IS31CS8968A

Advanced 8051 MCU with 128K Flash and 6K SRAM

GENERAL DESCRIPTION

IS31CS8968A is a general-purpose microcontroller with extensive peripherals suitable for a wide range of applications. The CPU is based on an enhanced 1-cycle 8051 core equivalent to ten times the speed of a conventional 12-T 8051. The total on-chip memory include 6KB SRAM and a total of 128KB 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, 24-bit T3 timer, and a 30-bit watchdog timer. Embedded in the CPU core are also a full-duplex UART ports, one I²C master/slave and one I²C pure slave controllers, up to 42 GPIO pins.

The flexibility in clock setting includes an on-chip precision oscillator with the accuracy deviation of +/-2%, or a slow power internal 100KHz oscillator, and a external 4MHz to 24MHz crystal oscillator, or an ultra low power precision real time clock (RTC). Unused clock sources can be disabled or used as GPIO pins for system optimization. The clock selections are combined with flexible power management schemes, including PMM, IDLE, and STOP, SLEEP modes to balance CPU speed and power consumption.

Other on-chip peripherals include one SPI control interface, one I²C master/slave and one I²C pure slave controllers, as well as a Programmable Counter Array (PCA) with 6 channels of Capture/Compare/PWM modules.

Analog peripherals include a high performance 12-Bit Analog to Digital Converter (ADC) with 4usec conversion time, 4 analog comparators with programmable threshold levels, and a 10-bit Voltage Output Digital to Analog Converter (VDAC).

A built-in programmable CEC (Consumer Electronics Control) Controller allows users to control all of the various audiovisual products in a user's environment easily.

IS31CS8968A also includes a CAN bus controller compliant to CAN2.0A/B and CAN OPEN specifications. It supports standard 11-bit and 29-bit identification modes. CAN receive path contains 4 acceptance filter for ID filtering and a parameterized FIFO (shared with the CPU SRAM storage) for received message buffering. The transmit path includes a dedicated transmit message buffer and also three dispatch message buffers. The dispatch messages can be used for periodic messaging without CPU intervention with programmable durations and priorities. The CAN controller supports normal operation mode with auto-recovery that compliant to CAN bus error handling standard. It also supports listen-only mode, and test mode which allows external loop-back test of the CAN functions.

IS31CS8968A 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 allows IS31CS8968A reliable operations under harsh environment. The code security is extremely secure based on sophisticated writer commands and ISP commands. The on-chip break point processor also allows easy debug environments that can be integrated with ISP.

Intended application fields of IS31CS8968A include LCD/PDP TV, LCD Monitor, automotive AV system, home appliance, and other embedded applications.

FEATURES

CPU and Memory

- 1-Cycle 8051 CPU core up to 24MHz operation frequency (One Wait State); 16MHz (Zero Wait State)
- 16-bit Timers T0/T1/T2 and 24-bit Timer T3
- Programmable 30-bit Watch Dog Timer
- Integrated break point controller for software debugger
- Software debugging port through I²C slave
- One full-duplex UART0 port
- Up to 10 external interrupts shared with GPIO pins
- Power saving mode – PMM, IDLE, STOP, and SLEEP modes
- 256B Internal SRAM and 5888B XSRAM
- 128KB Flash Memory and 256B Information Block
  - Configured to be shared by ISP code, program code, and data flash
  - Code security and content loss protection
  - Endurance: 100K cycles

Clock Sources

- Internal oscillator at 16MHz of +/- 2% accuracy
- Internal low power OSC of 100KHz
- Crystal oscillator 4MHz – 24MHz
- RTC - 32KHz of low power consumption

Digital Peripherals

- 16-bit PCA and 6 channel CCP modules
  - Capture/Compare/Timer Mode
  - 8-Bit and 16-bit PWM Mode
  - 8-Bit Windowed PWM Mode
- One 16-bit PWM output with programmable base frequency and duty cycles
- Two I²C Slave Controllers
- One Master/Slave SPI Controller
- One full-duplex LIN-capable EUART
- CEC Controller
- CAN Controller

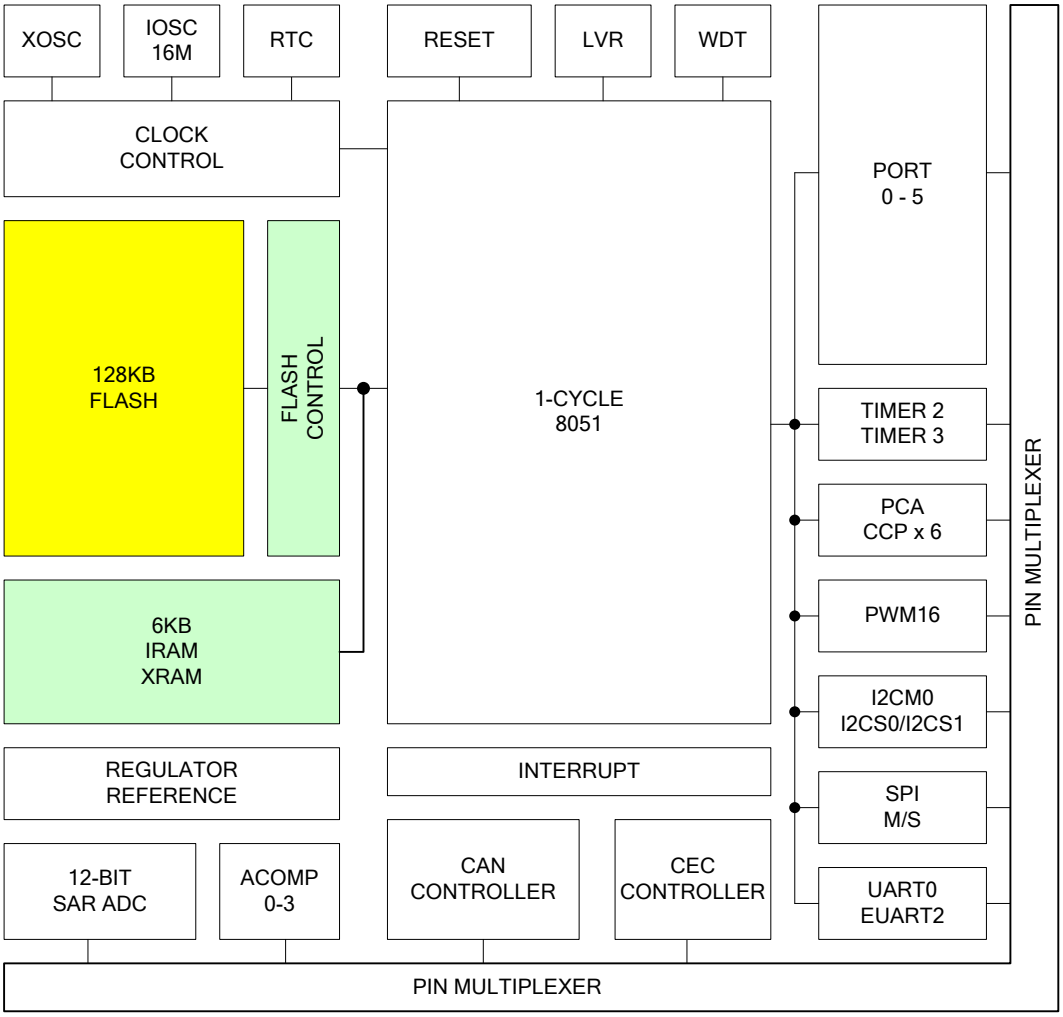
Analog Peripherals

- 12-Bit monotonic SAR ADC
  - 4 usec conversion time
  - 4 intrinsic time-multiplexed channels and dedicated result registers. 2 of which have sample and hold.
  - 10 inputs multiplexed with GPIO
  - On-chip temperature sensor
- 4 analog comparators
  - Two 8-bit programmable threshold or external threshold
- 10-bit Voltage Output DAC
  - 2mA full scale
  - Configurable Sink/Source
- Power on reset
- Low voltage detection on supply voltage

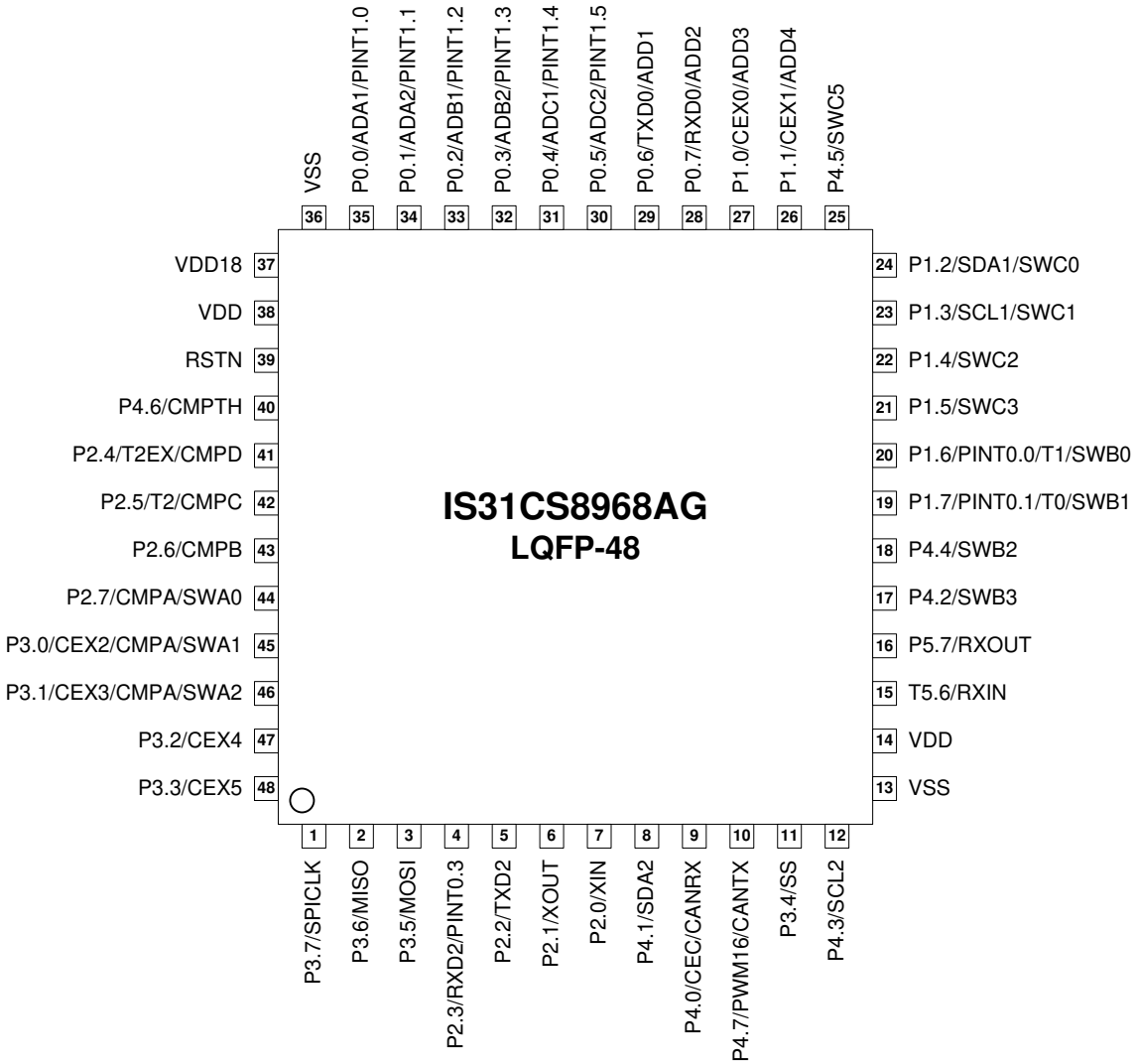
Miscellaneous

- Up to 42 GPIO pins
- 2.5V to 5.5V single supply with on-chip regulator or 1.8V direct single supply
- Low power standby (< 20uA) in SLEEP mode
- Operating temperature -40℃ – 85℃
- LQFP-48 package and RoHS compliant

