD0 write and finished with the last read of MD result register. MDEF is set if current operation is interrupted or restarted by improperly write of MD register before the operation is completed. MDEF is cleared if the operation completes successfully and proper write/read sequence is performed.

The overflow flag (MDOV) of MDU indicate an error of operations. MDOV is set if

1. Divide by zero
2. Multiplication overflow
3. Normalization operation has been performed on already normalized variables (MD3.7 =1)

3.0 On-Chip External Peripherals

This section describes the on-chip peripherals on CS8953. These peripherals are termed external because they are external to the CPU. These peripheral will continue functioning if enabled and SYSCLK is available. And they can also be configured to wake up the CPU through interrupt.

The peripherals included can be classified into mandatory and optional. The mandatory ones are those present for all Myson 8-bit MCU products. And the optional ones will depends on the individual device configurations for different application requirement.

The mandatory ones include On-chip Regulator, Internal Oscillator, Low Voltage Detection. These are mandatory because they play an integral part of the operation with CPU and other built-in function of the devices.

The optional peripherals are those more decoupled from CPU and system operations and thus are there for particular application focused requirement. The optional analog peripherals include Crystal Oscillator, Phase Lock Loop Clock Generator, Real Time Clock, Analog to Digital Converters, Digital to Analog Converters, Analog Comparators, Operational Amplifiers, Analog Multiplexers. The optional digital peripherals include Programmable Counter Array, SPI Interface Controller, I2C Slave Controller, Enhanced UART Controller, and others.

The following subsections describe the details of these on-chip peripherals.

3.1 On-Chip 2.5V Regulator

The core logic and embedded Flash is supplied by the 2.5V output from the on-chip regulator that regulates VDD (3.0V to 5.5V). There is variation of this 2.5V due to chip to chip difference. Because this 2.5V is also used to generate the reference for IOSC and other analog peripherals such as ADC, the relative accuracy of this supply voltage is important.

The on-chip 2.5V regulator thus has trimming options to fine tune the output level. The trimming is done by an XFR register REGTRM (A00h).

REGTRM (A000h) Regulator Trim Register WO 11111111 TB Protected

	7	6	5	4	3	2	1	0
RD	**REGTRM Register**							
WR	**REGTRM Register**							

The content of this register trims the 2.5V regulator output level. The default after reset is set to its maximum. User program should read calibrated data from IFB and write into REGTRM to get accurate 2.5V regulator output.

After reset, REGTRM is loaded with default 0xFF to set the regulator to its maximum. A calibrated value that produces accurate 2.5V +/- 0.1V is stored in IFB. User program should access this information and set the calibrated value into REGTRM. This should be done prior to trimming of IOSC or other analog peripherals such as LVD circuits. The following table lists the REGTRIM value versus the REGOUT value in terms of VBG ratio.

REGTRM	R_EQ	REGOUT/VBG
1111 1111	0	2.286
0111 1111	3.2K	2.200
1111 0111	4.8K	2.161
0011 1111	6.4K	2.125
0111 0111	8.0K	2.091
0001 1111	9.6K	2.059
0011 0111	11.2K	2.029
0000 1111	12.8K	2.000
1111 0001	14.4K	1.973
0011 0011	16.0K	1.947
0111 0001	17.6K	1.923
0001 0011	19.2K	1.900
0011 0001	20.8K	1.878
0000 0011	22.4K	1.857
0001 0001	24.0K	1.837
0011 0000	25.6K	1.818
0000 0001	27.2K	1.800
0001 0000	28.8K	1.783
0000 0000	32.0K	1.750

A fast way of obtaining the REGTRM value for 2.5V will be as following:

1. Set REGTRM = 11111111 (this is the default after reset).
2. Measure VDD25 = M.
3. Obtain N = M / 2.286.
4. Get the setting X which is closest to M. Also get Y and Z which are above and below X.
5. Set REGTRM = X, and measure VDD25.
6. Set REGTRM = Y, and measure VDD25.
7. Set REGTRM = Z, and measure VDD25.
8. Pick one of X, Y, and Z as the optimal setting.

Note the measurement need to wait for a certain time after setting to allow the regulator to stabilize. The time required depends on the decoupling capacitor on VDD25. Typical delay should be around 10msec.

3.2 Internal Oscillator

The internal oscillator is a very important peripheral as it provides the default clock source after reset and other critical timing. The internal oscillator has the salient features that it behaves well during the enable and disable transient. No clock glitches or extra clock edge is generated during the on/off transition, and the oscillator can reach to stable oscillations within very short time typically within 10 cycles. The IOSC consumes around 350uA when enabled. The IOSC is always enabled except entering into STOP mode. And in STOP mode when it is disabled, IOSC only consumes less than 1uA standby current.

Similar to the on-chip regulator, IOSC will also suffer chip to chip variations. A calibrated value that set IOSC at 3.68MHz +/- 2% is also stored in IFB. Thus the user program can set this value to IOSC trimming register, IOSCITRM (A001h) and IOSCVTRM (A002h). The IOSC frequency also vary chip to chip, but it has very little variations over the operation range (0 – 70C, -40 – 85C for I-Grade, VDD = 3V – 5.5V). The variation is typically less than +/-2% over the operation conditions. Thus it is possible that user program to set a different frequency other than 3.68MHz as long as user program provide a calibration method to set IOSC frequency at the desired value at typical operation condition, and it will be stable and accurate over the entire operation range.

The trimming registers of IOSC are described as following. The typical setting value versus resulting frequency is also shown as a reference. ITRIM register provides a coarse tuning range and VTRM register provides fine tuning.

IOSCITRM (A001h) IOSC Configuration Register WO 0000001 TB Protected

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	**ITRM1**	**ITRM0**
WR	-	-	-	-	-	-	**ITRM1**	**ITRM0**

ITRM1, ITRM0 Trim the internal current source of the IOSC. User program should obtain the calibrated value of these two bits from IFB and write into these two bits.

IOSCVTRM (A002h) IOSC Trim Register WO 10001110 TB Protected

	7	6	5	4	3	2	1	0
RD	**VTRM7**	**VTRM6**	**VTRM5**	**VTRM4**	**VTRM3**	**VTRM2**	**VTRM1**	**VTRM0**
WR	**VTRM7**	**VTRM6**	**VTRM5**	**VTRM4**	**VTRM3**	**VTRM2**	**VTRM1**	**VTRM0**

VTRM7 – VTM0 These bits fine trim the detection threshold of IOSC for allowing fine tuning the IOSC clock frequency. User program should obtain the calibrated value of these two bits from IFB and write into these bits.

The manufacturer supplied trim value is stored in IFB and is trimmed to 3.68MHz. The user program does have the freedom to set the IOSC at another preferred frequency but user should provide a calibration method to set the desired frequency. Once set, the IOSC frequency has accuracy within +/- 2% over the operation conditions. The following lists the range of the typical IOSC frequency for each trimming setting.

ITRM[1-0] = 00, VTRM[7-0], IOSC Frequency range = 2.00MHz (FF) – 3.45MHz (00)

ITRM[1-0] = 01, VTRM[7-0], IOSC Frequency range = 2.95MHz (FF) – 5.15MHz (00)

ITRM[1-0] = 10, VTRM[7-0], IOSC Frequency range = 3.85MHz (FF) – 6.80MHz (00)

ITRM[1-0] = 11, VTRM[7-0], IOSC Frequency range = 4.85MHz (FF) – 8.40MHz (00)

The trimming of the IOSC should use the following procedure to obtain the default setting for 3.68MHz.

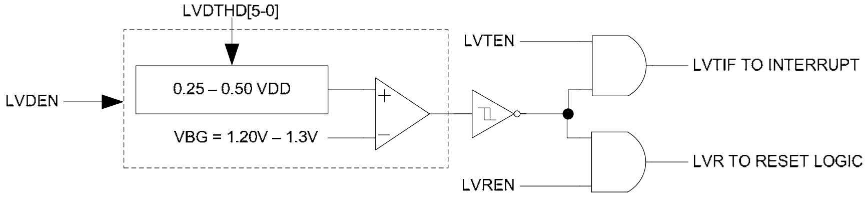
1. Set ITRM = 01, and
2. Set VTRM = 00, measure frequency
2. Set VTRM = FF, measure frequency
3. Set VTRM = 7F, measure frequency
4. Use binary search to obtain the closest setting for 3.68MHz

Note the frequency versus VTRM setting is monotonic. When VTRM = 00, the frequency is highest, and when VTRM = FF, the frequency is lowest.

3.3 LOW VOLTAGE DETECTION

The low voltage detect circuit detect the condition of VDD supply. When enabled, it detect VDD < VTH condition, and can be configured to generate an interrupt or a system reset. Because the existence of on-chip regulator the system and user program should have ample time to responds to the interrupt. The larger the decoupling capacitance on VDD25, the longer this time can be extended. To ensure a reliable MCU operation and prevent loss of data, user program should terminate all critical process and wait for power supply to get back to normal or as power supply continue to deteriorate and finally reset the system.

The block diagram of Low Voltage Detection Circuit is shown in the following figure.



When enabled LVD circuit consumes about 100uA to 400uA depending on VDD level. The LVDTHD[5-0] set the compare threshold from 0.25VDD to 0.50VDD against the internal reference voltage of 1.25V in uniform steps. Therefore the detection range is from 2.5V to 5.0V assuming the reference voltage is 1.25V.

The default state of the LVD circuit is disabled. The user program needs to enable the preferred configuration and set the appropriate detection threshold level. When enabled, the LVD circuit needs about 10usec to get initialized. The following XFR registers are used for this purpose.

LVDCFG (A010h) Supply Low Voltage Detection Configuration Register WO 10000000 TB Protected

	7	6	5	4	3	2	1	0
RD	**LVDEN**	**LVREN**	**LVTEN**	**-**	**-**	**-**	**-**	**LVTIF**
WR	**LVDEN**	**LVREN**	**LVTEN**	**-**	**-**	**-**	**-**	**LVTIF**

LVDEN Set to turn on supply voltage detection circuits
LVREN LVREN = 1 will allow low voltage detect condition to cause a system reset.
LVTEN LVTEN = 1 will allow low voltage detect condition to generate an interrupt.
LVTIF Low voltage detection interrupt flag. Set when LVD detection occurs and must be cleared by software.

LVDTHD (A011h) Supply Low Voltage Detection Threshold Register WO XX111111 TB Protected

	7	6	5	4	3	2	1	0
RD	**-**	**-**	**LVDTHD5**	**LVDTHD4**	**LVDTHD3**	**LVDTHD2**	**LVDTHD1**	**LVDTHD0**
WR	**-**	**-**	**LVDTHD5**	**LVDTHD4**	**LVDTHD3**	**LVDTHD2**	**LVDTHD1**	**LVDTHD0**

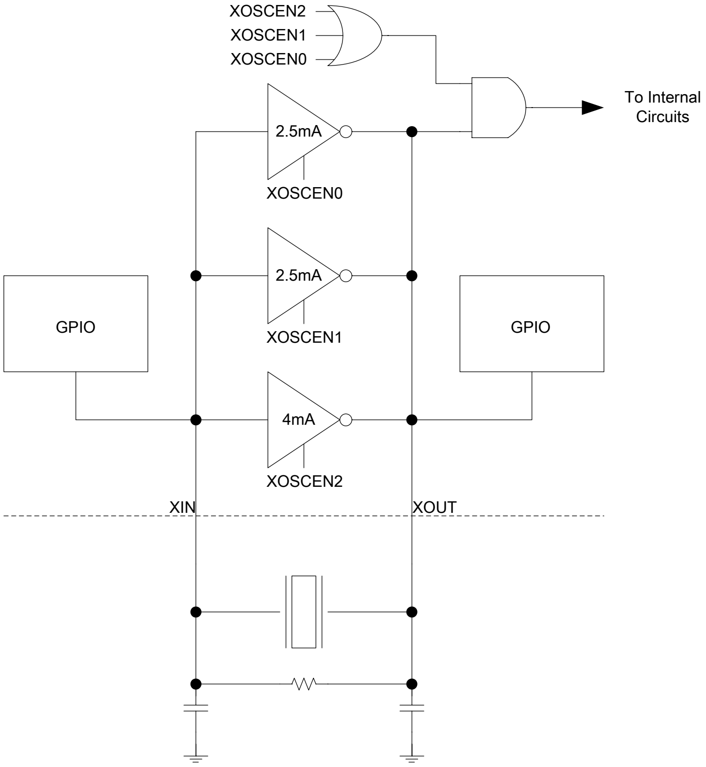
LVDTHD = 3F will set the detection threshold at its minimum (approximately 2.5V), and LVDTHD = 00 will set the detection threshold at its maximum (approximately 5.0V).

Because there is chip to chip variation of the internal reference voltage, the manufacturer provides two LVDTHD values in the IFB, one for 4.5V and one for 3.0V. Then user can use these two values to set LVDTHD for desired detection threshold. For example, the value for 4.5V and 3.0V detection threshold is 0x06 (0d06), and 0x33(0d51), and the desired detection threshold is 3.7V. Then the optimal setting of LVDTHD can be obtained first by its offset from 4.5V setting which is (51 – 6)/(4.5 – 3.0) * (4.5 – 3.7) = 24. Then the setting for detection of 3.7V is just 0x06 + 0x18 (0d24) = 0x1D.

3.4 Crystal Oscillator (XOSC)

Crystal oscillator (XOSC) provides a very accurate clock source for the system. The default for XOSC is in disabled state after power on or reset. Please note when enabling XOSC from disabled state to enabled state, it usually takes 10s of mili-second for XOSC to stabilize. The software needs to take this into considerations before switching into XOSC clock. Typically, the crystal pins are shared with GPIO pins. The software also should take care that the shared GPIO function is properly turned off not affecting the crystal oscillation.

The XOSC circuit is described in the following block diagram.



An external feedback resistor typically ranging from 1M Ohm to 4M Ohm is required. There are three oscillator circuits in parallel and separately controlled. The capability of a stable oscillation depends on several factors, the operating supply voltage, the frequency of crystal, the intended operating temperature range, the quality of the crystal, and the external capacitance load. In general, more driving capability of the oscillation inverter, the more reliable is the oscillation, and this is at the expense of higher power consumption.

XOSCCFG (A007h) XOSC Configuration Register WO 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	XOSCEN2	XOSCEN1	XOSCEN0
WR	-	-	-	-	-	XOSCEN2	XOSCEN1	XOSCEN0

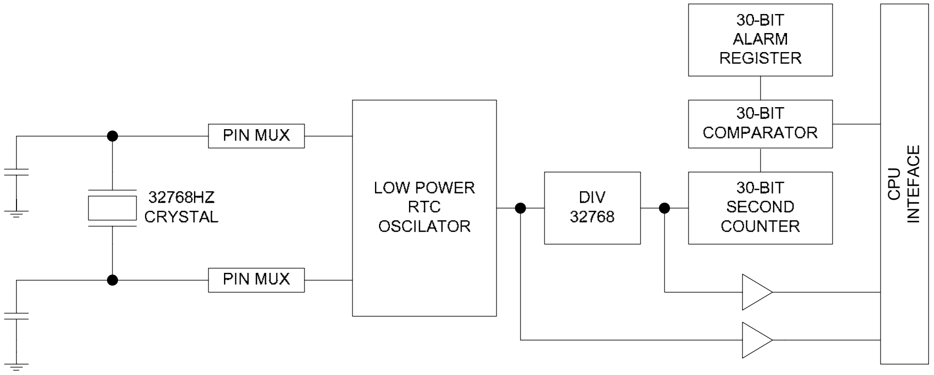
This register configures the operation of the crystal oscillator. The following setting is recommended but user can determine the optimal setting if the application environment is set.

XOSCEN2	XOSCEN1	XOSCEN0	OSC Operations
0	0	0	Powered Down
0	0	1	Low Power for 3V up to 8MHz, 5V up to 12MHz
0	1	1	Medium Power for 3V up to 16MHz, 5V up to 24MHz
1	1	1	High Power for 3V up to 25MHz, 5V up to 30MHz

If an external clock source is applied, it should be applied at XOUT pin. And only XOSCEN0 needs to be set to 1 to allow the external clock passing into the internal circuits.

3.5 Real Time Clock (RTC)

The on-chip RTC is a simplified real-time clock. It contains an ultra low power 32K clock oscillator typically consuming less than 1.2uA, and a 30-Bit SECOND counter, along with a 30-Bit ALARM register, and a 30-Bit comparator that generates RTC interrupt when the counter matches with the ALARM. In addition, it output a 32K clock (RTCCLK) as one of the selections for system clock sources. The block diagram of the RTC is shown in the following.



Because RTC consumes very little power, it is never powered down. The connections of the RTCXIN and RTCXOUT to the oscillator goes through pin multiplexer's analog port, therefore the appropriate set up is necessary to enable the RTC oscillator through I/O configuration.

The SECOND counter starts with a random number when RTC is powered up. The user software needs to record this initial value as the reference start time/date/month/year. The software can examine the new counter value and then needs to calculate the difference between the new value and the initial value to determine the elapsed time. The current time/date/month/year can then be derived from the elapse time in seconds. For 30-bit value, the maximum accountable elapse time exceeds 34 years.

RTCCMD (A00Eh) RTC Configuration and Command Register RW 0000000

	7	6	5	4	3	2	1	0
RD	SECINTEN	ALMINTEN	SECINT	ALMINT	4HZINTEN	4HZINT	ADD1	ADD0
WR	SECINTEN	ALMINTEN	SECINT	ALMINT	4HZINTEN	4HZINT	ADD1	ADD0

SECINTEN RTC Second Interrupt Enable Bit, Set to enable RTC second interrupt. When SECINTEN = 1, RTC generates an interrupt every second.

ALMINTEN Alarm Interrupt Enable Bit. Set to enable alarm interrupt.

SECINT Set when Second Interrupt occurs. Cleared by software

ALMINT Set when Alarm interrupt occurs. Cleared by software

4HZINTEN 4Hz Interrupt Enable Bit, Set to enable 4Hz interrupt.

4HZINT Set by hardware every 250 millisecond if enabled. Cleared by software

ADD1, ADD0 Address pointers for the second and alarm registers

ADD1	ADD0	OPERATION
0	0	Read SECOND REG 0, Write ALRAM REG 0
0	1	Read SECOND REG 1, Write ALARM REG 1
1	0	Read SECOND REG 2, Write ALARM REG 2
1	1	Read SECOND REG 3, Write ALARM REG 2

RTCDATA (A00Fh) RTC DATA Register 0 RO XXXXXXXX

	7	6	5	4	3	2	1	0
RD	RTC SECOND REGISTER							
WR	RTC ALRM REGISTER							

This is the register for transferring data between CPU and RTC. There are 4-bytes for SECOND and ALARM registers. The ADD0 and ADD1 address which byte of the registers is accessed.

3.6 10-Bit Analog to Digital Converter (ADC)

The on-chip ADC is a 10-bit SAR based ADC. The ADC has inherent monotonic characteristics. The ADC clock is programmable and set by the ADC clock scaler. The maximum ADC clock is 4MHz. And the minimum conversion time is 3.5usec. When enabled, the ADC consumes about 3mA of current.

The ADC circuit is implemented in 2.5V. As the result, the full-scale input range of the ADC is 0V – 2.5V. To accommodate larger input range, the input needs to be attenuated. There is a 0.5 attenuator built-in in the front-end sample and hold amplifier option. Setting ATTN bit in ADCCFG will turn on the attenuator, and allow the input range to expand by 2X.

The ADC has inherent 4 channels and has two operation modes. The single conversion mode converts the selected channel and after conversion it put the data in the result register of the corresponding channel. Each single conversion takes 14 clock cycles to complete. For continuous conversion mode, the ADC will cycle the selected channels and put the conversion results in the corresponding result register. Each conversion in continuous mode only takes 13 clock cycles by overlapping one clock cycle for sample and hold. For both operation modes, interrupt is generated when conversion is completed for each channel. In CS8953, each inherent channel is further multiplexed to two external pins. For Channel D, four external pins are multiplexed. This further expand the number of channels and flexibilities of pin usages Because the ADC result is in 10-bit format, the ADC configuration can be set either using MSB aligned or LSB aligned format. This can speed up CPU processing by avoiding data alignment operations when reading the ADC results.

The configurations and result registers are defined as following.

ADCCFG (00A9h) ADC Configuration Register RW 0000000

	7	6	5	4	3	2	1	0
RD	**ADCEN**	**ATTN**	**ADCINTE**	**CSTART**	**ADCFM**	**-**	**PRE1**	**PRE0**
WR	**ADCEN**	**ATTN**	**ADCINTE**	**CSTART**	**ADCFM**	**CONT**	**PRE1**	**PRE0**

ADCEN ADC Macro Enable Bit
Set to enable ADC macro. When cleared, ADC enters into power-down mode.

ATTN ADC Attenuation Enable Bit.
Set to attenuate the input by 0.5X.

ADCINTE ADC Interrupt Enable Bit
Set this bit 1 will enable ADC interrupts.

CSTART Conversion Start Bit
Set this bit 1 will trigger an ADC conversion. This bit is self cleared when the conversion is done. The conversion is done on the selected channel. Thus using this mode must ensure only one channel is selected.

ADCFM ADC Result Format Control Bit
ADCFM = 1 will set ADC result as MSB justified. ADCAH = ADC[9:2], ADCAL = ADC[1:0]:000000.
ADCFM = 0 will set ADC result as LSB justified. ADCAH = 000000:ADC[9:8], ADCAL = ADC[7:0].

CONT Conversion Mode Control Bit
Set this bit 1 will make ADC into a continuous conversion mode. The ADC cycles through the selected channels and perform conversion on each channel continuously.

PRE1, PRE0 ADC Clock Pre-Scaler

PRE1	PRE0	ADC CLOCK
0	0	SYSCLK/2
0	1	SYSCLK/4
1	0	SYSCLK/8
1	1	SYSCLK/16

ADCCHSL (00B9h) ADC Channel Selection and Interrupt Status RW 0000XXXX

	7	6	5	4	3	2	1	0
RD	**ADCCHA**	**ADCCHB**	**ADCCHC**	**ADCCHD**	**CHAIF**	**CHBIF**	**CHCIF**	**CHDIF**
WR	**ADCCHA**	**ADCCHB**	**ADCCHC**	**ADCCHD**	**-**	**-**	**-**	**-**

ADCCHA, ADCCHB, ADCCHC, ADCCHD ADC Channel Selection Bits
Setting corresponding bits will connect the channels to ADC for conversion. In single

conversion mode, only channel should be selected any one time. For continuous mode, multi-channel can be selected and ADC will cycle through the selected channel.

CHAIF, CHBIF, CHCIF, CHDIF ADC Interrupt Flag Bits

These bits are set by ADC indicating the corresponding conversion is completed and new results are available on corresponding data registers. These bits are cleared when corresponding result register is read.

ADCAL (00BAh) ADCAH (00BBh) ADC Channel A Result Register RO XXXXXXXX

	7	6	5	4	3	2	1	0
RD	CH A Result							
WR	-	-	-	-	-	-	-	-

ADCBL (00BCh) ADCBH (00BDh) ADC Channel B Result Register RO XXXXXXXX

	7	6	5	4	3	2	1	0
RD	CH B Result							
WR	-	-	-	-	-	-	-	-

ADCCL (00BEh) ADCCH (00BFh) ADC Channel C Result Register RO XXXXXXXX

	7	6	5	4	3	2	1	0
RD	CH C Result							
WR	-	-	-	-	-	-	-	-

ADCDL (00AAh) ADCDH (00ABh) ADC Channel D Result Register RO XXXXXXXX

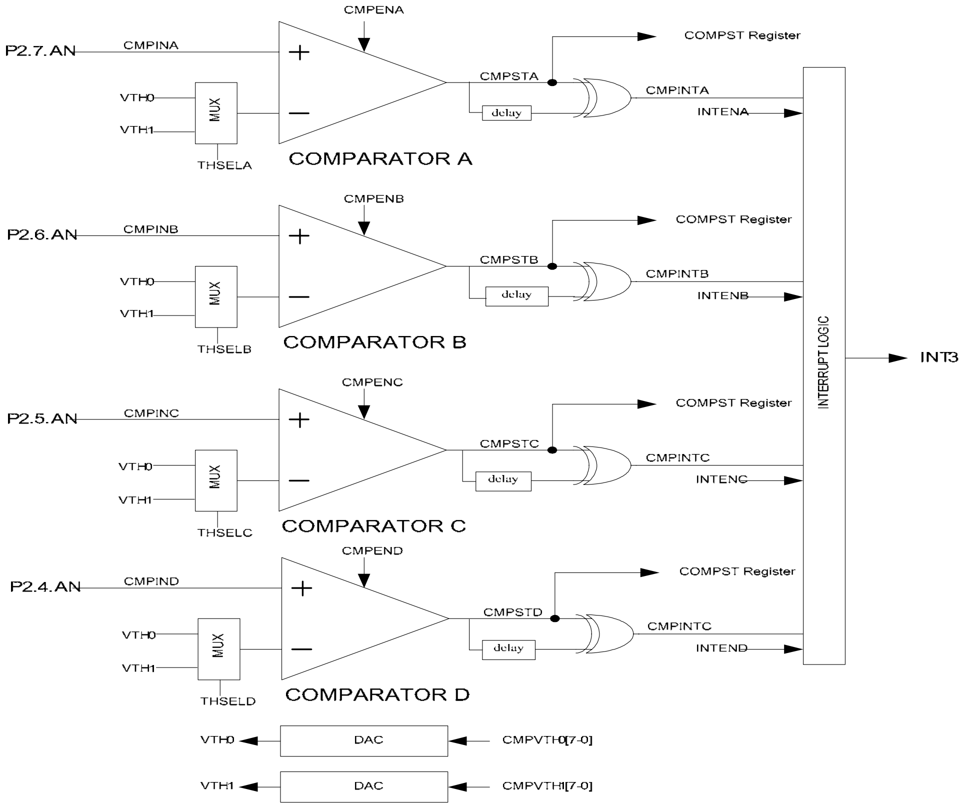
	7	6	5	4	3	2	1	0
RD	CH D Result							
WR	-	-	-	-	-	-	-	-

In CS8953, each channel of the ADC is further multiplexed to more than one pin. The corresponding setting in the MFCFGPxy that controls the configuration of multi-function pin must be configured accordingly to allow proper function of ADC.