BLOCK DIAGRAM



PIN CONNECTION



PIN DESCRIPTIONS

PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
P3.7	I/O	1	**Port 3.7**
			8051 P3.7 GPIO.
			SPICLK
			It behaves as the Master clock output (Master Mode), or Slave clock input (Slave Mode) of the on-chip SPI interface.
P3.6	I/O	2	**Port 3.6 GPIO**
			8051 P3.6 GPIO.
			MISO
			It behaves Master data Input of the on-chip SPI interface (Master Mode), or Slave data Output of the SPI interface (Slave Mode).
P3.5	I/O	3	**Port 3.5 GPIO**
			8051 P3.5 GPIO.
			MOSI
			It behaves Master data Output of the on-chip SPI interface (Master Mode), or Slave data Input of the SPI interface (Slave Mode).
P2.3	I/O,A	4	**Port 2.3 GPIO**
			8051 P2.3 GPIO.
			PINT0.3
			This pin also can be enabled as the expanded INT0 interrupt --- PINT0.3. Any toggle triggering interrupt condition can be set into the PINT0FG.3 if the PINT0EN.3 has been pre-set directly. No more I/O select bit is needed for PINT0.3 related pre-setting in MFCFGP2.3 register.
			RXD2
			This pin also can be configured as RXD of EUART2.
			VDAC
			10-bit DAC output: The 10-bit voltage output DAC driver to the port IO cell's analog output switch. To enable this function, the ANEN of IOCFGP2.3 must be turned on.
P2.2	I/O	5	**Port 2.2 GPIO**
			8051 P2.2 GPIO.
			TXD2
			This pin also can be configured as TXD of UART 2.
P2.1	I/O, A	6	**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.
P2.0	I/O, A	7	**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.
P4.1	I/OD	8	**Port 4.1 GPIO**

PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
			8051 P4.1 GPIO. The output is open-drain only, an external pull-up resistor is necessary to connect to VDD.
			SDA2
			This pin also can be configured as the SDA signal of the 2nd I²C slave controller. In this operation mode, this pin should also be configured as bi-directional I/O with open-drain output. [i.e., IOCFGP4.1 = 0b1000001x]
P4.0	I/OD	9	**8051 P4.0 GPIO**
			8051 P4.0 GPIO. The output is open-drain only, an external pull-up resistor is necessary to connect to VDD.
			PINT0.2
			This pin also can be enabled as the expanded INT0 interrupt --- PINT0.2. The function is dedicated to CAN and CEC remote wakeup only. No more I/O select bit is needed for PINT0.2 related pre-setting in MFCFGP4.0 register.
			CEC
			CEC Controller input/output pin. The falling edge can wake up this device from STOP mode. In this operation mode, this pin should also be configured as bi-directional I/O with open-drain output [i.e., IOCFGP4.0 = 0b1000001x].
			CANRX
			This is receive input of CAN Controller. Connect to external CAN transceiver RX output.
P4.7	I/O	10	**Port 4.7 GPIO**
			8051 P4.7 GPIO.
			PWM16
			16-bit PWM channel output pin.
			CANTX
			This is transmit output of CAN Controller. Connect to external CAN transceiver TX input.
P3.4	I/O	11	**Port 3.4 GPIO**
			8051 P3.4 GPIO.
			SS
			This pin also can be configured as slave chip enable pin for the on-chip SPI interface under slave mode. This allows multiple slaves to be connected to a SPI master through this chip select functions. The host SPI master asserts SS to low to enable the SPI slave.
P4.3	I/OD	12	**Port 4.3 GPIO**
			8051 P4.3 GPIO. The output is open-drain only, an external pull-up resistor is necessary to connect to VDD.
			SCL2
			This pin also can be configured as the SCL signal of the 2nd I²C slave controller. This pin should be configured as input only [i.e., IOCFGP4.3 = 0b1000000x].
VSSIO	P	13	Ground Voltage. 0V
VDD	P	14	**Supply Voltage. 2.5V ~ 5.5V**
			A good decoupling capacitor between VDD and VSS pins is critical for good performance.
P5.6	I/O,A	15	**Port 5.6 GPIO**
			8051 P5.6 GPIO
			RXIN
			Crystal IN for RTC oscillator. The control of the RXIN of RTC is done by setting of ANEN of IOCFGP5.6.
P5.7	I/O,A	16	**Port 5.7 GPIO**

PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
			8051 P5.7 GPIO
			RXOUT
			Crystal OUT for RTC oscillator. The control of the RXOUT of RTC is done by setting of ANEN of IOCFGP5.7.
P4.2	I/O,A	17	**Port 4.2 GPIO**
			8051 P4.2 GPIO.
			SWB3
			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 IOCFGP4.2.
P4.4	I/O,A	18	**Port 4.4 GPIO**
			8051 P4.4 GPIO.
			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 IOCFGP4.4.
P1.7	I/O,A	19	**Port 1.7 GPIO**
			8051 P1.7 GPIO.
			PINT0.1
			This pin also can be configured as the expanded INT0 interrupt.
			T0 Timer 0 External Input
			This pin also can be configured as Timer 0 external input either as clock or gate.
			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 IOCFGP1.7.
P1.6	I/O,A	20	**Port 1.6 GPIO**
			8051 P1.6 GPIO.
			PINT0.0
			This pin also can be configured as the expanded INT0 interrupt.
			T1 Timer 1 External Input
			This pin also can be configured as Timer 1 external input either as clock or gate.
			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 IOCFGP1.6.
P1.5	I/O,A	21	**Port 1.5 GPIO**
			8051 P1.5 GPIO.
			SWC3
			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 IOCFGP1.5.
P1.4	I/O,A	22	**Port 1.4 GPIO**
			8051 P1.4 GPIO.
			SWC2
			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 IOCFGP1.4.
P1.3	I/OD	23	**Port 1.3 GPIO**
			8051 P1.3 GPIO. The output is open-drain only, an external pull-up resistor is necessary to connect to VDD.
			SCL

PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
			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 [i.e., IOCFGP1.3 = 0b0000001x]. In I²C slave, this pin should be configured as input only [i.e., IOCFGP1.3 = 0b1000000x].
P1.2	I/OD	24	**Port 1.2 GPIO**
			8051 P1.2 GPIO. The output is open-drain only, an external pull-up resistor is necessary to connect to VDD.
			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 [i.e., IOCFGP1.2 = 0b1000001x].
P4.5	I/O,A	25	**Port 4.5 GPIO**
			8051 P4.5 GPIO.
			SWC5
			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 IOCFGP4.5.
P1.1	I/O,A	26	**Port 1.1 GPIO**
			8051 P1.1 GPIO.
			CEX1 of 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.
P1.0	I/O,A	27	**Port 1.0 GPIO**
			8051 P1.0 GPIO.
			CEX0 of 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.
P0.7	I/O,A	28	**Port 0.7 GPIO**
			8051 P0.7 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.
P0.6	I/O,A	29	**Port 0.6 GPIO**
			8051 P0.6 GPIO.
			TXD0
			This pin also can be configured as TXD of UART 0.
			ADD1

PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
			This pin also can be configured as the input to the ADC channel D by setting ANEN of IOCFGP0.6 to 1. Only one of ADD1, ADD2, ADD3 and ADD4 can be enabled at any one time.
P0.5	I/O,A	30	**Port 0.5 GPIO**
			8051 P0.5 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.
P0.4	I/O,A	31	**Port 0.4 GPIO**
			8051 P0.4 GPIO.
			PINT1.4
			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 IOCFGP0.4 to 1. Only one of ADC1 and ADC2 can be enabled at any one time.
P0.3	I/O,A	32	**Port 0.3 GPIO**
			8051 P0.3 GPIO.
			PINT1.3
			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 IOCFGP0.3 to 1. Only one of ADB1 and ADB2 can be enabled at any one time.
P0.2	I/O,A	33	**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.
P0.1	I/O,A	34	**Port 0.1 GPIO**
			8051 P0.1 GPIO.
			PINT1.1
			This pin also can be configured as the expanded INT1 interrupt.
			ADA2
			This pin also can be configured as the input to the ADC channel A by setting ANEN of IOCFGP0.1 to 1. Only one of ADA1 and ADA2 can be enabled at any one time.
P0.0	I/O,A	35	**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.

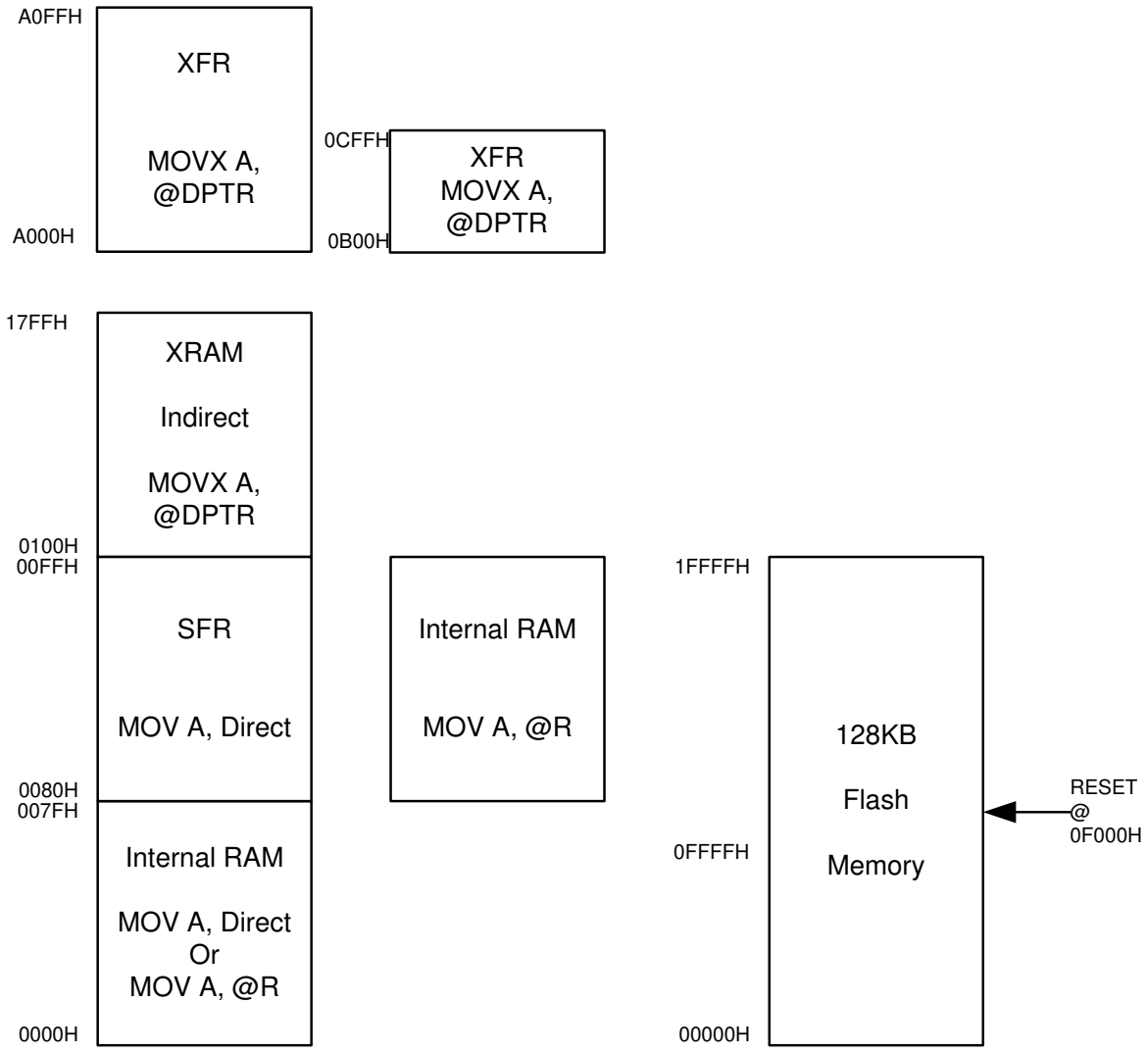
PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
VSS	P	36	VSS
VDD18	P	37	**Internal Regulator Output. 1.6V ~ 2.0V** Typical decoupling capacitors of 0.1uF and 4.7uF should be connected between VDD18 and VSS.
VDD	P	38	**Supply Voltage. 2.5V ~ 5.5V** A good decoupling capacitor between VDD and VSS pins is critical for good performance.
RSTN	I	39	**Reset Low Active.** Typically connect a resistor to VDD18 and a capacitor to VSS. Low asserted and threshold at 0.5*VDD18. When forced low, the chip enters into reset condition. This pin should not be connected to any level above VDD18.
P4.6	I/O	40	**Port 4.6 GPIO** 8051 P4.6 GPIO. **CMPTH** This pin can be configured as comparator external threshold by setting ANEN of IOCFGP4.6 to 1.
P2.4	I/O,A	41	**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. **CMPD** This pin can be configured as the comparator D input by setting ANEN of IOCFGP2.4 to 1.
P2.5	I/O,A	42	**Port 2.5 GPIO** 8051 P2.5 GPIO. **T2 Timer 2 Clock Input** This pin also can be configured as Timer 2 clock input. **CMPC** This pin can be configured as the comparator C input by setting ANEN of IOCFGP2.5 to 1.
P2.6	I/O,A	43	**Port 2.6 GPIO** 8051 P2.6 GPIO. **CMPB** This pin can be configured as the comparator B input by setting ANEN of IOCFGP2.6 to 1.
P2.7	I/O,A	44	**Port 2.7 GPIO** 8051 P2.7 GPIO. **CMPA** This pin can be configured as the comparator An input by setting ANEN of IOCFGP2.7 to 1. **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	45	**Port 3.0 GPIO**

PIN NAME	TYPE	PIN #	PIN FUNCTION DESCRIPTION
			8051 P3.0 GPIO.
			CEX2 of 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.
			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 IOCFGP3.0.
			CMPA
			This pin can be configured as the comparator An input by setting ANEN of IOCFGP3.0 to 1.
P3.1	I/O,A	46	**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.
			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 IOCFGP3.1.
			CMPA
			This pin can be configured as the comparator An input by setting ANEN of IOCFGP3.1 to 1.
P3.2	I/O	47	**Port 3.2 GPIO**
			8051 P3.2 GPIO
			CEX4 of 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.
			VDAC
			10-bit DAC output: The 10-bit voltage output DAC driver to the port IO cell's analog output switch. To enable this function, the ANEN of IOCFGP3.2 must be turned on.
P3.3	I/O	48	**Port 3.3 GPIO**
			8051 P3.3 GPIO
			CEX5 of 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.