3.7 Analog Comparator

CS8953 has four analog comparators as its on-chip external peripherals. When enabled, each comparator consumes about 250uA. There are two 8-bit R-2R DAC associated with the comparators to generate the compare threshold. The R-2R DAC uses the internal 2.5V supply as the full scale range thus limiting the comparator threshold from 0V to 2.5V in 256 steps. The inputs of the comparators are multiplexed with multi-function GPIO pins, P2.4, P2.5, P2.6, and P2.7. The CMPINA is further expanded to the analog input of P3.2 and P3.3. For certain applications, this can be used as time multiplexed comparators.

The block diagram of the analog comparator is shown in the following diagram.



The following registers are used for configuration and interface for the analog comparator peripherals.

CMPCFGAB (0xA030h) Analog Comparator A/B Configuration Register RW 0000000

	7	6	5	4	3	2	1	0
RD	**CMPENA**	**THSELA**	**INTENA**	**-**	**CMPENB**	**THSELB**	**INTENB**	**-**
WR	**CMPENA**	**THSELA**	**INTENA**	**-**	**CMPENB**	**THSELB**	**INTENB**	**-**

CMPENA Comparator A enable bit. Set to enable the comparator.
THSELA Comparator A threshold select bit. THSELA = 0, the comparator A uses VTH0 as the threshold. THSELA = 1, the comparator A uses VTH1 as the threshold.
INTENA Set to enable the comparator A interrupt.

CMPENB Comparator B enable bit. Set to enable the comparator.
THSELB Comparator A threshold select bit. THSELB = 0, the comparator A uses VTH0 as the threshold. THSELB = 1, the comparator A uses VTH1 as the threshold.
INTENB Set to enable the comparator B interrupt.

CMPCFGCD (0xA031h) Analog Comparator C/D Configuration Register RW 0000000

	7	6	5	4	3	2	1	0
RD	**CMPENC**	**THSELC**	**INTENC**	**-**	**CMPEND**	**THSELD**	**INTEND**	**-**
WR	**CMPENC**	**THSELC**	**INTENC**	**-**	**CMPEND**	**THSELD**	**INTEND**	**-**

CMPENC Comparator C enable bit. Set to enable the comparator.

THSELC Comparator C threshold select bit. THSELC = 0, the comparator C uses VTH0 as the threshold. THSELC = 1, the comparator C uses VTH1 as the threshold.
INTENC Set to enable the comparator C interrupt.

CMPEND Comparator D enable bit. Set to enable the comparator.
THSELD Comparator A threshold select bit. THSELD = 0, the comparator D uses VTH0 as the threshold. THSELD = 1, the comparator A uses VTH1 as the threshold.
INTEND Set to enable the comparator D interrupt.

CMPVTH0 (0xA032h) Analog Comparator Threshold Control Register RW 0000000

	7	6	5	4	3	2	1	0
RD	**VTH0 Register**							
WR	**VTH0 Register**							

CMPVTH0 register controls the comparator threshold VTH0 through 8-bit DAC. When set to 0x00h, the threshold is 0V. When set to 0xFFh, the threshold is at 2.5V. When not used, it should be set to 0x00 to save power consumption.

.

CMPVTH1 (0xA032h) Analog Comparator Threshold Control Register RW 0000000

	7	6	5	4	3	2	1	0
RD	**VTH0 Register**							
WR	**VTH0 Register**							

CMPVTH0 register controls the comparator threshold VTH0 through 8-bit DAC. When set to 0x00h, the threshold is 0V. When set to 0xFFh, the threshold is at 2.5V. When not used, it should be set to 0x00 to save power consumption.

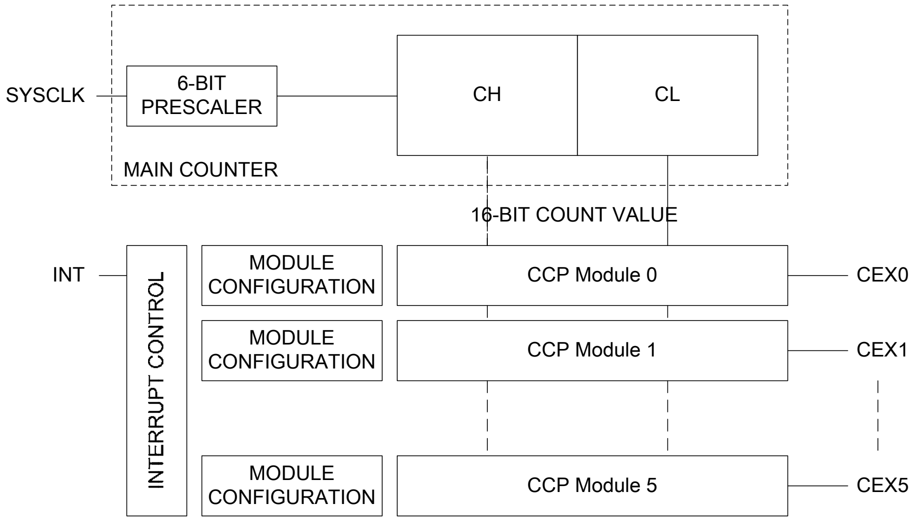
CMPST (0x94h) Analog Comparator Status Register RO 0000000

	7	6	5	4	3	2	1	0
RD	**CMPIFD**	**CMPIFC**	**CMPIFB**	**CMPIFA-**	**CMPSTD**	**CMPSTC**	**CMPSTB**	**CMPSTA**
WR	**CMPIFD**	**CMPIFC**	**CMPIFB**	**CMPIFA-**				

CMPIFD : comparator channel-D interrupt flag , this bit is set when CMPSTD is toggled and the comparator-D setting is enabled. Cleared by software .
CMPIFC : comparator channel-C interrupt flag , this bit is set when CMPSTC is toggled and the comparator-C setting is enabled. Cleared by software .
CMPIFB : comparator channel-B interrupt flag , this bit is set when CMPSTB is toggled and the comparator-B setting is enabled. Cleared by software .
CMPIFA : comparator channel-A interrupt flag , this bit is set when CMPSTA is toggled and the comparator-A setting is enabled. Cleared by software.
CMPSTx : these signals when read reflects the output of the analog comparator. If the corresponding comparator is disabled, the output will be forced to low.

3.8 Programmable Counter Array (PCA)

The PCA provides more enhanced timing functions with less CPU intervention than the standard 8051 timers T0, T1, and T2. The PCA is partitioned in two parts. The first is the main PCA Counter consists of CH:CL. The latter part consists up to 6 channels of Compare/Capture/PWM modules. This is shown as the following figures.



The output of the 16-bit counter (CH and CL) is sent to the modules for time base. And each CCP module receives the count value for comparison or capture purpose.

Main Counter

The MAIN COUNTER is configured and controlled by two register CMOD and CCON, and the counter value is accessed by CH and CL registers..

CMOD (0xD1) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**CIDL**	**CPS5**	**CPS4**	**CPS3**	**CPS2**	**CPS1**	**CPS0**	**ECF**
WR	**CIDL**	**CPS5**	**CPS4**	**CPS3**	**CPS2**	**CPS1**	**CPS0**	**ECF**

CIDL Counter Idle Control. Set to set PCA to be gated off during idle.

CPS5 – CPS0 PCA Pre-Scaler Setting. The SYSCLK is divided by CPS5:CPS0 and used by CH and CL. The effective clock of the main counter is SYSCLK / (CPS[5-0] + 1).

ECF Enable Counter Overflow Interrupt. Set to enable the main counter overflow to generate the interrupt.

CCON (0xE1) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**CF**	**CR**	**CCF5**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**
WR	**CF**	**CR**	**CCF5**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**

CF Counter Overflow Flag. Set by hardware when the main counter rolls over, and must be cleared by software.

CR Counter Run Control Bit. Set to turn the main counter on, and clear to turn off the counter.

CCF5 – CCF0 Module 5 – 0 Interrupt Flag. Set by the hardware in the module when a compare match occurs, and must be cleared by software.

CH (0xE9) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**CH7**	**CH6**	**CH5**	**CH4**	**CH3**	**CH2**	**CH1**	**CH0**
WR	**CH7**	**CH6**	**CH5**	**CH4**	**CH3**	**CH2**	**CH1**	**CH0**

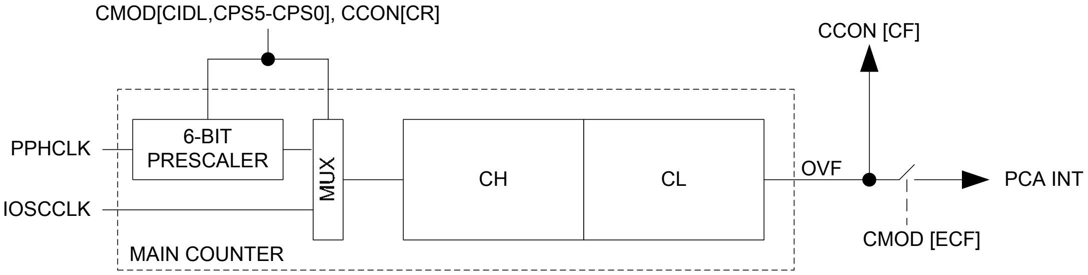
CH holds the upper 8-bit of the main counter value.

CL (0xD9) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**CL7**	**CL6**	**CL5**	**CL4**	**CL3**	**CL2**	**CL1**	**CL0**
WR	**CL7**	**CL6**	**CL5**	**CL4**	**CL3**	**CL2**	**CL1**	**CL0**

CL holds the lower 8-bit of the main counter value.

The simplest application using the PCA main counter is to configure it as a continuous timer. The PCA generates an interrupt when the 16-bit counter overflows. This is shown as the following diagram



COMPARE/CAPTURE and PWM Module (CCP)

The Compare/Capture modules receive the 16-bit count value from the main counter as the time base. Each module are configured by its configuration register CCAPMn, and contains two 8-bit registers used for compared value holder or for capture value storage. There are several basic modes of operation for CCP modules, and each CCP module can be configured in the same or different modes.

CCAPMn (0xB2, 0xB3, 0xB4, 0xB5, 0xB6, 0xB7) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**OF**	**ECOM**	**CAPP**	**CAPN**	**MAT**	**TOG**	**PWM**	**ECCF**
WR	**OF**	**ECOM**	**CAPP**	**CAPN**	**MAT**	**TOG**	**PWM**	**ECCF**

OF, TOG When the module is configured as Timer/Comparator modes, these two bits determine the output flag status on pin CEX when timer is up or comparison matches. The setting only affects the CEX and does not impact the interrupt generation

OF	TOG	CEX
0	0	CEX is unchanged
0	1	CEX toggles
1	0	CEX change to low (or remains low)
1	1	CEX change to high (or remains high)

When the module is configures as PWM mode, OF is don’t care, and TOG=1 will set CEX output to high constantly regardless of PWM value.

ECOM Enable Comparator. Set to enable comparator function. Clear to disable the comparator.

CAPP Capture Positive. Set to use positive edge as the capture edge. Clear to disable positive edge capture.

CAPN Capture Negative. Set to use negative edge as the capture edge. Clear to disable negative edge capture

MAT Match Control Bit. When MAT = 1, a match of CH/CL with CCAPH/CCAPL will cause CCF to be set and generate an interrupt.

PWM Pulse Width Modulation. Set to enable PWM function. CEX is the PWM output.

ECCF Enable capture/compare Interrupt. Set to enable the CCP module n to generate the capture/compare interrupt.

CCAPnL (0xD2, 0xD4, 0xD6, 0xE2, 0xE4, 0xE6) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	**CCAPnL**							
WR	**CCAPnL**							

CCAPnL register hold the compare value or capture value. It is also used as PWM value register.

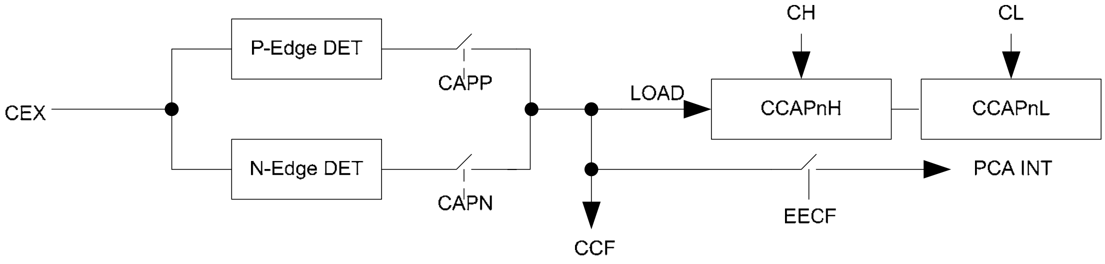
CCAPnH (0xD3, 0xD5, 0xD7, 0xE3, 0xE5, 0xE7) R/W (0x00)

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

CCAPnH register hold the compare value or capture value. It is also used as PWM value register.