Note: "P" denotes power supply pins
"G" denotes ground pins. All VSS pins are internally shorted resistively.
"O", "IO", "A" denotes output only, input/output, and analog types.
"OD" means the output is capable of open-drain drive without active high driving.
"PU" and "PD" denotes pins with internal pull-up or pull-down.

MEMORY MAP

There are total 256 bytes internal RAM in IS31CS8968A, the same as standard 8052. There are total 5888 bytes auxiliary RAM allocated in the 8051 extended RAM area 100h – 17FFh. Programs can use "MOVX" instruction to access the AUXRAM. The 128KB embedded flash occupies the program address space from 00000h – 1FFFFh, The CPU reset to address 0F000H. The memory map is shown in the following figure.



REGISTER MAP SFR(0x80 – 0xFF) and XFR (0xA000 – 0xA17F)

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 require 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	-	CLSR	CHSR	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0XE0	ACC	PCACON	CCAP3L	CCAP3H	CCAP4L	CCAP4H	CCAP5L	CCAP5H
0XD0	PSW	PCAMOD	CCAP0L	CCAP0H	CCAP1L	CCAP1H	CCAP2L	CCAP2H
0XC0	SCON1	-	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**
0XF8	EXIP	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
0XE8	EXIE	CH	MXAX	I2CSCON1	I2CSST1	I2CSADR1	I2CSDAT1	P4
0XD8	WDCON	CL	DPXR	I2CSCON2	I2CSST2	I2CSADR2	I2CSDAT2	P5
0XC8	T2CON	TB	RLDL	RLDH	TL2	TH2	ADCAVG	-
0XB8	IP	ADCCHSL	ADCAL	ADCAH	ADCBL	ADCBH	ADCCL	ADCCH
0XA8	IE	ADCCFG	ADCDL	ADCDH	CECCTL	CECSTS	PINT1EN	PINT1FG
0X98	SCON0	SBUF0	CECTXD	ESP	CECRDX	ACON	-	WKMASK
0X88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CKSEL

The register names in gray are for those external peripherals not included in IS31CS8968A and are reserved. The register names in green and red are for those registers not standard in 8051/8052/80390.

	0	1	2	3	4	5	6	7
A000	REGTRM	IOSCITRM	IOSCVTRM	-	-	-	-	XOSCCFG
A010	LVDCFG	LVDTHD	FLSHADM	INTPCT1	INTPCT2	-	-	-
A020	FLSHCMD	FLSHDAT	FLSHADH	FLSHADL	ISPCLKF	CNTPCTL	CNTPCTH	-
A030	CMPCFGAB	CMPCFGCD	CMPVTH0	CMPVTH1	-	-	DACL	DACH
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**
A008	RTCSCND0	RTCSCND1	RTCSCND2	RTCSCND3	RTCCNTR0	RTCCNTR1	RTCCMD	-
A018	-	-	-	-	-	-	-	-
A028	PWM16CFG	PWMPERDH	PWMPERDL	PWMDUTYH	PWMDUTYL	-	-	-
A038	-	-	-	-	-	-	-	-
A048	IOCFGP1.0	IOCFGP1.1	IOCFGP1.2	IOCFGP1.3	IOCFGP1.4	IOCFGP1.5	IOCFGP1.6	IOCFGP1.7
A058	MFCFGP1.0	MFCFGP1.1	MFCFGP1.2	MFCFGP1.3	MFCFGP1.4	MFCFGP1.5	MFCFGP1.6	MFCFGP1.7

A068	IOCFGP3.0	IOCFGP3.1	IOCFGP3.2	IOCFGP3.3	IOCFGP3.4	IOCFGP3.5	IOCFGP3.6	IOCFGP3.7
A078	MFCFGP3.0	MFCFGP3.1	MFCFGP3.2	MFCFGP3.3	MFCFGP3.4	MFCFGP3.5	MFCFGP3.6	MFCFGP3.7

	0	1	2	3	4	5	6	7
A080	CECCFG	CECADR	CECCKPRS	CECBPRD	CEC0LDTY	CEC1LDTY	CECSTPRD	CECSTLDTY
A090	LINCTRL	LINCNTRH	LINCNTRL	LINSBRH	LINSBRL	LININT	LININTEN	CECBFCTRL
A0A0	-	-	-	-	-	PCACPS	CLRLD	CHRLD
A0B0	-	-	-	-	-	-	-	-
A0C0	IOCFGP4.0	IOCFGP4.1	IOCFGP4.2	IOCFGP4.3	IOCFGP4.4	IOCFGP4.5	IOCFGP4.6	IOCFGP4.7
A0D0	MFCFGP4.0	MFCFGP4.1	MFCFGP4.2	MFCFGP4.3	MFCFGP4.4	MFCFGP4.5	MFCFGP4.6	MFCFGP4.7
A0E0	BPINTF	BPINTE	BPINTC	BPCTRL	PC5AL	PC5AH	PC5AT	-
A0F0	PC1AL	PC1AH	PC1AT		PC2AL	PC2AH	PC2AT	-
	8	**9**	**A**	**B**	**C**	**D**	**E**	**F**
A088	CECSTRBPNT	CECNTYPRD	CECNTYLDTY	CECSTXPNT	CECMINEVDUR	CECTMOUTLMT	CECRXFREE	CECSTLDT
A098	DBPCIDL	DBPCIDH	DBPCIDT	DBPCNXL	DBPCNXH	DBPCNXT	-	-
A0A8	-	-	-	-	-	-	IOCFGP5.6	IOCFGP5.7
A0B8	-	-	-	-	-	-	MFCFGP5.6	MFCFGP5.7
A0C8	-	-	-	-	-	-	-	-
A0D8	-	-	-	-	-	-	-	-
A0E8	PC6AL	PC6AH	PC6AT	-	PC7AL	PC7AH	PC7AT	SI2CDBGID
A0F8	PC3AL	PC3AH	PC3AT	-	PC4AL	PC4AH	PC4AT	STEPCTRL

	0	1	2	3	4	5	6	7
A100	TMR3CFG1	TMR3CFG2	TMR3L	TMR3M	TMR3H	TMR3RLL	TMR3RLM	TMR3RLH
A110	TMR3SR0L	TMR3SR0M	TMR3SR0H	TMR3SR1L	TMR3SR1M	TMR3SR1H	-	TMR3SSC
A120	-	-	-	-	-	-	-	-
A130	-	-	-	-	-	-	-	-
A140	-	-	-	-	-	-	-	-
A150	-	-	-	-	-	-	-	-
A160	-	-	-	-	-	-	-	-
AIE0	DBPC_L_ID	DBPC_H_ID	DBPC_T_ID	NXPC_L_ID	NXPC_H_ID	NXPC_T_ID	-	-
	8	**9**	**A**	**B**	**C**	**D**	**E**	**F**
A108	TMR3CMPL	TMR3CMPM	TMR3CMPH	-	-	-	-	-
A118	-	-	-	-	-	-	-	-
A128	-	-	-	-	-	-	-	-
A138	-	-	-	-	-	-	-	-
A148	-	-	-	-	-	-	-	-
A158	-	-	-	-	-	-	-	-
A168	-	-	-	-	-	-	-	-
A178	-	-	-	-	-	-	-	-

CAN XFR Table

	0	1	2	3	4	5	6	7
AB00	CAN_MODE	CAN_COMAND	CAN_STATUS	CAN_INTF	CAN_INTE	CAN_RX_Start_Addr	CAN_RX_Stop_Addr-	CAN_Current_WR_AddrL
AB10	CAN_Arb_Lostt_Capture	CAN_Error_Code	CAN_Warning_Limit	CAN_RX_Err_CNT	CAN_TX_Err_CNT	-	-	CAN_TX_Info
AB20	CAN_TX_DATA5	CAN_TX_DATA6	CAN_TX_DATA7	CAN_TX_DATA8	CAN_Disp1_Time3	CAN_Disp1_Time2	CAN_Disp1_Time1	CAN_Disp1_Time0
AB30	CAN_Filt1_Acc3	CAN_Filt1_Mask3	CAN_Filt1_Acc2	CAN_Filt1_Mask2	CAN_Filt1_Acc1	CAN_Filt1_Mask1	CAN_Filt1_Acc0	CAN_Filt1_Mask0
AB40	CAN_Filt3_Acc3	CAN_Filt3_Mask3	CAN_Filt3_Acc2	CAN_Filt3_Mask2	CAN_Filt3_Acc1	CAN_Filt3_Mask1	CAN_Filt3_Acc0	CAN_Filt3_Mask0
AC30	CAN_Disp1_Info	CAN_Disp1_ID3	CAN_Disp1_ID2	CAN_Disp1_ID1	CAN_Disp1_ID0	CAN_Disp1_Data1	CAN_Disp1_Data2	CAN_Disp1_Data3
AC40	CAN_Disp2_Info	CAN_Disp2_ID3	CAN_Disp2_ID2	CAN_Disp2_ID1	CAN_Disp2_ID0	CAN_Disp2_Data1	CAN_Disp2_Data2	CAN_Disp2_Data3
AC50	CAN_Disp3_Info	CAN_Disp3_ID3	CAN_Disp3_ID2	CAN_Disp3_ID1	CAN_Disp3_ID0	CAN_Disp3_Data1	CAN_Disp3_Data2	CAN_Disp3_Data3
AC90	-	CAN_Loop_Key	CAN_Loop_En	-	-	-	-	-
	8	**9**	**A**	**B**	**C**	**D**	**E**	**F**
AB08	CAN_Current_WR_AddrH	CAN_Timing0	CAN_Timing1	-	-	-	-	CAN_Filter_EN
AB18	CAN_TX_ID3	CAN_TX_ID2	CAN_TX_ID1	CAN_TX_IN0	CAN_TX_DATA1	CAN_TX_DATA2	CAN_TX_DATA3	CAN_TX_DATA4
AB28	CAN_Filt0_Acc3	CAN_Filt0_Mask3	CAN_Filt0_Acc2	CAN_Filt0_Mask2	CAN_Filt0_Acc1	CAN_Filt0_Mask1	CAN_Filt0_Acc0	CAN_Filt0_Mask0
AB38	CAN_Filt2_Acc3	CAN_Filt2_Mask3	CAN_Filt2_Acc2	CAN_Filt2_Mask2	CAN_Filt2_Acc1	CAN_Filt2_Mask1	CAN_Filt2_Acc0	CAN_Filt2_Mask0
AC28	CAN_Disp2_Time1	CAN_Disp2_Time0	-	-	-	-	-	-
AC38	CAN_Disp1_Data4	CAN_Disp1_Data5	CAN_Disp1_Data6	CAN_Disp1_Data7	CAN_Disp1_Data8	CAN_Disp1_CTRL/ Status	-	-
AC48	CAN_Disp2_Data4	CAN_Disp2_Data5	CAN_Disp2_Data6	CAN_Disp2_Data7	CAN_Disp2_Data8	CAN_Disp2_CTRL/ Status	-	-
AC58	CAN_Disp3_Data4	CAN_Disp3_Data5	CAN_Disp3_Data6	CAN_Disp3_Data7	CAN_Disp3_Data8	CAN_Disp3_CTRL/ Status	-	-