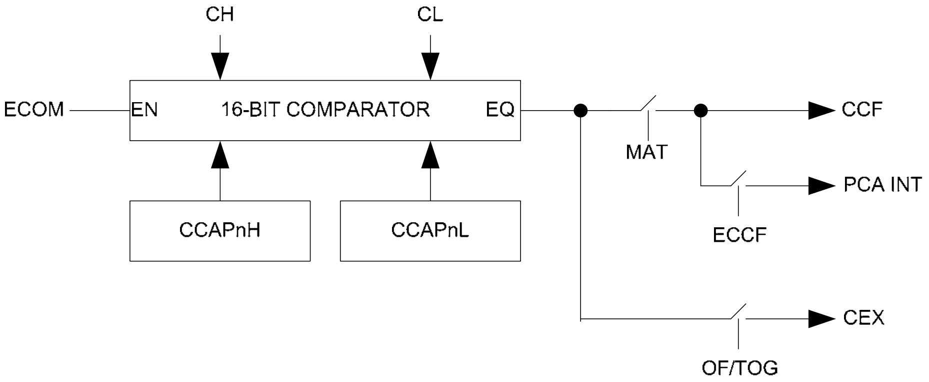
CAPTURE MODE

The capture mode is used to measure the elapse time of an external event between the edges of the external signal enabled when either CAPP or CAPN is set. The external CEX is sampled for transition. When a valid capture edge occurred in CEX, the current CH/CL count value is loaded into CCAPnH and CCAPnL register. At the same time CCFn in CCON register is set, and interrupt is generated if enabled. The block diagram showing this configuration is shown in following diagram.



COMPARE/TIMER/HIGH SPEED OUTPUT MODE

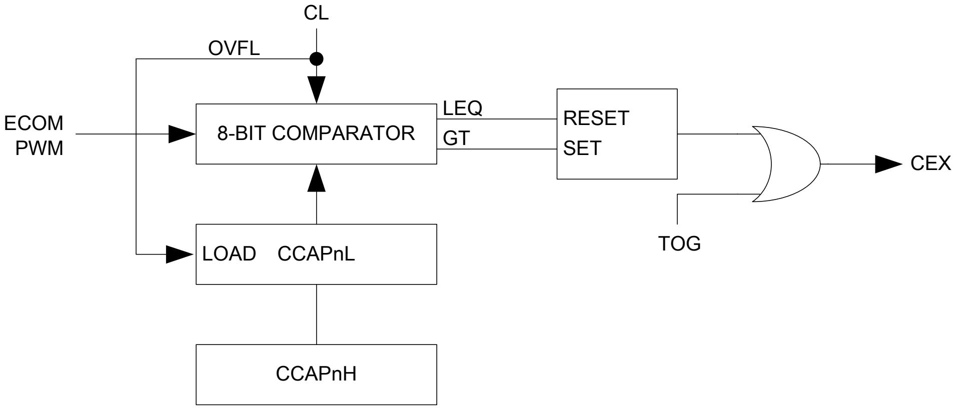
The compare timer mode can be used as a software timer. This mode is enabled when ECOM is set and CAPP, CAPN are set to low. To allow the compare result to be used, MAT/ECCF also needs to be set. The CCAPnH and CCAPnL hold the 16-bit Timer value and are compared against the incrementing value CH and CL from the main counter. When a match occurs, CCF is set and interrupt is generated. The block diagram of this mode is shown as following.



The match result can also be used to generate CEX output change. Depending on the CCAPM's OF and TOG setting, CEX output is changed at the compare match instant. However, the triggering of the change of CEX does not require MAT qualifier. Using CEX, waveform of precision duty cycle waveform or frequency modulation can be generated. The effect of OF/TOG on CEX is described in CCAPM register. To avoid unwanted glitch or match condition when updating the CCAPnH ad CCAPnL registers, once ECOM is set, writing CCAPnL will clear ECOM. Writing to CCAPnH will set ECOM to start the comparator. Therefore user program should update CCAPnL first and then CCAPnH. Of course, ECOM bit can be controlled directly through CCAPMn register.

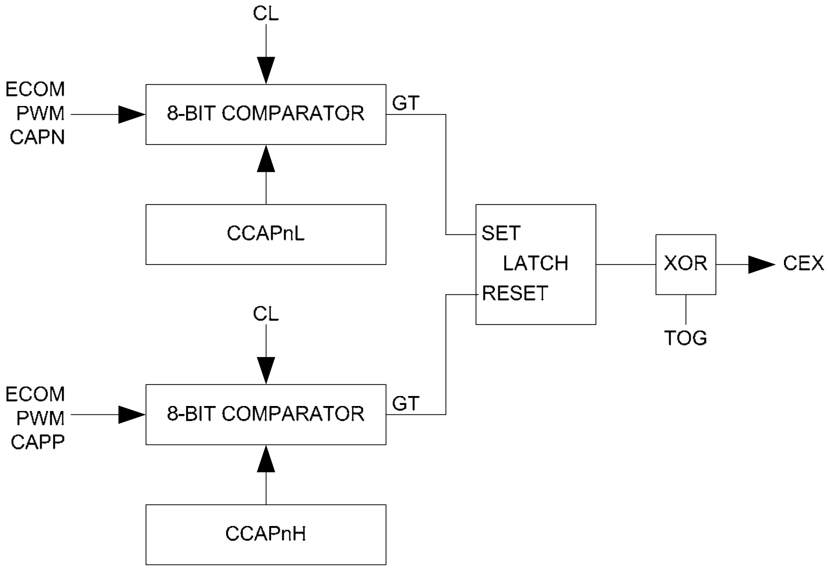
PULSE WIDTH MODULATOR (PWM) MODE

This mode is used to generate 8-bit precision PWM output on CEX. The time base of the PWM is provided by CL of the main counter. CCAPnH and CCAPnL are used as a double buffer thus preventing glitches on CEX output when PWM data is changed. CCAPnL is used to compare with CL value. When CL is less than or equal to CCAPnL, CEX is 0. When CL is greater than CCAPnL, CEX is high. When CL overflows, CCAPnL is reloaded from CCAPnH. The PWM mode is enabled when ECOM and PWM bits are both set, and CAPP, CAPN are both low. Note that under the above compare method, maximum CEX duty cycle is 255/256. If TOG is set to 1 in this mode, CEX will be forced high to provide 256/256 full high duty cycle. The following block diagram shows the PWM mode operation.



WINDOWED PULSE WIDTH MODULATOR (WPWM) MODE

This mode is used to generate 8-bit precision PWM output on CEX. The difference from regular PWM mode is that the CEX becomes high during a window of CL count. CEX becomes high when CL is greater than CCAPnL, CEX is reset to low when CL is greater than CCAPnH. Therefore the value in CCAPnH must be larger than CCAPnL to prevent abnormal operations. This mode is enabled when ECOM, PWM, and both CAPP and CAPN are set to 1. If TOG is 1, then CEX is inverted.



CCP FUNCTION SUMMARY

CCP Function		OF	ECOM	CAPP	CAPN	MAT	TOG	PWM
No operation		X	0	0	0	X	X	0
16 bit capture mode	Triggered by positive edge of CEX	X	X	1	0	X	X	0
	Triggered by negative edge of CEX			0	1			
	Triggered by both edges of CEX			1	1			
16 bit compare mode	CEX is unchanged	0	1	0	0	0/1	0	0
	CEX toggles	0					1	
	CEX change to low (or remains low)	1					0	
	CEX change to high (or remains high)	1					1	

8 bit PWM mode	CEX = 0 when CL < = CCAPnL CEX = 1 when CL > CCAPnL	X	1	0	0	X	0	1
	CEX = 1						1	
8 bit WPWM mode	CEX = 0 when CL > CCAPnH CEX = 1 when CL > CCAPnL	X	1	1	1	X	0	1
	CEX = 1 when CL > CCAPnH CEX = 0 when CL > CCAPnL						1	

Note:

1. In 16 bit compare mode, ECOM can be set to 1 by hardware (when writing to CCAPnH) or software, and can be cleared to 0 by hardware (when writing to CCAPnL) or software. When ECOM is cleared to 0 in this mode, the CCP function will enter into no operation mode.
2. ECOM can not be set to 1 by hardware (when writing to CCAPnH) if all bits (OF, ECOM, CAPP, CAPN, MAT, TOG, PWM) in CCAPM are 0 (no operation mode).

3.9 I2C Slave Controller

The I2C slave is controlled through SFR. There are up to two I2C slave controllers depending on the device implementation for Myson-Century MCU platform. In CS8953, there is only one I2C slave controller.

The address of the slave is set by I2CSADR1 register. The MSB in I2CSADR1 is the enable bit control for the I2C slave controller. I2CSADR1[6-0] specifies the actual slave address.

In receive mode, the controller detects a valid matching address and issues an ADDRMI interrupt. At the same time, the data bit on SDA line is shifted into receive buffer. The RCBI interrupt is generated whenever a complete byte is received and is ready to be read from I2CSDAT. If for any reason, the software does not respond to RCBI interrupt in time (i.e. RCBI is not cleared), and a new byte is received, the controller will either force an NACK response on I2C (if CLKSTREN bit is not set) or by pulling and holding SDA low (if CLKSTREN bit is set) to stretch the SCL low duration to force the master into a wait state. In clock stretching mode, SCL is released when the software responds to RCBI interrupt and cleared RCBI flag.

In transmit mode, the controller detects a valid matching address and issue an ADDRMI interrupt. At the same time, the data preloaded in the transmit data register through I2CSDAT is transferred to the transmit shift register and is serially shifted out onto SDA line. When this is occurring, the controller generates a TXBI interrupt to inform the software that a new byte can be written into I2CSDAT. When the shift register is empty and ready for the next transmit, the slave controller checks if new byte is been written to the I2CSDAT. If TXBI is still not cleared indicating no new data is available, the slave controller will hold SCL line low to stretch the current clock cycle if CLKSTREN is set. If clock stretching is not enabled, the slave controller will take the old byte into the shift register and reply with NACK, and thus will cause data corruption. On the other hand, if the master returns the NACK after the byte transfer, this indicates the end of data to the I2C slave. In this case, the I2C slave will release the data line to allow the master to generate a STOP or repeated START condition.

The I2C slave controller also implements the input noise spike filter. This is enabled by the INFILEN bit in the I2CSCON register. The filter is implemented using digital circuit. When INFILEN is set, the spike under 1/2 EPPCLK period on the input of SDA and SCL lines will be filtered out. If INFILEN is low, no input filtering is done.

The following registers are related to I2C Slave Controller.

I2CSCON1 (0xEB) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	**XMT**
WR	**I2CSRST**	**EADDRMI**	**ESTOPI**	**ERPSTARTI**	**ETXBI**	**ERCBI**	**CLKSTREN**	**INFILEN**

I2CSRST I2C Slave Reset Bit. Set this bit will cause the Slave Controller to reset all internal state machine. Clear this bit for normal operations. Set this bit will also clear the I2CSADR1 (I2C slave address).

EADDRMI ADDRMI interrupt Enable Bit. Set this bit will allow ADDRMI interrupt as the I2C slave interrupt. This interrupt is generated when I2C slave received a matching address.

ESTOPI STOPI Interrupt Enable Bit. Set this bit will allow STOPI interrupt as the I2C slave interrupt.

ERPSTARTI RPTSTARTI Interrupt Enable Bit. Set this bit will allow RPTSTARTI interrupt as the I2C slave interrupt.

ETXBI TXBI Interrupt Enable Bit. Set this bit will allow TXBI interrupt as the I2C slave interrupt.

ERCBI RCBI Interrupt Enable Bit. Set this bit will allow RCBI interrupt as the I2C slave interrupt.

CLKSTREN Set to enable the clock stretching function of the slave controller. Clock stretching is an optional feature defined in I2C specification.

INFILEN Set this bit to enable the input noise filter of SDA and SCL lines. When the filter is enabled, it will filter out the spike of less than 50nsec.

XMT This bit is set by the controller when the I2C slave is in transmit operation, and is 0 when the I2C slave controller is in receive operation.

I2CSST1 (0xEC) R/W (0x00)

	7	6	5	4	3	2	1	0

RD	FIRSTBT	ADDRMI	STOPI	RPSTARTI	TXBI	RCBI	START	NACK
WR	-	ADDRMI	STOPI	RPSTARTI	HOLDT[3]-	HOLDT[2]	HOLDT[1]	HOLDT[0]

FIRSTBT This bit is set to indicate the data in the data register is the first byte received after address match. This bit is cleared after the first byte of the transaction is read. This bit is generated by slave controller and is read-only.