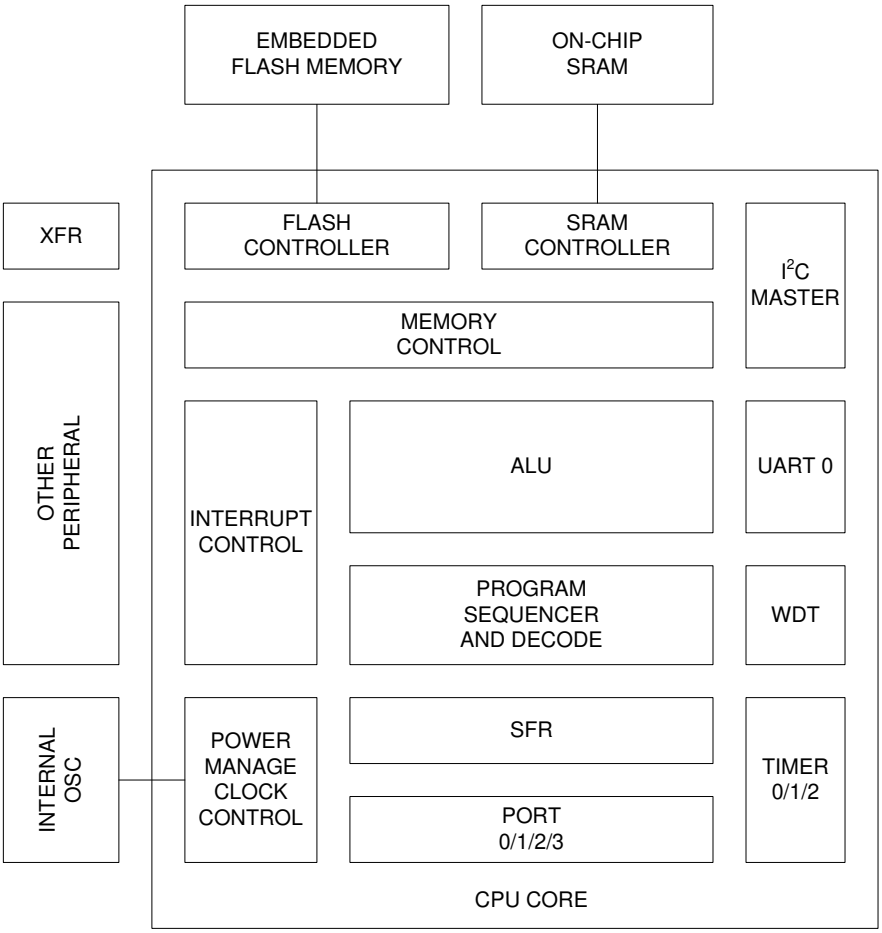
1. Enhanced 1-Cycle 8051 CPU

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, DPX1, 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 supports switchback features.
- Access Control of critical registers - TA, and TB registers
- Eight break pointers allows integration of common IDE

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.

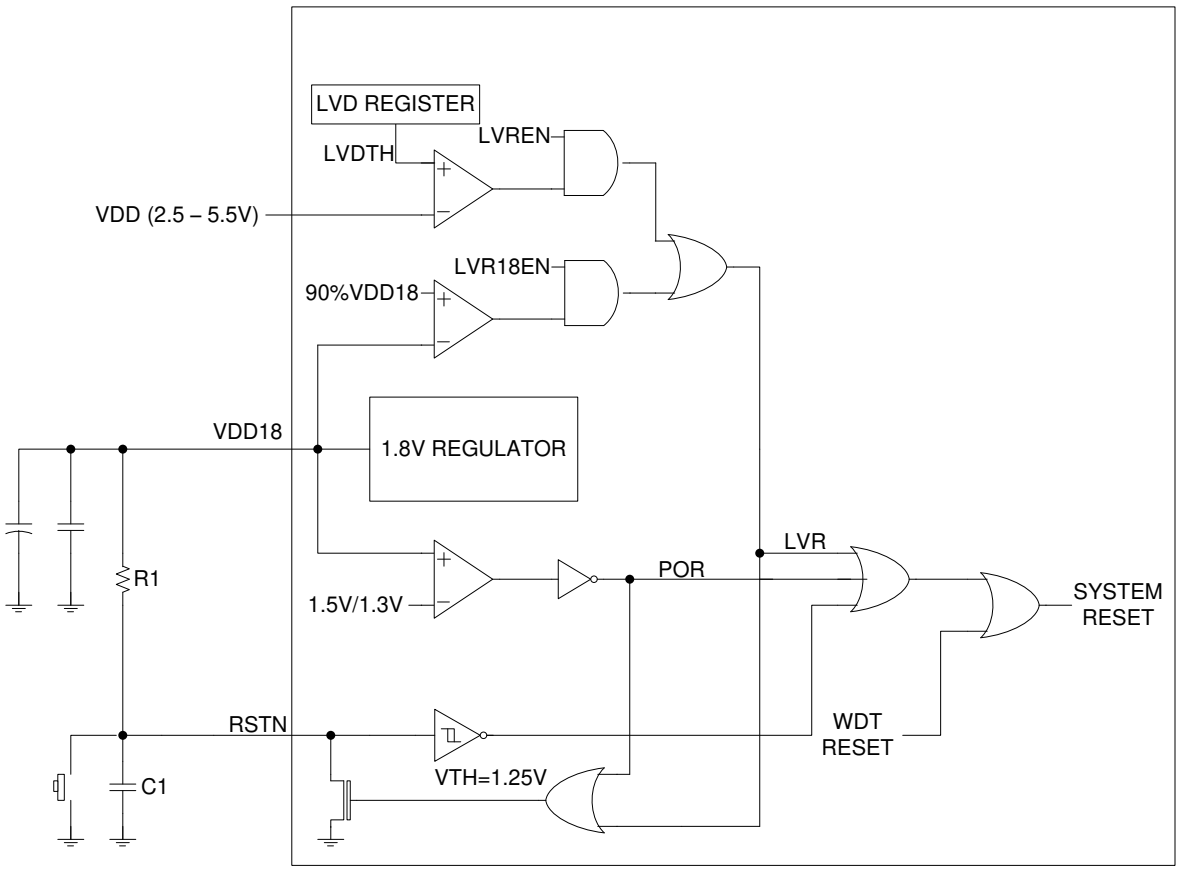
- Four 8-Bit I/O ports
- 16-bit Watch Dog Timer. WDT, WDCON, and CKCON registers
- Three 16-bit Timers, T0/T1 and T2. TCON, RLDL, RLDH, TL2, TH2, and T2CON registers
- UART0.
- I²C Master Controller. I2CMSA, I2CMCR, I2CMBUF, and I2CMTP registers.



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.

1.1 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 pin 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 (POR) is based on the detection of output level of the internal regulator. This detection also serves as the low-level detection of the core 1.8V supply voltage level. The internal regulator output is 1.8V. 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 1.8V regulator has not reached or fallen below the 90% of its target value. In case of interference that the output of the regulator is disturbed and fall below its 75% level, the power on/off reset will also be asserted.

The LVD circuits detect the main supply voltage level VDD and the threshold can be adjusted. LVD reset is disabled by default, and must be enabled by the software. Also a VDD18 LVD circuits detects the drops of 90% of VDD18 level. The LVD and LVD18 can be combined as LVR conditions. If LVR occurs, RSTN is also forced low. 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 (R1) connected to the internal regulator output, and a capacitor (C1) 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 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 0x0F000 after reset. This differs from standard 8051. In typical cases, 0x0F000 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 and SLEEP modes.

1.2 Addressing Mode and Memory Operations

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

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

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

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 is wait state register that controls the program access wait state only.

WTST[3-0] Wait State Control register. WTST set the wait state in CPU clock period

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 40 nsec. Therefore the user 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 16MHz, then WTST should be set higher than 1 to allow enough read access time.

1.3 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.

DPL (0x82) Data Pointer Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DPL[7-0]							
WR	DPL[7-0]							

DPL register holds the low byte of data pointer, DPTR.

DPH (0x83) Data Pointer High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DPH[7-0]							
WR	DPH[7-0]							

DPH register holds the high byte of data pointer, DPTR.

DPL1 (0x84) Extended Data Pointer Low R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DPL1[7-0]							
WR	DPL1[7-0]							

DPL1 register holds the low byte of extended data pointer 1, DPTR1.

DPH1 (0x85) Extended Data Pointer High R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DPH1[7-0]							
WR	DPH1[7-0]							

DPH1 register holds the high byte of extended data pointer 1, DPTR1.

DPX (0x93) Data Pointer Top R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DPX[7-0]							
WR	DPX[7-0]							

DPX is used to provide top 8-bit address of DPTR when address above 64KB. The lower 16-bit address is formed by DPH and DPL.

DPX1 (0x95) Extended Data Pointer Top R/W (0x00)

	7	6	5	4	3	2	1	0
RD	DPX1[7-0]							
WR	DPX1[7-0]							

DPX1 is used to provide top 8-bit address of DPTR when address above 64KB. The lower 16-bit address is formed by DPH and DPL.

MXAX (0xEA) MOVX Extended Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	MXAX[7-0]							
WR	MXAX[7-0]							

MXAX is used to provide top 8-bit address for an “MOVX @R0” or “MOVX @R1” instruction. The lower 16-bit address is formed by P2 and R0/R1 (if DPXREN=0), or formed by DPXR and R0/R1 (if DPXREN=1).

MCON (0xC6) XRAM Relocation Register R/W (0x00) TA Protected

	7	6	5	4	3	2	1	0
RD	MCON[7-0]							
WR	MCON[7-0]							

MCON holds the starting address of XRAM in 4KB steps. For example, if MCON[7-0]=0x01, the starting address is 0x001000h.

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.

1.4 Interrupt System

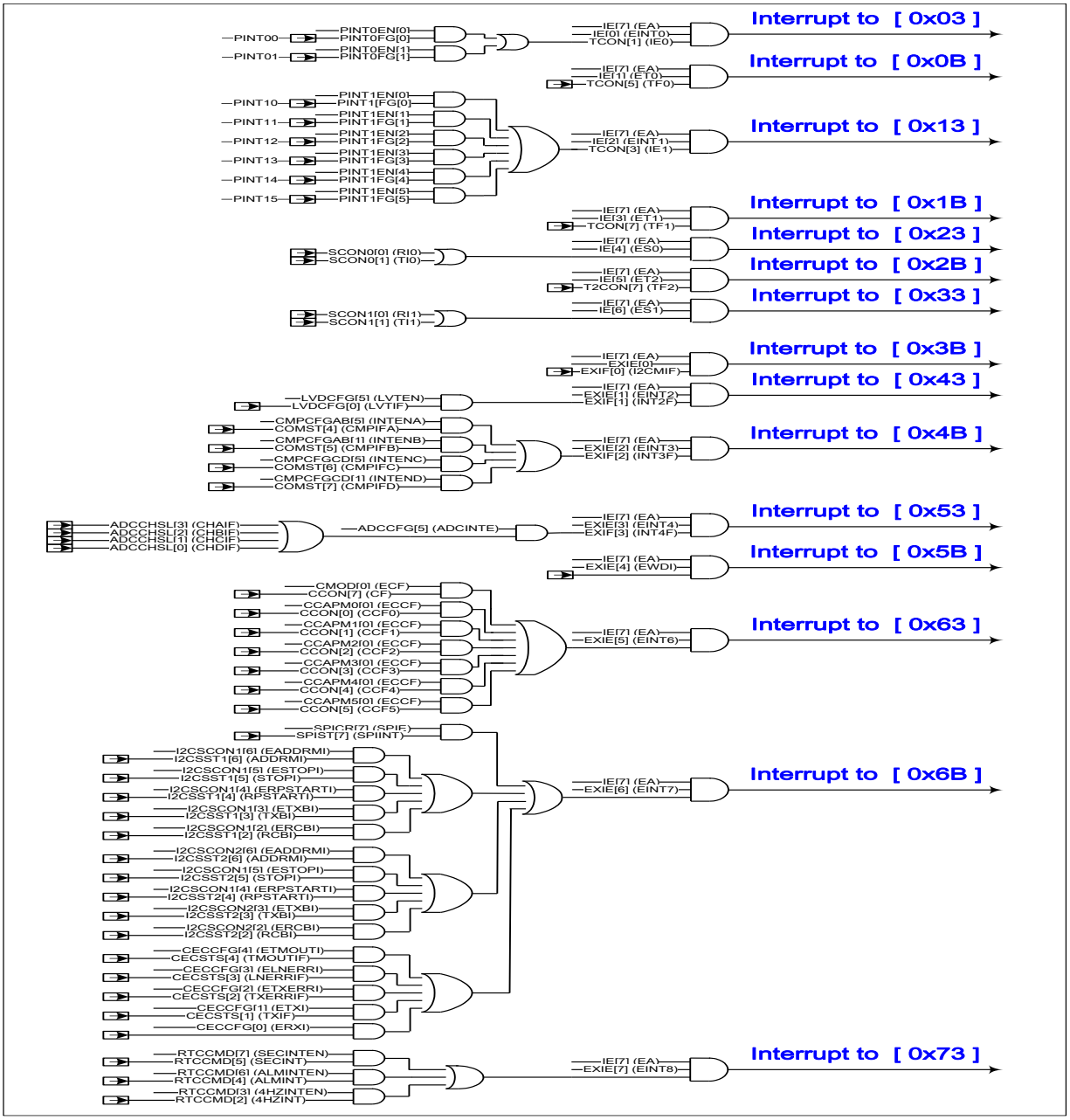
The CPU implements an enhanced Interrupt Control that allows total 15 interrupt sources and each with two programmable priority levels. The interrupts are sampled at rising edge of SYSCLK. If interrupts are present and enabled, the CPU will enter interrupt service routine by vectoring to the highest 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
TI2/RI2	LIN-capable 16550-like UART2	0x33	Software	7
I2CM	I^2C Master	0x3B	Software	8
INT2	LVT/LVT18	0x43	Software	9
INT3	Comparator (A-D) / CAN	0x4B	Software	10
INT4	ADC (A-D)/PWM	0x53	Software	11
WDIF	Watchdog	0x5B	Software	12
INT6	PCA	0x63	Software	13
INT7	SPI/I2CS1/I2CS2/CEC	0x6B	Software	14
INT8	RTC/Timer 3	0x73	Software	15
BKP	Break Point	0x7B	Software	0
DBG	I2CS Debug	0x83	Software	0