ADDRMI Slave Address Match Interrupt Flag. This bit is set when the received address matches the address defined in I2CSADR1. If EADDMI is set, this will generate an interrupt. This bit must be cleared by software.

STOPI Stop Condition Interrupt Flag. This bit is set when the slave controller detects a STOP condition on the SCL and SDA lines. This bit must be cleared by software.

RPTSARTI Repeat Start Condition Interrupt Flag. This bit is set when the slave controller detects a repeat Start condition on the SCL and SDA lines. This bit must be cleared by software.

TXBI Transmit Buffer Interrupt Flag. This bit is set when the slave controller is ready to accept a new byte for transmit. This bit is cleared when new data is written into I2CSDAT register.

RXBI Receiver Buffer Interrupt Flag. This bit is set when the slave controller have put a new data in the I2CSDAT and ready for software to read. This bit is cleared after the software read out the I2CSDAT.

START Start Condition. This bit is set when the slave controller have detected a Start condition on the SCL and SDA lines. This bit is not very useful as the start of transaction can be indicated by address match interrupt. This bit read-only and is cleared when Stop condition is detected.

NACK NACK Condition. This bit is set when the host has responded with NACK in the byte transaction. This bit is only meaningful for slave-transmit operation. Please note if the master returns with NACK on the byte transaction, the slave will not upload new data into the shift register. And the slave will re-transmit the old data again in the next transfer, and this re-transmission will continue if NACK condition is repeated until the transmission is successful with ACK returned.. This bit is cleared when a new ACK condition is detected or it can be cleared by software.

HOLD[3-0] These four bit defines the number of EPPCLK cycles that SDA to SCL hold time. The I2C specification requires for minimum of 300nsec hold time. For example, if EPPCLK is 4MHz, then HOLD[3-0] should be set as 2. If EPPCLK is 20MHz, then HOLD[3-0] should be set to 10.

I2CSADR1 (0xED) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	I2CSEN	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

I2CSENT Set this bit to enable the I2C slave controller.

ADDR6-ADDR0 These are the 7 bits slave address.

I2CSDAT1 (0xEE) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	I2C Slave Receive Data Register							
WR	I2C Slave Transmit Data Register							

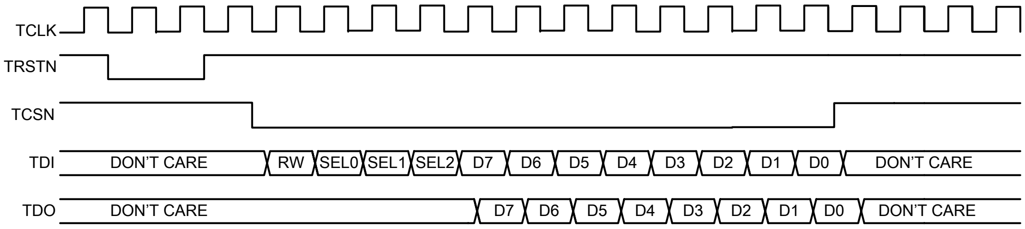
4.0 Flash Programming and ISP/IAP

The embedded Flash memory can be accessed in two different methods, through writer mode or software program using Flash controller. The Flash memory has two blocks, a 128Byte Information Block and 32KB/64KB/128KB main memory block. The main memory block is organized in sectors, each sector contains 512Byte. The characteristic of flash is that the native (after erasure) state is "1", and can be written to state "0". Once written to "0", a flash memory bit must be erased to return to state "1". And the flash can not be erased byte by byte, instead the minimum space to perform erase operation is one sector. In similar fashion, the whole IFB is in one sector, thus it can only be erased as a whole.

4.1 Writer Programming Mode

Entry to parallel programming mode is by raising RSTN pin to 3.5V. In this mode, all the pin functions of all the pins are disabled and are used for parallel programming purpose. And there are 5 signal pins for writer mode are multiplexed out to perform the writing and verification of the Flash Memory. These are TRSTN, TESTN, TCLK, TDI, and TDO.

The writer mode command is issued similar to a serial interface, the DUT work as the salve while the writer host acts as the master. The protocol is in command packet format and is enclosed by TCSN low. TDI is sampled at the rising edge of the TCLK. TDO is valid output after the fifth valid TCLK rising edge, if the command is a READ operation with RW=0. .



TCLK should be a continuous clock at a fixed frequency of 1MHz and duty cycle of 50%. The command is executed at the first rising edge of TCLK after TCSN is high. It is recommended that TRSTN is brought to low for at least 2 cycles to reset the writer interface. There should be a minimum of two clock edges between two packets. The WRITER mode provides the following commands to operate on the embedded flash memory.

Erase Main Memory
Erase Main Memory and IFB (Mass Erase)
Write Byte Main Memory
8-Byte Verify
Write Byte IFB
Read Byte IFB

There is no READ Byte command. Therefore once written, the content of the main memory can only be checked through Sector Verify command. The Verify command receives the starting byte address and the data of 16 bytes then responds with success if the whole 16 bytes data matches with the stored data.

With this arrangement, the inherent content security from in WRITER mode is extremely good, and the chance of illegal access of the content is (2^8)^8 = 10E19. This is an extremely large number and thus unbreakable even with an average trial rate of 1x10E6 per second yield the break time of more than thirty thousand years..

The WRITER mode is used by manufacturer to write ISP boot code into the top sectors of main memory and manufacturer device information in IFB.

4.2 In-System-Programming and In-Application-Programming

The ISP is implemented in software. The software implementations of ISP allow maximum flexibilities in applications. Each device is pre-loaded with manufacture-supplied ISP boot code (here we refer to MISP boot code) in the top sectors (located in 0xFE00) of the main memory. This is also the starting program address after power-on or reset. Therefore, ISP code is always executed. Depending on the application requirement, although not recommended, the user can replace the MISP code with user ISP code or use this as user program using WRITER mode. The user program then can optimize the ISP and IAP requirements.

MISP only supports the following commands to ensure content security.
ISP Identify
ISP Erase Main Memory (except MISP itself)
ISP Write Byte Main Memory (except MISP itself)
ISP 8-Byte Verify
ISP Write Byte IFB (only User Data)
ISP Read Byte IFB
ISP Exit

These commands are similar to WRITER mode. The ISP Identify is the command sent by host to identify the ISP connection is valid. ISP Exit command is issued at the end of ISP, and the MISP program ends and user program starting at 0x0000 is entered.

The following sequences outline the MISP flow.

1. Initialization of REG, and IOSC with calibrated data from IFB.
2. Check if (ISPWAIT EQ 00) then jump to 0000 for user program start
3. Check UART ISP Identity condition and scan for ISPWAIT * 1msec
4. If no UART ISP activity, then jump to 0000 for user program start
5. If UART ISP activity, then enter in ISP command interpretation loop

The executions of ISP command interpretation depends on the value of ISPWAIT which is stored in the IFB. The default un-programmed value of ISPWAIT is FF. Therefore, MISP code will scan 256 msec for valid ISP IDENTIFY command in UART after reset, if no ISP IDENTIFY command is present, MISP will exit and jump to user program. This wait time can be modified according to the application requirement. For example, if the application can ensure ISP IDENTIFY command will be present in less than 41 msec, ISPWAIT can be programmed (through ISP) to 0x19. The extreme case is that ISPWAIT can be programmed to 00 and MISP is not executed at all. Please note, once ISPWAIT can only be programmed once, it can not be re-written unless IFB is erased in writer mode using mass erase command.

The Erase Main Memory Command does not erase the MISP boot code itself. The main memory portion defined by “MISP program sector count” in IFB is protected against modification. This restriction also applies to ISB Write Byte Main Memory command. And ISP Write byte IFB command can only modify the contents in the User data defined in IFB.

The MISP code implementation is based on the use of Flash Controller command which provide the capabilities of sector erase, byte write, IFB byte write, byte read and IFB byte read as described in previous sections. Thus how to do ISP is a pure software implementation. The MISP can be replaced by user ISP code (UISP) and UISP can be written through writer mode before the device is mounted on the system board. Or UISP can reside in user program space that can be programmed through MISP procedures and co-exist with MISP. Therefore it is possible to use other interface to accomplish ISP or IAP, for example through any GPIO, or any selected UART.

MISP implementation provides the same security level as in writer mode due to only Sector Verify command is allowed for content reading. Thus to read the content our, a user must know the content beforehand and only verifying whether the contents agree with the known content.

4.3 Information Block (IFB)

The IFB contains critical manufacture and calibration information. Therefore should not be erased by the user. If for any reason, the IFB needs to be erased, the original content that containing this critical information should be written back to the IFB. The following table described the usage of IFB. The column marked M/U indicates whether it is manufacturer data or user data. The User data are all FF for fresh devices. User program can use the User data location as a one-time nonvolatile storage. And if IFB needs to be erased due to modification of user data, then all M data needs to be read out and copied back. If this is not done, then it will violate the warrantees.

ADDRESS	M	DESCRIPTION
00 - 07	M	Device Name
08 – 09	M	Device Version

0A – 0F	M	Die Record
10 – 17	M	Lot number
18 – 1F	M	FT Date Code
20	M	REGTRM value for 2.5V
21	M	IOSC ITRM value for 3.68MHz
22	M	IOSC VTRM value for 3.68MHz
23	M	LVDTHD value for detection of 4.5V
24	M	LVDTHD value for detection of 3.0V
25 – 27	R	Reserved and can not be written
28 – 2C	M	MISP program version
2D – 2F	M	MISP program sector count
30 – 37	M	Security Key and CRC for (00 – 2F)
38 – 3F	R	Reserved and can not be written
40	U	SPWAIT indicates how long MISP needs to scan for ISP condition
41 – 7F	U	User data

Note:

1. M data can not be modified and can only be written in writer mode when the entire Flash is freshly erased.
2. U data read out as FF after the Flash is erased. It can only be programmed once after the Flash is mass erased.

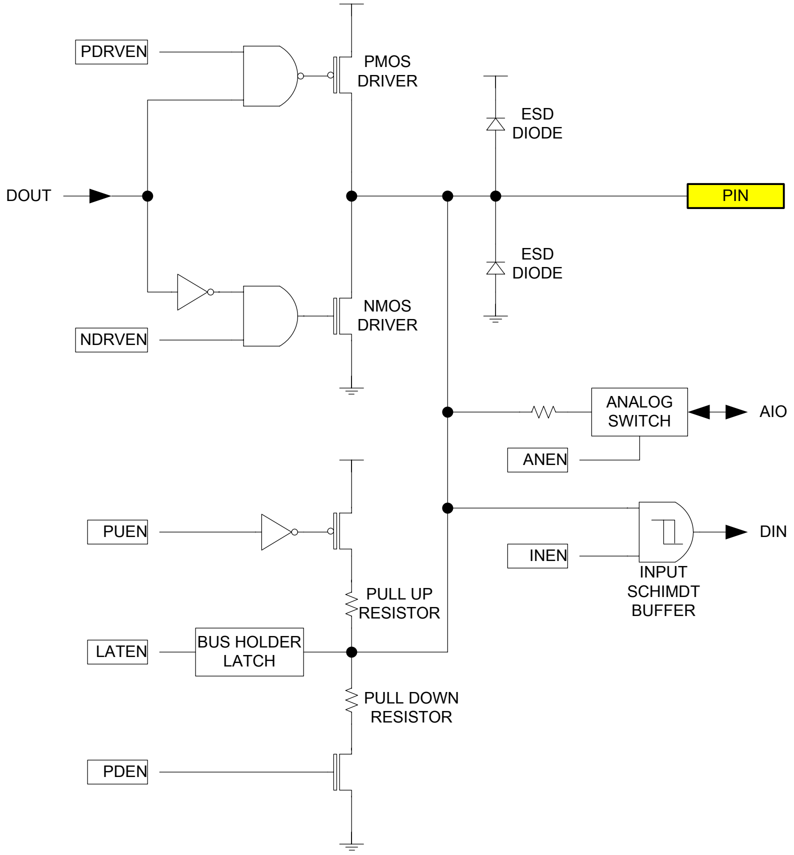
5.0 Pin Function and Configurations

This section describes the pin functions and configurations. Almost every signal pin is a multi-function pin with default function as a GPIO port pin. Therefore each signal pin requires two registers to configure the I/O capability and the function selection. The following describes the control and contents of these register and the registers names and pin names are referenced by their default GPIO port name. For example, in CS8953, pin 26 of SSOR-28 and pin 20 of LQFP-32 is defined as P0.0/PINT1.0/ADA1. This means this pin has the multi-function of GPIO port 0.0, shared with Pin Interrupt 1.0, and ADC channel A1 input. The I/O capability and the multi-function select registers therefore are referred as P0.0 name.