In addition to the 15 peripheral interrupts, there are two highest priority interrupts associated with debug and break point. DBG interrupt is generated when I^2C slave is configured as a debug port and a debug request from the host matches the debug ID. BKP interrupt is generated when break point match condition occurs. DBG has higher priority than BKP.

The following diagram shows the interrupt sources and the expanded pin interrupts



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	ES2	ET2	ES0	ET1	PINT1EN	ET0	PINT0EN
WR	EA	ES2	ET2	ES0	ET1	PINT1EN	ET0	PINT0EN

EA	Global Interrupt Enable bit.
ES2	LIN-capable 16550-like UART2 Interrupt Enable bit.
ET2	Timer 2 Interrupt Enable bit.
ES0	UART0 Interrupt Enable bit.
ET1	Timer 1 Interrupt Enable bit.
PINT1EN	Pin PINT1.x Interrupt Enable bit.
ET0	Timer 0 Interrupt Enable bit.
PINT0EN	Pin PINT0.x Interrupt Enable bit.

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 and Timer 3 Interrupt Enable bit.
EINT7	SPI, I²C Slave and CEC Interrupt Enable bit.
EINT6	PCA Interrupt Enable bit.
EWD1	Watchdog Timer Interrupt Enable bit.
EINT4	ADC/PWM Interrupt Enable bit.
EINT3	Analog Comparator and CAN Controller 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	-	PS2	PT2	PS0	PT1	PX1	PT0	PX0
WR	-	PS2	PT2	PS0	PT1	PX1	PT0	PX0

PS2	LIN-capable 16550-like UART2 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 and Timer 3 Priority bit.
EINT7	INT7 SPI, I²C Slave, and CEC Priority bit.
EINT6	INT6 PCA Priority bit.
EWDI	Watchdog Priority bit.
EINT4	INT4 ADC/PWM Priority bit.
EINT3	INT3 Analog Comparator and CAN Controller Priority bit.
EINT2	INT2 Low Voltage Detection Priority bit.
EI2CM	I²C Master Priority bit.

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

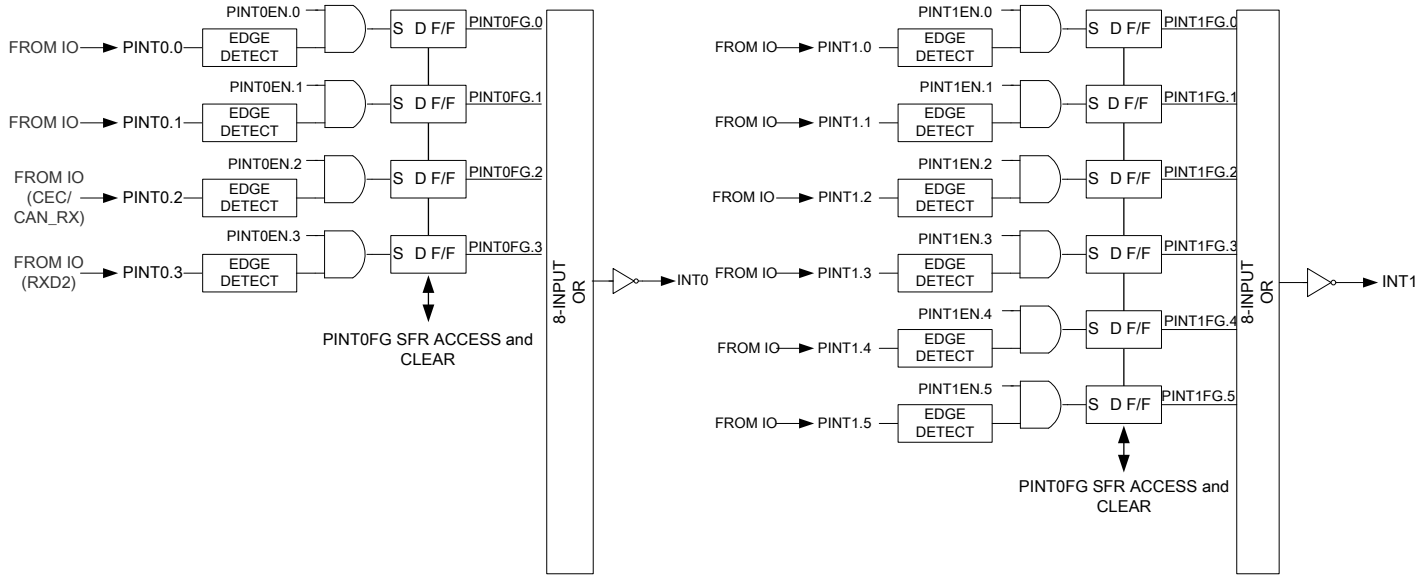
	7	6	5	4	3	2	1	0
RD	INT8F	INT7F	INT6F	-	INT4F	INT3F	INT2F	I2CMIF
WR	-	-	-	-	-	-	-	I2CMIF

INT8F	INT8 RTC and Timer 3 Interrupt Flag
INT7F	INT7 SPI, I²C Slave, and CEC interrupt Flag
INT6F	INT6 PCA Interrupt Flag
INT4F	INT4 ADC/PWM Interrupt Flag
INT3F	INT3 Analog Comparator and CAN Controller Interrupt Flag
INT2F	INT2 Low Voltage Detection Interrupt Flag
I2CMIF	I²C Master Interrupt Flag. This bit 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 EXIF 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 INT2, INT2F is set to one when one or more than one of SECINT, ALMINT, and 4HZINT of RTC is set, i.e., INT2F = (SECINT + ALMINT + 4HZINT). The software needs 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.

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.



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

	7	6	5	4	3	2	1	0
RD	TF1	TR1	TF0	TR0	PINT1F	-	PINT0F	-
WR	-	TR1	-	TR0	PINT1F	-	PINT0F	-

TF1	Timer 1 Interrupt Flag. TF1 is cleared by hardware when entering the interrupt routine.
TR1	Timer 1 Run Control bit. Set to enable Timer 1.
TF0	Timer 0 Interrupt Flag. TF0 is cleared by hardware when entering the interrupt routine.
TR0	Timer 0 Run Control bit. Set to enable Timer 0.
PINT1F	PINT1 Interrupt Flag. PINT1F is the OR status of PINT1FG.x.
PINT0F	PINT0 Interrupt Flag. PINT1F is the OR status of PINT0FG.x.

PINT0EN (0x96) W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	PINT0EN.3	PINT0EN.2	PINT0EN.1	PINT0EN.0
WR	-	-	-	-	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.3	PINT0FG.2	PINT0FG.1	PINT0FG.0
WR	-	-	-	-	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	-	-	PINT0FG.5	PINT0FG.4	PINT1EN.3	PINT1EN.2	PINT1EN.1	PINT1EN.0
WR	-	-	PINT0FG.5	PINT0FG.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	-	-	PINT0FG.5	PINT0FG.4	PINT1FG.3	PINT1FG.2	PINT1FG.1	PINT1FG.0
WR	-	-	PINT0FG.5	PINT0FG.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 registers while the PINT0EN and PINT1EN are the interrupt enable control for the expansion interrupt INT0 and INT1. PINT0.[3-0] and PINT1.[5-0] 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.[3-0] and PINT1.[5-0] 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 PINT0F or PINT1F be set 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. Please note PINT0F and PINT1F are direct reflection of the OR result of PINT0FG bits and PINT1FG bits. Therefore writing to PINT0F and PINT1F will have no effect.

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.

1.5 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	-	-	-	-	-	-	-	TASTAT
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. When read, the bit 0 of TA indicates whether TA is locked or not (1 indicates unlock and 0 indicates lock).

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. The TB protected registers include two SFR registers, CKSEL (0x8F) and WKMASK (0x9F), and twelve XFR registers, which are REGTRM (0xA000), IOSCITRM (0xA001), IOSCVTRM (0xA002), XOSCCFG (0xA007), LVDCFG (0xA010), LVDTHD (0xA011), CNTPCTL (0xA025), CNTPCTH (0xA026), INTPCT1 (0xA013), INTPCT2 (0xA014), BPINTE (0xA0E1), and SI2C_DebugID (0xA0EF). 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 registers should bear extreme cautions. It is strongly advised that these two registers 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.

1.6 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.

In order to reduce the power consumption, all analog components must be disabled before enter STOP or SLEEP mode. In ADC, the REFSEL must be set to 1 then turn-off ADC function by clear ADCEN.

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

	7	6	5	4	3	2	1	0
RD	SMOD0	-	-	-	-	-	-	-
WR	SMOD0	-	-	-	-	SLEEP	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.

SLEEP — Sleep Mode Control bit. When set to 1, the clock goes to the CPU and all peripherals is stopped and enter into SLEEP mode if STOP bit is also set together. The SLEEP mode can only be waked up by non-clocked interrupt or reset. Upon exiting SLEEP mode, SLEEP bit and STOP bit in PCON is automatically cleared. In terms of power consumption, the following relationship applies: IDLE mode > STOP mode > SLEEP mode. In essence, SLEEP mode is the same as STOP mode, except it also turns off the bandgap and the regulator. It uses a back-up very low power regulator (< 5uA). When waking up from SLEEP mode, it takes longer time (< 64 IOSC clock cycles, compared with STOP mode) because it need time for regulator to be stable.

STOP — Stop Mode Control bit. When set the clock goes to the CPU and all peripherals is stopped and enter into STOP mode if SLEEP bit is in reset state. 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. But the clock goes to peripherals external to CPU are still active, such as PCA, ADC, LIN-capable 16550-like UART2, SPI, T3, I^2C 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, CD0 — Clock Divider Control bit. 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 PMM mode where CPU and its integrated peripherals operate at a clock rate divided by 257. Note that in PMM mode, all integrated peripherals such as UART0, WDT, and T0/T1/T2 run at this reduced rate, thus may not function properly. All external peripherals to CPU still operate at full speed in PMM mode.

SWB — Switch Back Control bit. Setting this bit allows the actions to occur in integrated peripherals to automatically switch back to normal operation mode.