Because every signal pin is a general purpose I/O as well as multi-functional pin, CS8953 employs a configurable I/O buffer design. The standardized I/O design allows flexible configuration of the digital I/O function such as open-drain, open-source, pull-up, pull-down, bus-holder capabilities. In addition to digital I/O function, the standardized I/O also provide analog I/O capability that can be selected when the GPIO pin is shared with analog peripheral purpose such as analog OPAMP, ADC input or DAC output.

The supply voltage of the I/O buffer uses VDD (3.0V to 5.5V). The input and output level is referenced to VDD and 0V. Since the design is standardized, the I/O design offers a uniform ESD performance.

The functional block diagram of the standard I/O buffer is shown in the following diagram.



From the diagram, there are 7 control bits for the IOCFGPx.y register, and these registers are located at XFR 0xA040 – 0xA047 for P0.0 to P0.7, 0xA048 – 0xA04F for P1.0 to P1.7, 0xA060 – 0xA067 for P2.0 to P2.7, and 0xA068 – 0xA06F for P3.0 to P3.7. The bit definitions of IOCFGPx.y are described in the following table.

IOCFGPx.y (0xA040 – 0xA04F, 0xA060 – 0xA06F) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	**INEN**	**LATEN**	**PUEN**	**PDEN**	**ANEN**	**PDRVEN**	**NDRVEN**	-

INEN Input buffer control. Set this bit to enable the GPIO's input buffer. If the input buffer is not used, it should be disabled to prevent leakage current when pin is floating. Default is disabled.

LATEN Bus holder latch control. Set this bit to enable the bus holder latch connected to the pin. When enable, the bus holder will hold the last actively driven state of the pin. The latch only provides a very weak drive therefore should not affect the signal when pin is actively driven. Default is disabled.

PUEN Pull up resistor enable control. Set this bit to enable pull-up resistor connection to the pin. The pull-up resistor is approximately 100K Ohm. Default is disabled.

PDEN Pull down resistor enable control. Set this bit to enable pull-down resistor connection to the pin. The pull-down resistor is approximately 100K Ohm. Default is disabled.

ANEN Analog MUX enable control. Set this bit to connect the pin to the internal analog peripheral. Default is disabled.

PDRVEN Output PMOS driver enable. Set this bit to enable the PMOS of the output driver. Default is disabled.

NDRVEN Output NMOS driver enable. Set this bit to enable the NMOS of the output driver. Default is disabled.

The following table shows various configurations of the I/O buffer.

IO Functions	INEN	LATEN	PUEN	PDEN	ANEN	PDRVEN	NDRVEN
Input only	1	0	0	0	0	0	0
Input /w pull up	1	0	1	0	0	0	0
Input /w pull down	1	0	0	1	0	0	0
Input /w bus holder	1	1	0	0	0	0	0
Output with CMOS push-pull	0	0	0	0	0	1	1
Output /w NMOS open-drain (sink)	0	0	0	0	0	0	1
Output /w NMOS open-drain (sink) and weak pull up	0	0	1	0	0	0	1
Output /w PMOS open-drain (source)	0	0	0	0	0	1	0
Output /w PMOS open-drain (source) and weak pull down	0	0	0	1	0	1	0
I/O 8051 like	1	1	1	0	0	0	1
I/O CMOS	1	0	0	0	0	1	1
Analog function	0	0	0	0	1	0	0
Oscillator pin	0	0	0	0	0	0	0

Because each signal pin is a multi-function pin that the function is shared with GPIO port, therefore each pin requires a MFCFGPx.y register to control which function is in effect and which peripherals are connected to the signal pin. These selection and definitions are pin specific and product specific. The following description describes the selection and control for CS8953 signal pins.

MFCFGP0.0 (0xA050) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	**PINTEDG**	**ADA1EN**	**PINTEN**	**GPIOEN**

P0.0 is shared with PINT1.0 and ADC Channel A1 input. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT1.0. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

ADA1EN Set this bit to enable the P0.0 pin as the ADC channel A1 input. Please note to enable this function, the IOCFGP0.0's ANEN bit must also be set. Default is disabled.

PINTEN Pin Interrupt enable control bit. Set this bit to enable the P0.0 pin as the pin interrupt PINT1.0. Default is disabled.

GPIOEN P0.0 GPIO function enable bit. Set this bit to enable P0.0's GPIO function. Default is disabled.

MFCFGP0.1 (0xA051) W (0x00)

	7	6	5	4	3	2	1	0

RD	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
WR	**-**	**-**	**-**	**-**	**PINTEDG**	**ADD1EN**	**PINTEN**	**GPIOEN**

P0.1 is shared with PINT1.1 and ADC Channel A2 input. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT1.1. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

ADD1EN Set this bit to enable the P0.1 pin as the ADC channel D1 input. Please note to enable this function, the IOCFGP0.1's ANEN bit must also be set. Default is disabled.

PINTEN Pin Interrupt enable control bit. Set this bit to enable the P0.1 pin as the pin interrupt PINT1.1. Default is disabled.

GPIOEN P0.1 GPIO function enable bit. Set this bit to enable P0.1's GPIO function. Default is disabled.

MFCFGP0.2 (0xA052) W (0x00)

	7	6	5	4	3	2	1	0
RD	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
WR	**-**	**-**	**-**	**-**	**PINTEDG**	**ADB1EN**	**PINTEN**	**GPIOEN**

P0.2 is shared with PINT1.2 and ADC Channel B1 input. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT1.2. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

ADB1EN Set this bit to enable the P0.2 pin as the ADC channel B1 input. Please note to enable this function, the IOCFGP0.2's ANEN bit must also be set. Default is disabled.

PINTEN Pin Interrupt enable control bit. Set this bit to enable the P0.2 pin as the pin interrupt PINT1.2. Default is disabled.

GPIOEN P0.2 GPIO function enable bit. Set this bit to enable P0.2's GPIO function. Default is disabled.

MFCFGP0.3 (0xA053) W (0x00)

	7	6	5	4	3	2	1	0
RD	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
WR	**-**	**-**	**-**	**-**	**PINTEDG**	**ADB2EN**	**PINTEN**	**GPIOEN**

P0.3 is shared with PINT1.3 and ADC Channel B2 input. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT1.3. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

ADB2EN Set this bit to enable the P0.3 pin as the ADC channel B2 input. Please note to enable this function, the IOCFGP0.3's ANEN bit must also be set. Default is disabled.

PINTEN Pin Interrupt enable control bit. Set this bit to enable the P0.3 pin as the pin interrupt PINT1.3. Default is disabled.

GPIOEN P0.3 GPIO function enable bit. Set this bit to enable P0.3's GPIO function. Default is disabled.

MFCFGP0.4 (0xA054) W (0x00)

	7	6	5	4	3	2	1	0
RD	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
WR	**-**	**-**	**-**	**-**	**PINTEDG**	**ADC1EN**	**PINTEN**	**GPIOEN**

P0.4 is shared with PINT1.4 and ADC Channel C1 input. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT1.4. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

ADC1EN Set this bit to enable the P0.4 pin as the ADC channel C1 input. Please note to enable this function, the IOCFGP0.4's ANEN bit must also be set. Default is disabled.

PINTEN Pin Interrupt enable control bit. Set this bit to enable the P1.4 pin as the pin interrupt PINT1.4. Default is disabled.

GPIOEN P0.4 GPIO function enable bit. Set this bit to enable P0.4's GPIO function. Default is disabled.

MFCFGP0.5 (0xA05) W (0x00)

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

WR	-	-	-	-	PINTEDG	ADC2EN	PINTEN	GPIOEN

P0.5 is shared with PINT1.5 and ADC Channel C2 input. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT1.5. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

ADC2EN Set this bit to enable the P0.5 pin as the ADC channel C2 input. Please note to enable this function, the IOCFGP0.5's ANEN bit must also be set. Default is disabled.

PINTEN Pin Interrupt enable control bit. Set this bit to enable the P0.5 pin as the pin interrupt PINT1.5. Default is disabled.

GPIOEN P0.5 GPIO function enable bit. Set this bit to enable P0.5's GPIO function. Default is disabled.

MFCFGP0.6 (0xA06) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	ADA2EN	TXD0EN	GPIOEN

P0.6 is shared with TXD0 and ADC Channel D1 input. Only one bit can be set during any time.

ADA2EN Set this bit to enable the P0.6 pin as the ADC channel A2 input. Please note to enable this function, the IOCFGP0.6's ANEN bit must also be set. Default is disabled.

TXD0EN Set this bit to enable P0.6 pin as TXD signal for UART0. Default is disabled.

GPIOEN P0.6 GPIO function enable bit. Set this bit to enable P0.6's GPIO function. Default is disabled.

MFCFGP0.7 (0xA07) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	ADD2EN	RXD0EN	GPIOEN

P0.7 is shared with RXD0 and ADC Channel D2 input. Only one bit can be set during any time.

ADD2EN Set this bit to enable the P0.7 pin as the ADC channel D1 input. Please note to enable this function, the IOCFGP0.7's ANEN bit must also be set. Default is disabled.

RXD0EN Set this bit to enable P0.7 pin as RXD signal for UART0. Default is disabled.

GPIOEN P0.7 GPIO function enable bit. Set this bit to enable P0.7's GPIO function. Default is disabled.

MFCFGP1.0 (0xA058) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	CEX0EN	ADD3EN	GPIOEN

P1.0 is shared with ADC Channel D3 input. Only one bit can be set during any time.

CEX0EN Set this bit to enable the P1.0 pin as CEX0 for PCA peripheral. Default is disabled.

ADD3EN Set this bit to enable the P1.0 pin as the ADC channel D3 input. Please note to enable this function, the IOCFGP1.0's ANEN bit must also be set. Default is disabled.

GPIOEN P1.0 GPIO function enable bit. Set this bit to enable P1.0's GPIO function. Default is disabled.

MFCFGP1.1 (0xA059) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	CEX1EN	ADD4EN	GPIOEN

P1.1 is shared with ADC Channel D4 input. Only one bit can be set during any time.

CEX1EN Set this bit to enable the P1.1 pin as CEX1 for PCA peripheral. Default is disabled.

ADD4EN Set this bit to enable the P1.1 pin as the ADC channel D4 input. Please note to enable this function, the IOCFGP1.1's ANEN bit must also be set. Default is disabled.

GPIOEN P1.1 GPIO function enable bit. Set this bit to enable P1.1's GPIO function. Default is disabled.

MFCFGP1.2 (0xA05A) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**MSDAEN**	**SSDAEN**	**GPIOEN**

P1.2 is shared with I2C Master and I2C Slave's SDA. Only one bit can be set during any time.

MSDAEN Set this bit to enable the P1.2 pin as the Master I2C slave SDA signal. The corresponding IOCFGP1.2 should also be set correctly as SDA signal should be an open-drain configuration. Default is disabled.

SSDAEN Set this bit to enable the P1.2 pin as the Master I2C slave SDA signal. The corresponding IOCFGP1.2 should also be set correctly as SDA signal should be an open-drain configuration. Default is disabled.

GPIOEN P1.2 GPIO function enable bit. Set this bit to enable P1.2's GPIO function. Default is disabled.

MFCFGP1.3 (0xA05B) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**MSCLEN**	**SSCLEN**	**GPIOEN**

P1.3 is shared with I2C Master and I2C Slave's SCL. Only one bit can be set during any time.

MSCLEN Set this bit to enable the P1.3 pin as the Master I2C slave SCL signal. The corresponding IOCFGP1.3 should also be set correctly as SCL signal should be an open-drain configuration. Default is disabled.

SSCLEN Set this bit to enable the P1.3 pin as the Master I2C slave SCL signal. The corresponding IOCFGP1.3 should also be set correctly as SCL signal should be an open-drain configuration. Default is disabled.

GPIOEN P1.3 GPIO function enable bit. Set this bit to enable P1.3's GPIO function. Default is disabled.

MFCFGP1.4 (0xA05C) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**RTCRXINEN**	**GPIOEN**

P1.4 is shared with RTC crystal oscillator RXIN. Only one bit can be set during any time.

RTCRXINEN Set this bit to enable the P1.4 pin as the RTC crystal in RXIN. The RTC crystal oscillator is considered as an analog peripheral. The corresponding IOCFGP1.4 ANEN should also be set. Default is disabled.

GPIOEN P1.4 GPIO function enable bit. Set this bit to enable P1.4's GPIO function. Default is disabled.

MFCFGP1.5 (0xA05D) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**RTCXOUTEN**	**GPIOEN**

P1.5 is shared with RTC crystal oscillator XIN. Only one bit can be set during any time.

RTCXOUTEN Set this bit to enable the P1.5 pin as the RTC crystal out RXOUT. The RTC crystal oscillator is considered as an analog peripheral. The corresponding IOCFGP1.5 ANEN should also be set. Default is disabled.

GPIOEN P1.5 GPIO function enable bit. Set this bit to enable P1.5's GPIO function. Default is disabled.

MFCFGP1.6 (0xA05E) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**PINTEDG**	**PINTEN**	**GPIOEN**

P1.6 is shared with pin interrupt PINT0.0. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT0.0. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

PINTEN Set this bit to enable the P1.6 pin as pin interrupt PINT0.0. Default is disabled.