STATUS (0xC5) RO (0x00)

	7	6	5	4	3	2	1	0
RD	-	HIP	LIP	-	-	-	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 SLEEP, 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.
SPTA0 UART0 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 (0xC0) System Clock Selection Register TB Protected

	7	6	5	4	3	2	1	0
RD	WKDLY[2-0]			-	-	-	CLKSEL[1]	CLKSEL[0]
WR	WKDLY[2-0]			-	-	-	CLKSEL[1]	CLKSEL[0]

WKDLY[3-0] Sleep Mode Wake Up Delay in IOSC.
000 = 0
001 = 128
010 = 256
011 = 512
100 = 1024
101 = 2048
110 = 4096
111 = 8192

CLKSEL[1-0] Clock Source Selection Bit.
These two bits define the clock source of the system clock SYSCLK. The selections are shown in the following table. The default after reset is IOSC.

CLKSEL[1]	CLKSEL[0]	SYSCLK
0	0	IOSC
0	1	XOSC
1	0	RTC
1	1	SOSC

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

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.

WKMASK 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 the internal oscillator is turned on and SYSCLK resumes. The interrupt can be set as a level trigger or an edge trigger and the wake-up always runs in accordance with the edge. Please note the wake-up control is wired separately from the interrupt logic, therefore, after waking up, the CPU does not necessarily enter the interrupt service routine if the corresponding interrupt is not enabled. In this case, the CPU continues onto the next instruction which initiates the STOP mode. Extra attention should be exerted as designing the exit and re-entry of modes to ensure proper operation.

Please also note that all clocks are stopped in STOP mode, therefore peripherals requiring clock such as I^2C slave, UARTx, ADC, LVD, T3 can not perform wake up function. Only external pins and peripherals that do not require clock can be used for wake up purpose. Such peripherals in IS31CS8968A are analog comparator and RTC.

1.6.1 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, T0/T1/T2, and I²C Master operate at 257 times slower than SYSCLK. All other external peripherals such as PCA, ADC etc. are still operating under normal clock. The PMM mode saves power because the CPU, internal Flash memory and SRAM by operating at much slower frequency. The program continues to run while the CPU is operating at a reduced rate. To further save power, the 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 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 OR-ed with these interrupts.

UART0 receive Start bit detection

UART0 transmit buffer loaded

When an 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 may not operate correctly to receive or transmit data. 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 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 recovers to normal mode automatically when it writes in the transmit buffer. Once it recovers, UART operates under normal frequency to correctly transmit the data.

The return of PMM mode after switchback must be activated manually with software. The exit of PMM mode occurs when WDT or external RSTN resets.

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

1.6.2 IDLE Mode

IDLE mode provides a further power saving than PMM mode by stopping the clock for CPU and its integrated peripherals while keeping the external peripherals at normal operating conditions. The external peripherals still function normally thus can generate interrupts that wake up the CPU from IDLE mode. The IDLE mode is introduced by setting Idle bits 1.

The CPU halts in the idle mode, hence no processing is possible. All integrated internal peripherals such as T0/T1/T2, UART0, and I²C Master are inaccessible during idling. The IDLE mode can be excited by hardware reset through RSTN pin or by external interrupts as well as the interrupts from external peripherals that are ORed with the external interrupts. The triggering external interrupts need be enabled properly. Upon exiting from IDLE mode, the CPU resumes operation as the clock is being turned on. CPU immediately vectors to the interrupt service routine of the corresponding interrupt sources that wake up the CPU. When the interrupt service routine completes, RETI returns to the program and immediately follows the one that invokes the IDLE mode. Upon returning from IDLE mode to normal mode, Idle bit in PCON is automatically cleared. As the purpose of the IDLE mode is to save power, the use of IOSC clock is strongly recommended in place of SYSCLK before entering IDLE mode since it consumes significantly less power than the crystal oscillator or other clock sources.

1.6.3 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 consumption, before entering STOP mode, it is essential to turn off all peripherals and the current operating clock oscillators such as crystal oscillator and PLL. It is also important that the software switches to the IOSC clock and disables all other clock generators such as crystal oscillator or PLL clock generator before entering STOP mode. This is critical to ensure a smooth transition when resuming its normal operations. Selecting other clock sources, such as XTAL oscillator or PLL clock as CPU system clock may burden the system as the clock sources may take a significant amount of time to stabilize during the wake-up. Upon entering STOP mode, the system uses the last edge of IOSC clock to shut down the IOSC clock generator. The minimum power consumption state is achieved through this mechanism.

Hardware reset through RSTN pin or by interrupts generated via external pins (INT0 and INT1) or INT2 to INT8 brings the system out of STOP mode. Since all clocks are inactive, none of the peripherals like UART, Timers, I²C master and slave, ADC, or LVD contribute to the exit of STOP mode. Peripherals like Analog comparator and RTC interrupt; however, can be used to trigger the exit of STOP mode as they are implemented asynchronously or their own clock sources.

The triggering interrupt source must be enabled and its Wake-up bit is set in the WKMASK register. External pins require LOW-level triggers; however the INT flags of on-chip external peripherals require HIGH-level triggers. The IOSC circuit is activated by triggering event and the CPU is woken up at the first IOSC clock edge. Please note that the IOSC is activated as soon as STOP mode exits. As CPU resumes the normal operation using the IOSC clock when an interrupt presents, the CPU immediately vectors to the interrupting service routine of the corresponding interrupt source. When the interrupt service routine completes, RETI returns to the program immediately to execute the instruction that invokes the STOP mode. The Stop bit in PCON is automatically cleared by hardware reset during the waking up.

Please note the wake-up control WKMASK register and interrupt enable registers IE and EXIE which are specifically responsible for the wake-up and interrupt. Extra attention should be taken while programming for coherent application design. In STOP mode, clocks of CPU and peripherals are disabled (except RTC). Therefore only external pins and peripherals such as analog comparator and RTC that do not require clock can be used to initiate the wake-up process. Peripherals such as UART, Timers, I²C master and slave, ADC, or LVD can not generate wake-up interrupt in this mode.

1.6.4 SLEEP Mode

In STOP mode, the main regulator providing 1.8V (VDDC) to internal logic, memory and flash circuits are still active. The regulator and its internal Bandgap reference circuits consumes approximately about 200uA. SLEEP mode is used to further reduce the standby power through turning off the regulator and reference circuits. The logic behavior of SLEEP mode is the same as STOP mode and is entered by setting both STOP and SLEEP bits to 1 in PCON register. In SLEEP mode, a very low-power back-up regulator is used to provide supply voltage to the internal logic, memory and flash circuits. The back-up regulator consumes about 10uA to 20uA, and can supply up to 1mA of load. The output voltage of the back-up regulator is lower than the main regulator, and typically is around 1.45V.

The exit of SLEEP mode is the same as exit of STOP mode by wake-up events, and exits directly back to normal operation and the main regulator is turned on. Note the enabling time of the main regulator is about 10usec, therefore, after wake-up from SLEEP mode, the software should be kept at NOP for at least 20usec before resuming. It is also recommended that if SLEEP mode is used, the decoupling capacitor on VDDC should contains at least 4.7uF.

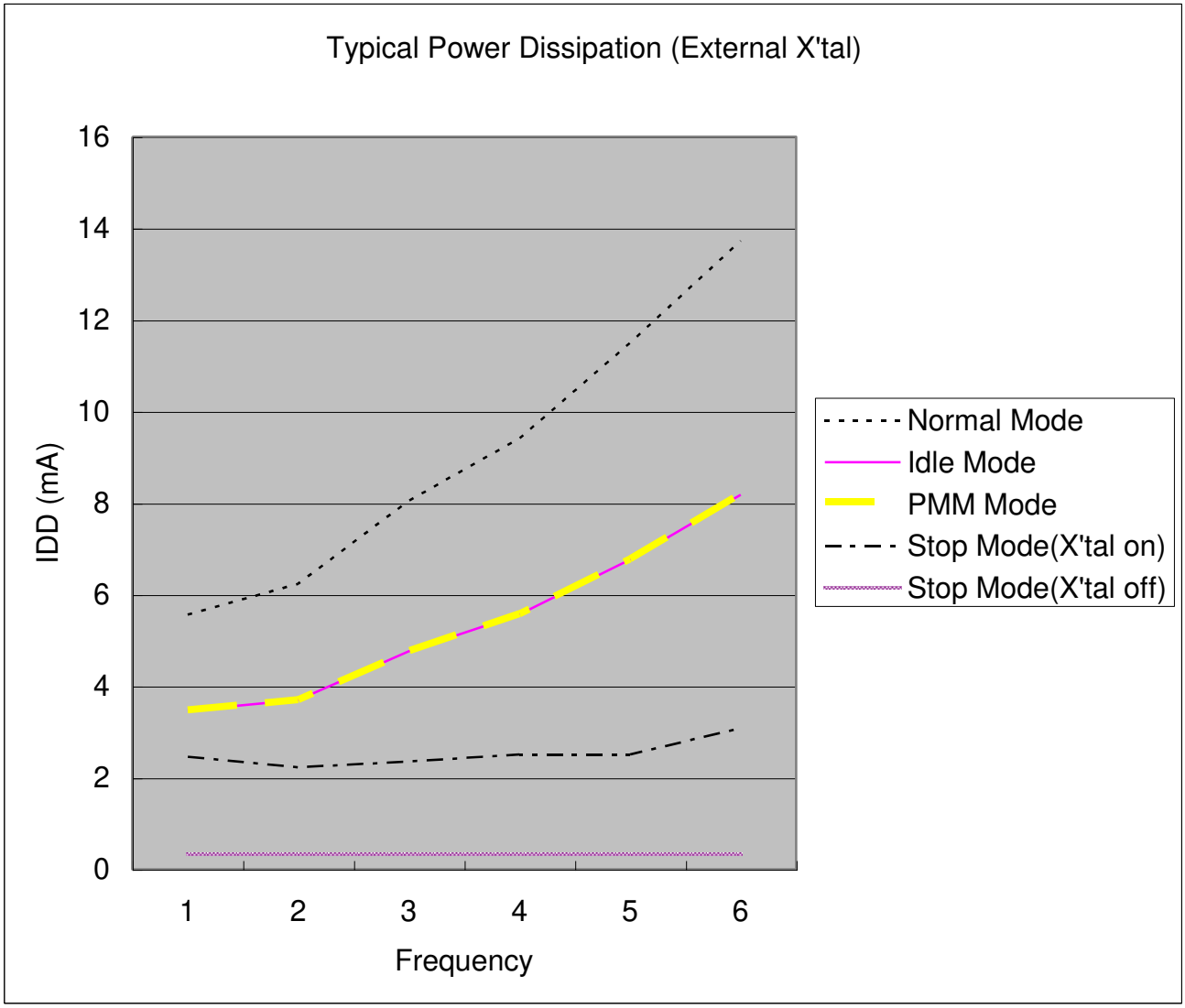
1.6.5 Clock Control

The clock selection is defined by CKSEL register (0x8F). An IOSC is a critical component in MCU although not integrated in the CPU core. It is enabled except in STOP mode. An IOSC also handles critical timing conformance for flash programming and the default manufactured calibrated IOSC is set at 16MHz. Although users can manually reset the IOSC frequency but reset value should not deviate more than 50% from its typical setting to avoid flash performance problems.

An IOSC is recommended that for the transition of clock-source-switching to ensure a smooth and glitch-free transition. This is also true for switching among different power saving modes. Please note that when waking up from STOP mode, the clock selection is switched automatically to IOSC. If other clock sources are preferred, optional configurations are available through software set-up.

When switching clock sources, it is also important to note the crystal oscillator, real time clock and the phase lock loop take a significant amount of time to stabilize. The software needs to be designed to turn on the corresponding clock source first and wait for the stabilization time before CKSEL settings take place.

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



The values of performance frequency in IDLE and PMM modes are close therefore the lines appear overlapped in the graph.

The IDD result does not include the power dissipation of the clock oscillator. The graph shows that during normal operation, the power dissipation increases approximately at ~0.36mA/MHz; in idle mode it increases at about ~0.2mA/MHz (the power dissipation still increases as the frequency increases due to operation of peripheral clock).

WARNING: If an uninstalled clock source is being selected, it may cause the system to hang. There is NO hardware protection against this peril. Therefore extreme precautions must be exerted during programming.

1.7 Break Point Controller

The CPU core also includes a Break Point Controller for software debugging purposes and handling exceptions. Program Counter break point triggers at PC address matching, and there are seven PC matching settings available. Single Step break point triggers at interaction return from an interrupt routine.

Upon the matching of break point conditions, the Break Point Controller issues BKP Interrupt for handling the break points. The BKP Interrupt vector is located at 0x7B. Upon entering the BKP ISR (Break Point Interrupt Service Routine), all interrupts and counters (WDT, T0, T1, and T2) are disabled. To allow further interrupts and continuing counting, the BKP ISR must be enabled. At the exiting, the BKP ISR setting must be restored to resume normal operations.