GPIOEN P1.6 GPIO function enable bit. Set this bit to enable P1.6's GPIO function. Default is disabled.

MFCFGP1.7 (0xA05F) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**PINTEDG**	**PINTEN**	**GPIOEN**

P1.7 is shared with pin interrupt PINT0.1. Only one bit can be set during any time.

PINTEDGE This bit controls the active interrupt edge of PINT0.1. Set to use positive edge and clear to use negative edge. This bit is effective only PINTEN is set.

PINTEN Set this bit to enable the P1.7 pin as pin interrupt PINT0.1. Default is disabled.

GPIOEN P1.7 GPIO function enable bit. Set this bit to enable P1.6's GPIO function. Default is disabled.

MFCFGP2.0 (0xA070) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**XINEN**	**GPIOEN**

P2.0 is shared with crystal oscillator XIN. Only one bit can be set during any time.

XINTEN Set this bit to enable the P2.0 pin as crystal oscillator XIN. The crystal oscillator is considered as an analog peripheral. The corresponding IOCFGP2.0 ANEN should also be set. Default is disabled.

GPIOEN P2.0 GPIO function enable bit. Set this bit to enable P2.0's GPIO function. Default is disabled.

MFCFGP2.1 (0xA071) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**XOUTEN**	**GPIOEN**

P2.1 is shared with crystal oscillator XOUT. Only one bit can be set during any time.

XOUTEN Set this bit to enable the P2.1 pin as crystal oscillator XOUT. The crystal oscillator is considered as an analog peripheral. The corresponding IOCFGP2.1 ANEN should also be set. Default is disabled.

GPIOEN P2.1 GPIO function enable bit. Set this bit to enable P2.1's GPIO function. Default is disabled.

MFCFGP2.2 (0xA072) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**TXD1EN**	**GPIOEN**

P2.2 is shared with UART1 TXD1. Only one bit can be set during any time.

TXD1EN Set this bit to enable the P2.2 pin as TXD1. Default is disabled.

GPIOEN P2.2 GPIO function enable bit. Set this bit to enable P2.2's GPIO function. Default is disabled.

MFCFGP2.3 (0xA073) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**RXD1EN**	**GPIOEN**

P2.3 is shared with UART1 RXD1. Only one bit can be set during any time.

RXD1EN Set this bit to enable the P2.3 pin as RXD1. Default is disabled.

GPIOEN P2.3 GPIO function enable bit. Set this bit to enable P2.3's GPIO function. Default is disabled.

MFCFGP2.4 (0xA074) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**CMPDEN**	**T2EXEN**	**GPIOEN**

P2.4 is shared with Timer 2 T2EX signal.. Only one bit can be set during any time.

T2EXEN Set this bit to enable the P2.4 pin as T2EX. Default is disabled.

GPIOEN P2.4 GPIO function enable bit. Set this bit to enable P2.4's GPIO function. Default is disabled.

CMPDEN Set this bit to enable P2.4 to be used as the input of comparator D. The ANEN bit of P2.4 pin control register must also be enabled.

MFCFGP2.5 (0xA075) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**CMPCEN**	**T2EN**	**GPIOEN**

P2.5 is shared with Timer 2 T2 signal.. Only one bit can be set during any time.

T2EN Set this bit to enable the P2.5 pin as T2. Default is disabled.

GPIOEN P2.5 GPIO function enable bit. Set this bit to enable P2.5's GPIO function. Default is disabled.

CMPCEN Set this bit to enable P2.5 to be used as the input of comparator C. The ANEN bit of P2.5 pin control register must also be enabled.

MFCFGP2.6 (0xA076) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**CMPBEN**	-	**GPIOEN**

P2.6 is shared with Timer 1 T1 signal.. Only one bit can be set during any time.

~~T1EN Set this bit to enable the P2.6 pin as T1. Default is disabled.~~

GPIOEN P2.6 GPIO function enable bit. Set this bit to enable P2.6's GPIO function. Default is disabled.

CMPBEN Set this bit to enable P2.6 to be used as the input of comparator B. The ANEN bit of P2.6 pin control register must also be enabled.

MFCFGP2.7 (0xA077) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	**CMPAEN**	-	**GPIOEN**

P2.5 is shared with Timer 2 T2 signal.. Only one bit can be set during any time.

~~T0EN Set this bit to enable the P2.7 pin as T1. Default is disabled.~~

GPIOEN P2.7 GPIO function enable bit. Set this bit to enable P2.7's GPIO function. Default is disabled.

CMPAEN Set this bit to enable P2.7 to be used as the input of comparator A. The ANEN bit of P2.6 pin control register must also be enabled.

MFCFGP3.0 (0xA078) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**CEX2EN**	**GPIOEN**

P3.0 is shared with PCA CEX0.. Only one bit can be set during any time.

CEX2EN Set this bit to enable the P3.0 pin as CEX2 for PCA peripheral. Default is disabled.

GPIOEN P3.0 GPIO function enable bit. Set this bit to enable P3.0's GPIO function. Default is disabled.

MFCFGP3.1 (0xA079) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**CEX3EN**	**GPIOEN**

P3.1 is shared with PCA CEX0.. Only one bit can be set during any time.

CEX3EN Set this bit to enable the P3.1 pin as CEX3 for PCA peripheral. Default is disabled.

GPIOEN P3.1 GPIO function enable bit. Set this bit to enable P3.1's GPIO function. Default is disabled.

MFCFGP3.2 (0xA07A) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**CEX4EN**	**GPIOEN**

P3.2 is shared with PCA CEX2.. Only one bit can be set during any time.

CEX4EN Set this bit to enable the P3.1 pin as CEX4 for PCA peripheral. Default is disabled.

GPIOEN P3.2 GPIO function enable bit. Set this bit to enable P3.2's GPIO function. Default is disabled.

MFCFGP3.3 (0xA07B) W (0x00)

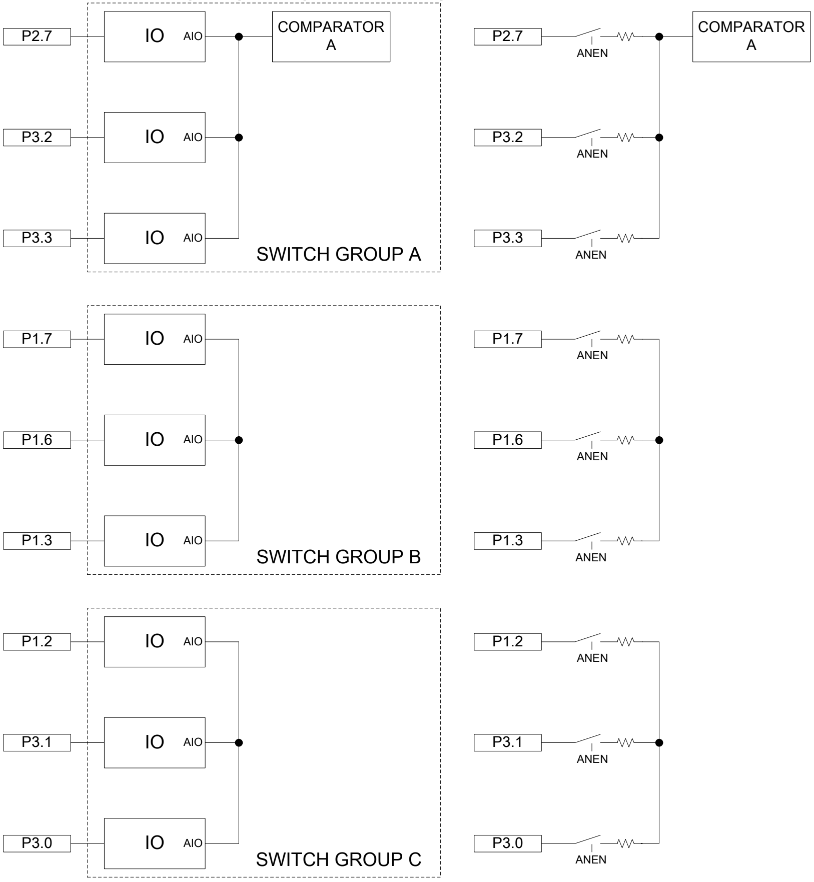
	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	-	-	-	-	-	-	**CEX5EN**	**GPIOEN**

P3.3 is shared with PCA CEX3.. Only one bit can be set during any time.

CEX5EN Set this bit to enable the P3.1 pin as CEX5 for PCA peripheral. Default is disabled.

GPIOEN P3.3 GPIO function enable bit. Set this bit to enable P3.3's GPIO function. Default is disabled.

Using the analog switch within the PIN IO circuits, it is possible to form analog multiplexer function. In CS8953, there are three groups of pins that implement this function, and are shown in the following diagram



The right is the equivalent circuit. The resistor is the equivalent resistor of the analog switch which is typically less than 1K Ohm. Using ANEN control, analog multiplexer can be achieved. The input/output range of the analog signal is limited to 0 – VDD. And note only switch group C is available in 28-pin package.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

SYMBOL	PARAMETER	RRATING	UNIT
VDD	Positive Power Supply	5.5	V
T_A	Ambient Operating Temperature	-40 to 85	°C
T_{STG}	Storage Temperature	-65 to 150	°C

Recommend Operating Condition

SYMBOL	PARAMETER	RRATING	UNIT
VDD	Positive Power Supply	3.0 to 5.5	V
T_A	Ambient Operating Temperature	-40 to 85	°C

DC/AC Electrical Characteristics VDD = 3.0V to 5.5V, T_A = -40°C to 85°C)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT	NOTE
Power Supply Current						
IDD, normal XOSC	Normal mode Supply Current using XOSC as clock source	-	15	35	mA	1
IDD, normal IOSC	Normal mode Supply Current using IOSC as clock source	-	12	25	mA	1
IDD, normal RTC	Normal mode Supply Current using RTC as clock source	-	4	10	mA	1
IDD, normal	Normal mode IDD versis					
IDD, PMM IOSC	PMM mode Supply Current using IOSC as clock source	-	8	20	mA	1
IDD, PMM RTC	PMM mode Supply Current using RTC as clock source	-	2	8	mA	1
IDD, idle IOSC	Idle mode Supply Current using IOSC as clock source	-	4	10	mA	1
IDD, idle RTC	Normal mode Supply Current using RTC as clock source	0.5	1	5	mA	1
IDD, stop 5.0V	Stop mode Supply Current	200	400	600	uA	1
IDD, stop 3.3V	Stop mode Supply Current	200	250	400	uA	1
IO Characteristics						
V_{IH}	Input High Level	0.7VDD	-	VDD+0.5	V	2
V_{IL}	Input Low Level	-0.5	-	0.3VDD	V	2
V_{TH}	Input Threshold	-	0.5VDD	-	V	
V_{INHYS}	Input Hysteresis	0.05	0.1	0.25	V	2
V_{OH}	Output High Level @ 1mA	VDD – 0.5	-	VDD	V	2
	Output High Level @ 2mA	VDD – 0.6	-	VDD	V	2
	Output High Level @ 4mA	VDD – 0.8	-	VDD	V	2
V_{OL}	Output Low Level @ 4mA	0	-	0.4	V	2
	Output Low Level @ 8mA	0	-	0.5	V	2
	Output Low Level @12mA	0	-	0.8	V	2
$R_{PU, 5V}$	Input Pull Up Resistance	150K	300K	600K	Ohm	2
$R_{PD, 5V}$	Input Pull Down Resistance	30K	70K	150K	Ohm	2
$R_{PU, 3V}$	Input Pull Up Resistance	300K	600K	1200K	Ohm	2
$R_{PD, 3V}$	Input Pull Down Resistance	50K	125K	250K	Ohm	2
$V_{IH,RSTN}$	Input High Level, RSTN	2.0	-	2.5	V	3
$V_{IL,RSTN}$	Input Low Level, RSTN	-0.2	-	0.6	V	3
T_{RISE}	Output Rise Time	8	20	35	nsec	7

T_{FALL}	Output Fall Time	8	20	35	nsec	7
R_{ANSW}	Equivalent analog switch resistance	300	750	1.5K	Ohm	8
Internal 2.5V Regulator						
VDD25	Internal 2.5V regulator output	2.4	2.5	2.6	V	4
VDD25, 10mA	Internal 2.5V regulator output	2.3	2.5	2.6	V	5
VDD25, reset	Power on/off reset level	70	80	90	%	6

Note 1: Does not include load current and tested under NOP loop and all peripheral disabled.
Note 2: For Digital IO only and single IO is enabled without power line shift.
Note 3: For RSTN pin only
Note 4: Supply to internal digital and analog circuit only after trim.
Note 5: Supply to internal digital and analog circuit and an external 20mA load.
Note 6: This is measured as the percentage of steady state value of VDD25.
Note 7: This is measured with 20pF load and 20% to 80 %
Note 8: This is for analog multiplexer applications