BPINTF (A0E0h) Break Point Interrupt Flag Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	STEP_IF	PC7IF	PC6IF	PC5IF	PC4IF	PC3IF	PC2IF	PC1IF
WR	STEP_IF	PC7IF	PC6IF	PC5IF	PC4IF	PC3IF	PC2IF	PC1IF

This register is for reading the Break Points interrupt flags.

STEP_IF This bit is set when the Break Point conditions set by a new instruction fetching from an interrupt routine. This bit must be cleared by software.

PC7IF – PC1IF These bits are set when Break Point conditions set by PC7 – PC1 address are met. These bits must be cleared by software.

BPINTE (A0E1h) Break Point Interrupt Enable Register R/W (0x00) TB Protected

	7	6	5	4	3	2	1	0
RD	STEP_IE	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE
WR	STEP_IE	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE

This register controls the enabling of individual Break Points interrupt.

STEP_IE Set this bit to enable Single Step event break point interrupt.

PC7IE – PC1IE Set these bits to enable PC7 to PC1 address match break point interrupt.

BPINTC (A0E2h) Break Point Interrupt Control Register R/W (0x00)

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

This register is reserved for future application.

BPCTRL (A0E3h) DBG and BKP ISR Control and Status Register R/W (b'11111100)

	7	6	5	4	3	2	1	0
RD	DBGINTEN	DBGWDTEN	DBGT2EN	DBGT1EN	DBGT0EN	-	-	DBGGST
WR	DBGINTEN	DBGWDTEN	DBGT2EN	DBGT1EN	DBGT0EN	-	-	DBGGST

When entering the DBG or BKP ISR (Interrupt Service Routine), all interrupts and timers are disabled. The enabled bits are cleared by hardware reset in this register. As the interrupts and timers are disabled, the ISR can process debugging requirement in a suspended state. If a specific timer should be kept active, it must be enabled by ISR after ISR entry. Before exit of DBG and BKP ISR, the control bits should be enabled to allow the timers to resume operating. This register should be modified only in Debug ISR.

DBGINTEN Set this bit to enable all interrupts (except WDT interrupt). This bit is cleared automatically at the entry of DBG and BKP ISR. Set this bit to allow ISR to be further interrupted by other interrupts. This is sometimes necessary if DBG or BKP ISR needs to use UART or I^2C, for example.

DBGWDEN Set this bit to allow WDT counting during the DBG and BKP ISR. This bit should always be set before exiting the ISR.

DBGT2EN Set this bit to allow T2 counting during the DBG and BKP ISR. This bit should always be set before exiting the ISR. This bit only controls the counting but not T2 interrupt.

DBGT1EN Set this bit to allow T1 counting during the DBG and BKP ISR. This bit should always be set before exiting the ISR. This bit only controls the counting but not T1 interrupt.

DBGT0EN Set this bit to allow T0 counting during the DBG and BKP ISR. This bit should always be set before exiting the ISR. This bit only controls the counting but not T0 interrupt.

DBGST This bit indicates the DBG and BKP ISR status. This bit is set to 1 when entering DBG and BKP ISR. This signal should be cleared when exiting the DBG and BKP ISR. Checking this bit allows other interrupt routine to determine whether it is a sub-service of the DBG and BKP ISR.

PC1AL (A0F0h) Program Counter Break Point 1 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC1AL[7-0]							
WR	PC1AL[7-0]							

This register defines the PC low address for PC match break point 1.

PC1AH (A0F1h) Program Counter Break Point 1 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC1AH[7-0]							
WR	PC1AH[7-0]							

This register defines the PC high address for PC match break point 1.

PC1AT (A0F2h) Program Counter Break Point 1 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC1AT[7-0]							
WR	PC1AT[7-0]							

This register defines the PC top address for PC match break point 1. PC1AT:PC1HT:PC1LT together form the 24 bit compare value of break point 1 for Program Counter.

PC2AL (A0F4h) Program Counter Break Point 2 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC2AL[7-0]							
WR	PC2AL[7-0]							

This register defines the PC low address for PC match break point 2.

PC2AH (A0F5h) Program Counter Break Point 2 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC2AH[7-0]							
WR	PC2AH[7-0]							

This register defines the PC high address for PC match break point 2.

PC2AT (A0F6h) Program Counter Break Point 2 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC2AT[7-0]							
WR	PC2AT[7-0]							

This register defines the PC top address for PC match break point 2 PC2AT:PC2HT:PC2LT together form the 24 bit compare value of PC break point 2 for Program Counter.

PC3AL (A0F8h) Program Counter Break Point 3 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC3AL[7-0]							
WR	PC3AL[7-0]							

This register defines the PC low address for PC match break point 3.

PC3AH (A0F9h) Program Counter Break Point 3 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC3AH[7-0]							
WR	PC3AH[7-0]							

This register defines the PC high address for PC match break point 3.

PC3AT (A0FAh) Program Counter Break Point 3 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC3AT[7-0]							
WR	PC3AT[7-0]							

This register defines the PC top address for PC match break point 3. PC3AT:PC3HT:PC3LT together form the 24 bit compare value of break point 3 for Program Counter.

PC4AL (A0FCh) Program Counter Break Point 4 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC4AL[7-0]							
WR	PC4AL[7-0]							

This register defines the PC low address for PC match break point 4.

PC4AH (A0FDh) Program Counter Break Point 4 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC4AH[7-0]							
WR	PC4AH[7-0]							

This register defines the PC high address for PC match break point 4.

PC4AT (A0FEh) Program Counter Break Point 3 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC4AT[7-0]							
WR	PC4AT[7-0]							

This register defines the PC top address for PC match break point 4. PC4AT:PC4HT:PC4LT together form the 24 bit compare value of break point 4 for Program Counter.

PC5AL (A0E4h) Program Counter Break Point 5 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC5AL[7-0]							
WR	PC5AL[7-0]							

This register defines the PC low address for PC match break point 5.

PC5AH (A0E5h) Program Counter Break Point 5 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC5AH[7-0]							
WR	PC5AH[7-0]							

This register defines the PC high address for PC match break point 5.

PC5AT (A0E6h) Program Counter Break Point 5 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC5AT[7-0]							
WR	PC5AT[7-0]							

This register defines the PC top address for PC match break point 5. PC5AT:PC5HT:PC5LT together form the 24 bit compare value of break point 5 for Program Counter.

PC6AL (A0E8h) Program Counter Break Point 6 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC6AL[7-0]							
WR	PC6AL[7-0]							

This register defines the PC low address for PC match break point 6.

PC6AH (A0E9h) Program Counter Break Point 6 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC6AH[7-0]							
WR	PC6AH[7-0]							

This register defines the PC high address for PC match break point 6.

PC6AT (A0EAh) Program Counter Break Point 6 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC6AT[7-0]							
WR	PC6AT[7-0]							

This register defines the PC top address for PC match break point 6 PC6AT:PC6HT:PC6LT together form the 24 bit compare value of PC break point 6 for Program Counter.

PC7AL (A0ECh) Program Counter Break Point 7 Low Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC7AL[7-0]							
WR	PC7AL[7-0]							

This register defines the PC low address for PC match break point 7.

PC7AH (A0EDh) Program Counter Break Point 7 High Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC7AH[7-0]							
WR	PC7AH[7-0]							

This register defines the PC high address for PC match break point 7.

PC7AT (A0EEh) Program Counter Break Point 7 Top Address Register R/W (b'00000000)

	7	6	5	4	3	2	1	0
RD	PC7AT[7-0]							
WR	PC7AT[7-0]							

This register defines the PC top address for PC match break point 7. PC7AT:PC7HT:PC7LT together form the 24 bit compare value of break point 7 for Program Counter.

STEPCTRL (A0FFh) Single Step Break Point Interrupt Control Register R/W (0x00)

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

The value "0x96" must be programed into this register to enable the single step break point interrupt.

Host or program can obtain the status of the break point controller through the current break point address and next PC address register. DBPCID[23-0] contains the PC address of just executed instruction when the break point occurs. DBNXPC[23-0] contains the next PC address to be executed when the break point occurs, therefore, it is usually exactly the same value of the break pointer setting.

DBPCIDL (A098h) Debug Program Counter Address Low Register RO (b'00000000)

	7	6	5	4	3	2	1	0
RD	DBPCID[7-0]							
WR	-							

DBPCIDH (A099h) Debug Program Counter Address High Register RO (b'00000000)

	7	6	5	4	3	2	1	0
RD	DBPCID[15-8]							
WR	-							

DBPCIDT (A09Ah) Debug Program Counter Address Top Register RO (b'00000000)

	7	6	5	4	3	2	1	0
RD	DBPCID[23-16]							
WR	-							

DBPCNXL (A09Bh) Debug Program Counter Next Address Low Register RO (b'00000000)

	7	6	5	4	3	2	1	0
RD	DBPCNX[7-0]							
WR	-							

DBPCNXH (A09Ch) Debug Program Counter Next Address High Register RO (b'00000000)

	7	6	5	4	3	2	1	0
RD	DBPCNX[15-8]							
WR	-							

DBPCNXT (A09Dh) Debug Program Counter Next Address Top Register RO (b'00000000)

	7	6	5	4	3	2	1	0
RD	DBPCNX[23-16]							
WR	-							

1.8 Debug and ISP

The I²C Slave 2 (I2CS2) can be configured as the debug and ISP port for IS31CS8968A. This is achieved by assigning a predefined debug ID for the I²C Slave address. When a host issues an I²C access to this special address, a DBG interrupt is generated. DBG Interrupt has the highest priority. The DBG interrupt vector is located at 0x83. DBG ISR is used to communicate with the host and is usually closely associated with BKP ISR.

SI2CDBGID (A0EFh) Slave I²C Debug ID Register R/W (b'00110110) TB Protected

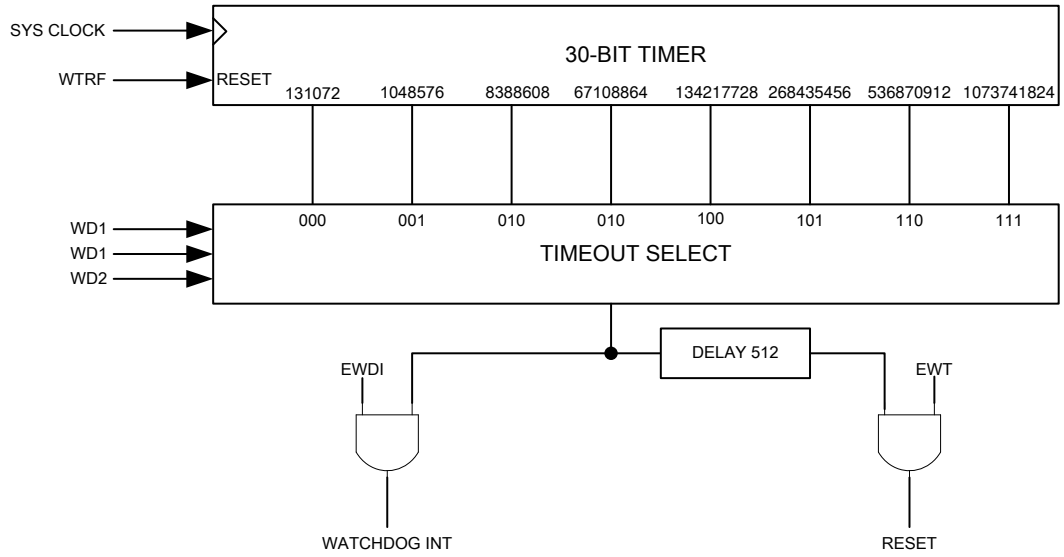
	7	6	5	4	3	2	1	0
RD	DBGSI2C2EN	SI2CDBGID[6:0]						
WR	DBGSI2C2EN	SI2CDBGID[6:0]						

DBGSI2C2EN DBGSI2C2EN=1 enables I2CS2 as debug port. When I2CS2 receives an access of I²C address matching SI2CDBGID[6:0], a debug interrupt is generated.

SI2CDBGID[6:0] Slave I²C ID address for debug function.

1.9 Watchdog Timer

The Watchdog Timer is a 30-bit timer that can be used by a system supervisor or as an event timer. The 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 its functions. The following diagram shows the structure of the Watchdog Timer. Note WDT shares the same clock with the CPU, thus WDT is disabled in IDLE mode or STOP mode however it runs at a reduced rate in PMM mode.