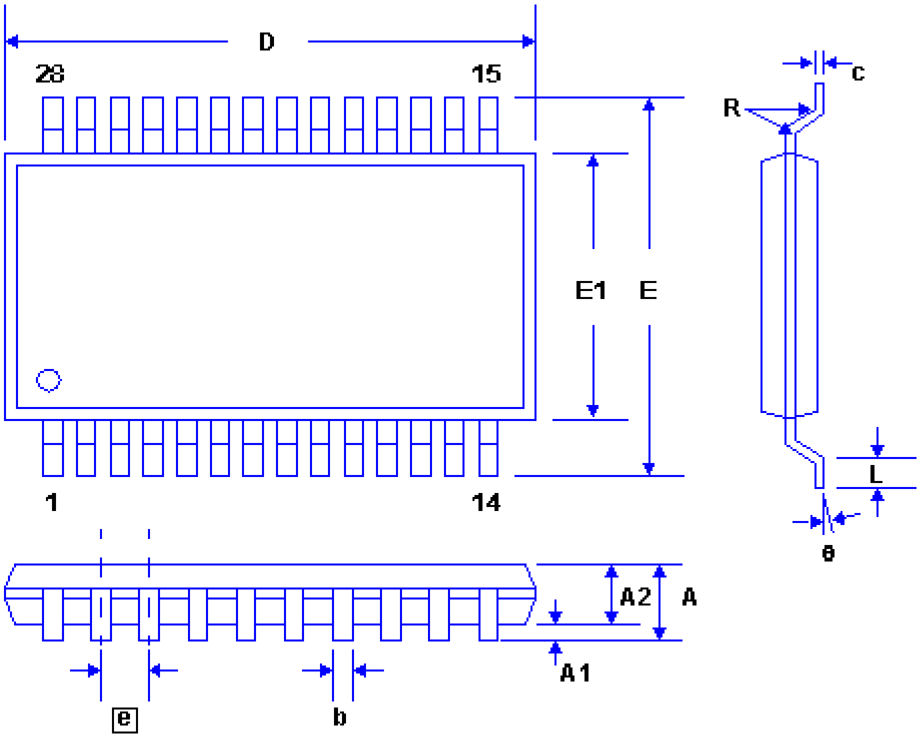
SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT	NOTE
Internal Oscillator						
	IOSC current consumption		500		uA	At 4MHz
F, IOSC	Factory Calibration, 25 °C	-	3.68 +/- 1%	-	MHz	
F, IOSC	Deviation over -40°C to -85°C	-5%	+/- 2%	+5%		
External Crystal Oscillator						
	XOSC current consumption	-	2.5	-	mA	Per-stage
F, XOSC	Crystal frequency	1	4	25	MHz	
T, XOSC	Clock period	1000	250	40	nsec	
D, XOSC	XOSC duty cycle	30	50	70	%	
D, XOSC	XOSC duty cycle	30	50	70	%	
Analog to Digital Converter						
	ADC power consumption	-	3mA	-	mA	At 4MHz
	ADC Resolution	-	10	-	Bits	-
T, ADCCLK	ADC Clock period	250	-	-	nsec	
T, CONV	Conversion Time	3.5	-	20	usec	1
TC, CONV	Conversion Cycles	-	14	-	Cycles	Single Conversion
Vin, FS	Full scale input range, 1X mode	0	-	VDD25	V	
Vin, FS	Full scale input range, 0.5X mode	0	-	2 x VDD25	V	
DNL	Differential Linearity	-1	0	+1	LSB	Monotonic
INL, mid-range	Absolute Linearity, VIN = 0.1 FS to 0.9FS	-2	0	+2	LSB	
INL, low and high range	Absolute Linearity, VIN = 0 to 0.1FS and VIN = 0.9FS to FS	-5	0	+5	LSB	
Gain Error, ADC	Gain error of ADC		4	10	LSB	
Offset Error, ADC	Offset error of ADC		4	8	LSB	
ATTN Gain	Attenuator gain	0.48	0.5	0.52		
R_{IN}	Input equivalent resistance	1MEG	-	-	Ohm	2
C_{IN}	Input equivalent capacitance	20	50	100	pF	2

Analog Comparator						
	Analog Comparator current consumption		300		uA	
Vin	Input range	0	-	VDD	V	
Vsen	Input Sensitivity	-10	+/-3	+10	mV	
Vth	Threshold programmable range	0	-	VDD25	V	
Vth, step	Threshold programmable steps			256		
T, CMP	Propagation Delay	100	150	250	nsec	3
C_{IN}	Input equivalent capacitance	20	50	100	pF	4
Low Voltage Detect						
	LVD current consumption	-	125	-	uA	

Note 1: The maximum is limited by the S/H leakage.
Note 2: The input to ADC is pure capacitor to S/H.
Note 3: With +/- 50mV overdriven input
Note 4: Pure capacitance load

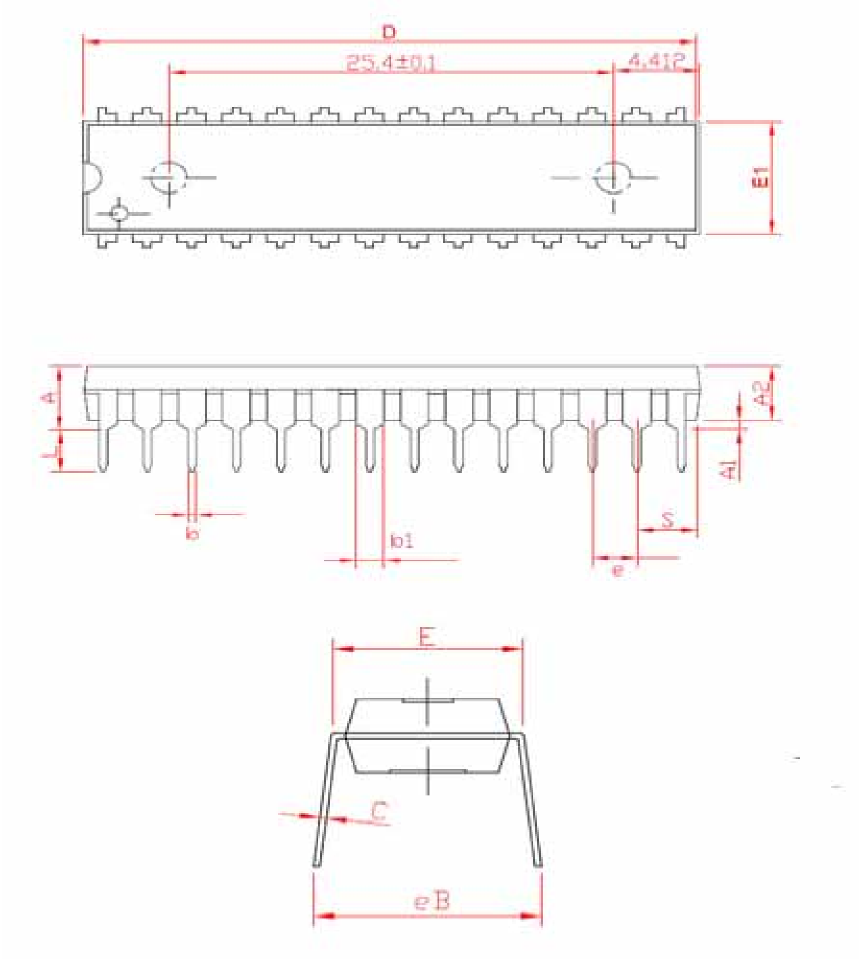
PACKAGE OUTLINE

SSOP-28



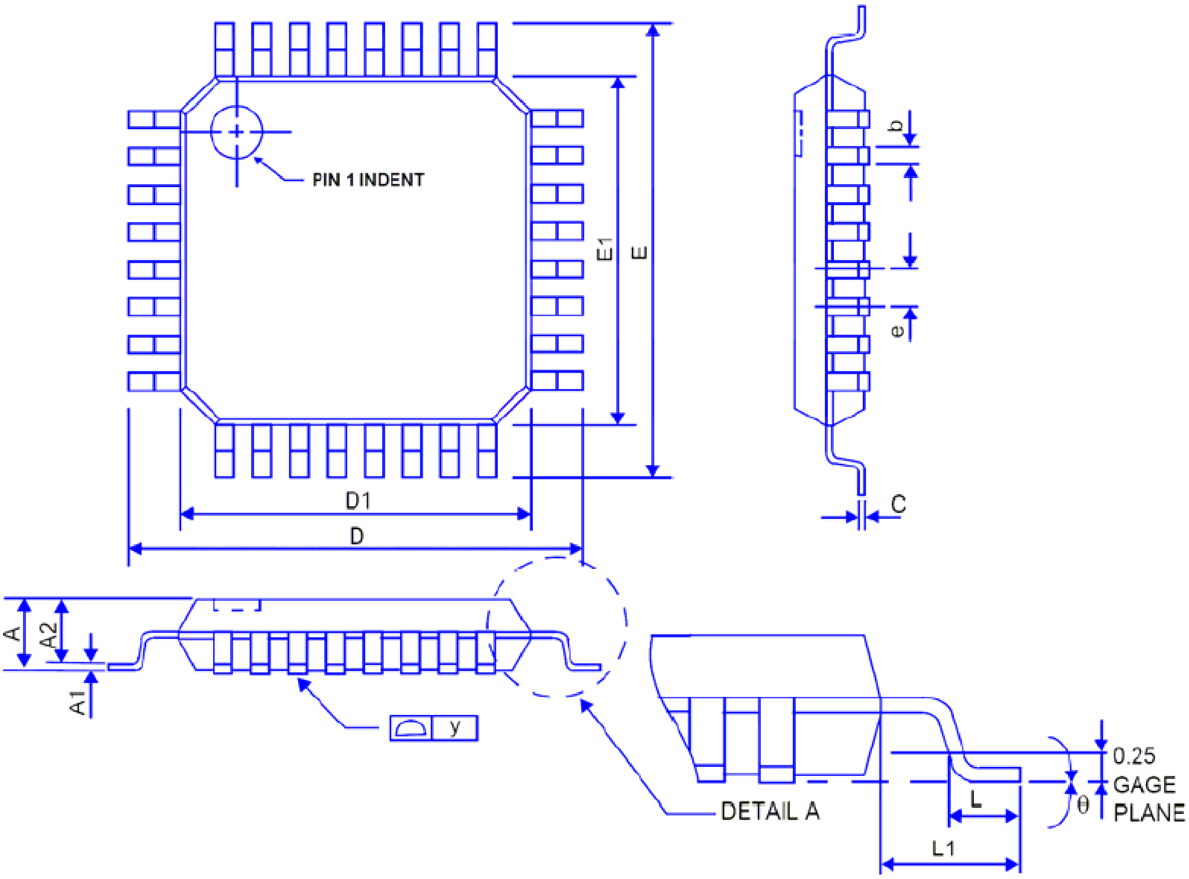
Symbol	Dimension in Inches		
	Min	Nom	Max
A	0.053		0.078
A1	0.002		0.010
A2	0.04	-	0.072
b	0.008		0.014
c	0.003		0.009
D	0.395		0.413
E	0.301		0.322
E1	0.196		0.220
e	0.025 BSC		
L	0.021		0.037
θ	0º		8º

Skinny DIP-28



Symbol	Dimensions in millimeters			Dimensions in inches		
	MIN	NOM	MAX	MIN	NOM	MAX
A	3.67	3.69	3.70	0.144	0.145	0.146
A1	0.33	0.39	0.46	0.013	0.015	0.018
A2	3.27	3.30	3.31	0.129	0.130	0.131
b	-	0.46	-	-	0.018	-
b1	1.52	1.55	1.61	0.060	0.061	0.063
C	0.24	-	0.26	0.009	-	0.01
D	34.96	35.18	35.30	1.376	1.385	1.389
S	3.34	3.38	3.40	0.131	0.133	0.134
E	8.12	8.13	8.15	0.319	0.320	0.321
E1	7.32	7.38	7.40	0.288	0.290	0.291
e	-	2.54	-	-	0.1	-
L	-	3.23	3.25	-	0.127	0.128
eB	8.4	8.9	9.4	0.330	0.350	0.370

LQFP-32



Symbol	Dimensions in millimeters			Dimensions in inches		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	1.60	-	-	0.063
A1	0.05	-	0.15	0.002	-	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	0.37	0.45	0.012	0.014	0.017
C	0.09	-	0.20	0.003	-	0.008
E	-	9.00	-	-	0.354	-
E1	-	7.00	-	-	0.276	-
D	-	9.00	-	-	0.354	-
D1	-	7.00	-	-	0.276	-
e	-	0.80	-	-	0.031	-
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	-	1.00	-	-	0.039	-
θ	0°	3.5°	7°	0°	3.5°	7°
y	0.0	-	0.10	0.000	-	0.004

ORDERING INFORMATION

PREMIX	PART NUMBER	GREEN PACKAGE TYPE	SUFFIX	REMARK
CS	8953	AG: LQFP-32 AR: SSOP-28 AN: Skinny DIP-28		