WDCON (0xD8) R/W (0x02)

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

WDIF — WDT Interrupt Flag bit. This bit is set when the session expires regardless of a WDT interrupt is enabled or not. Note the WDT interrupt enable control is located in EIE (0xE8).4 EWDI bit. It must be cleared by software

WTRF — WDT Reset Flag bit. A hardware reset generates a WDT reset whereas a software reset does not. It can be cleared by software or external reset from RSTN pin

EWT — Watchdog Timer Reset Enable bit. Set this bit to enable the watchdog reset function. The default WDT reset is enabled and WDT timeout is set to maximum.

RWT — Reset the Watchdog timer. Setting RWT resets the timer counting. The clearing action of Watchdog timer is protected by TA access. In another word, to clear Watchdog timer, TA must be unlocked then and then followed by writing RWT bit to 1. If TA is still locked, the program can write 1 into RWT bit, but it does not reset the Watchdog timer.

CKCON (0x8E) R/W (0xC0)

	7	6	5	4	3	2	1	0
RD	WD1	WD0	T2CKDCTL	T1CKDCTL	T0CKDCTL	WD2	-	-
WR	WD1	WD0	T2CKDCTL	T1CKDCTL	T0CKDCTL	WD2	-	-

T2CKDCTL — Timer 2 Clock Source Division Factor Control Flag. Setting this bit to 1 sets the Timer 2 division factor to 4, the Timer 2 clock frequency equals CPU clock frequency divided by 4. Setting this bit to 0 (the default power on value) sets the Timer 2 division factor to 12, the Timer 2 clock frequency equals CPU clock frequency divided by 12..

T1CKDCTL — Timer 1 Clock Source Division Factor Control Flag. Setting this bit to 1 sets the Timer 1 division factor to 4, the Timer 1 clock frequency equals CPU clock frequency divided by 4. Setting this bit to 0 (the default power on value) sets the Timer 1 division factor to 12, the Timer 1 clock frequency equals CPU clock frequency divided by 12.

T0CKDCTL — Timer 0 Clock Source Division Factor Control Flag. Setting this bit to 1 sets the Timer 0 division factor to 4, the Timer 0 clock frequency equals CPU clock frequency divided by 4. Setting this bit to 0 (the default power on value) sets the Timer 0 division factor equals 12, the Timer 0 clock frequency equals CPU clock frequency divided by 12.

WD[2:0] — This register controls the time out value of WDT as the following table. The time out value is shown as follows and the default is set to “011” and about 4 sec.

WD2	WD1	WD0	Time Out Value
0	0	0	131072
0	0	1	1048576
0	1	0	8388608
0	1	1	67108864
1	0	0	134217728
1	0	1	268435456
1	1	0	536870912
1	1	1	1073741824

1.10 System Timers – T0, T1

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

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 bit. 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 bit. 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	These bits are related to configurations of expanded interrupt INT1 and INT0. These are described in the Interrupt System section.

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

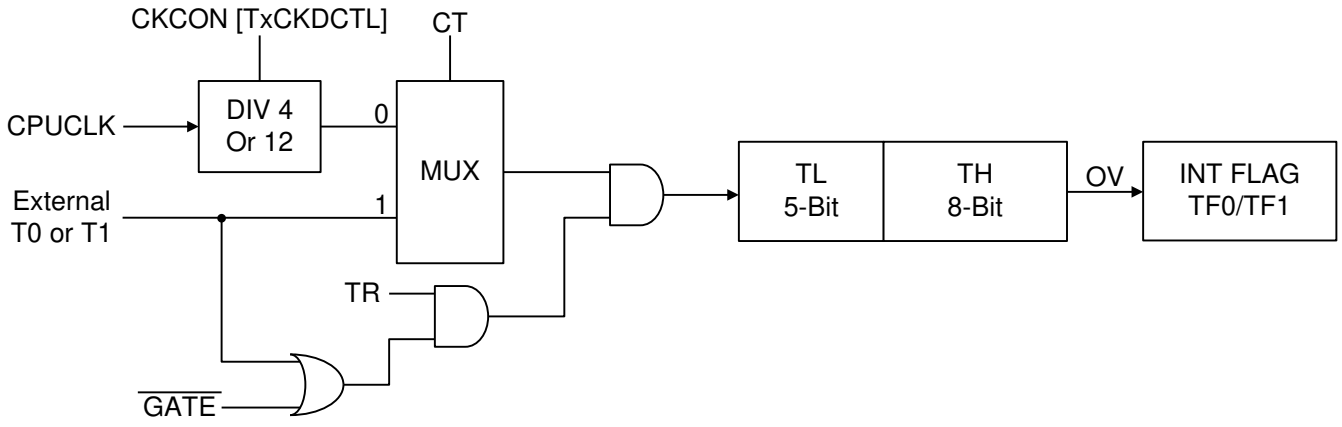
	7	6	5	4	3	2	1	0
RD	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0
WR	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0

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 access 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 T0 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.

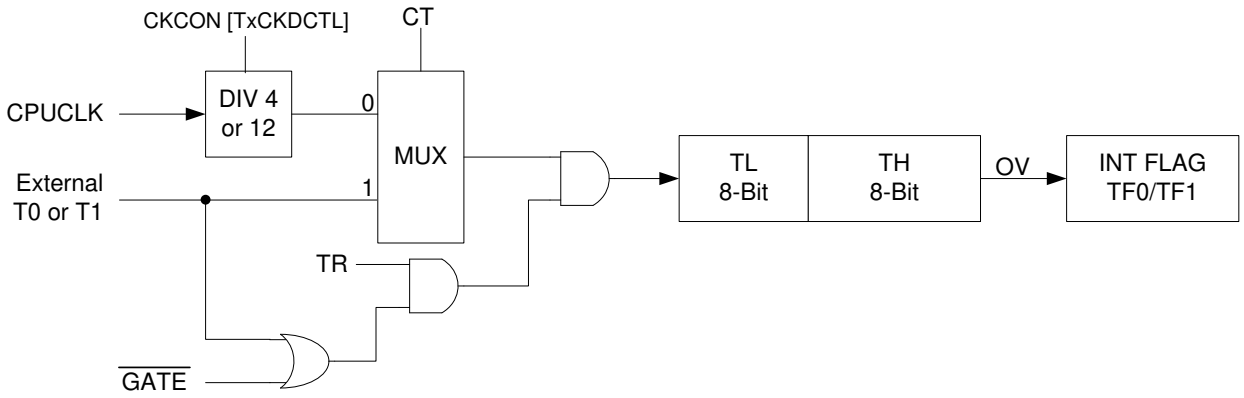
1.10.1 Mode 0

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.



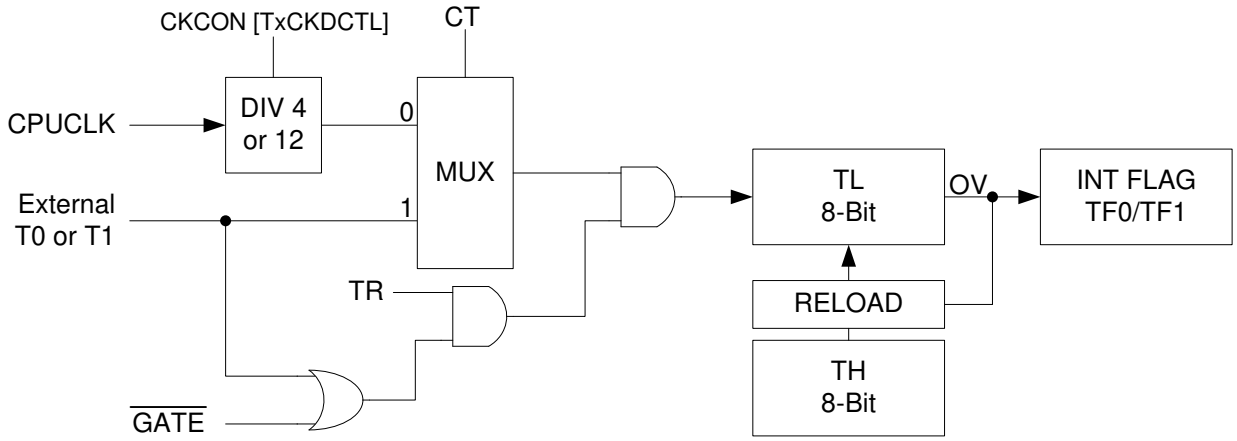
1.10.2 Mode 1

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.



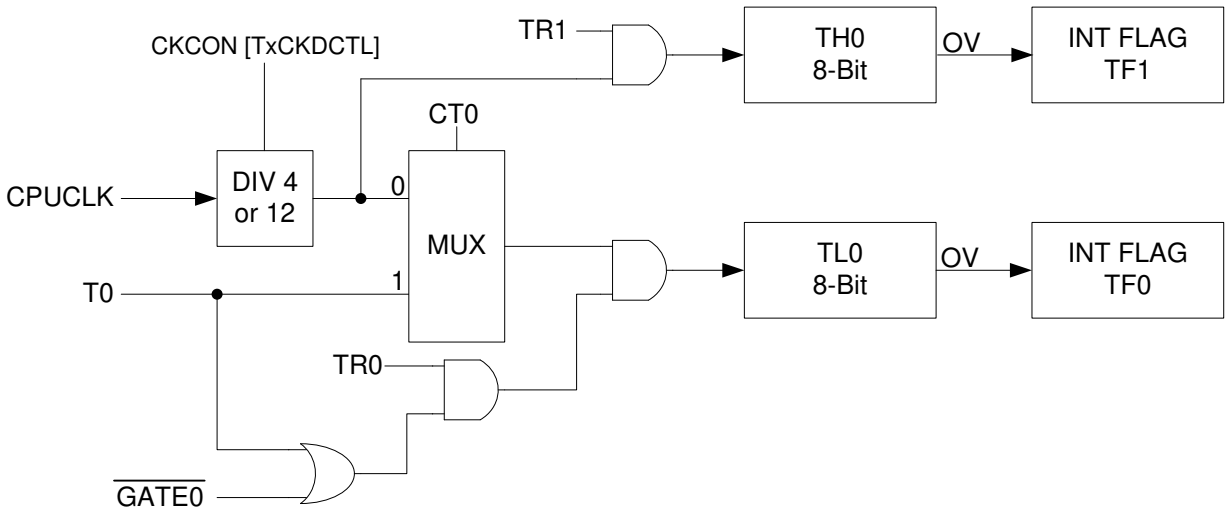
1.10.3 Mode 2

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.



1.10.4 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.



1.11 System Timer – T2

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

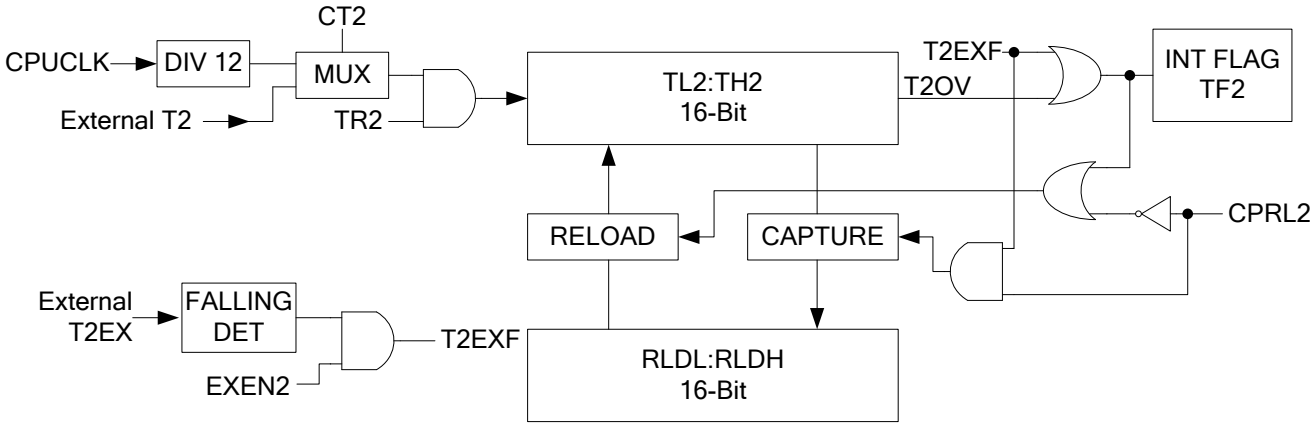
CT2 Timer 2 Timer/Counter Mode Select
1 – External event counter uses T2 pin as the clock source
0 – Internal clocked timer mode

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.

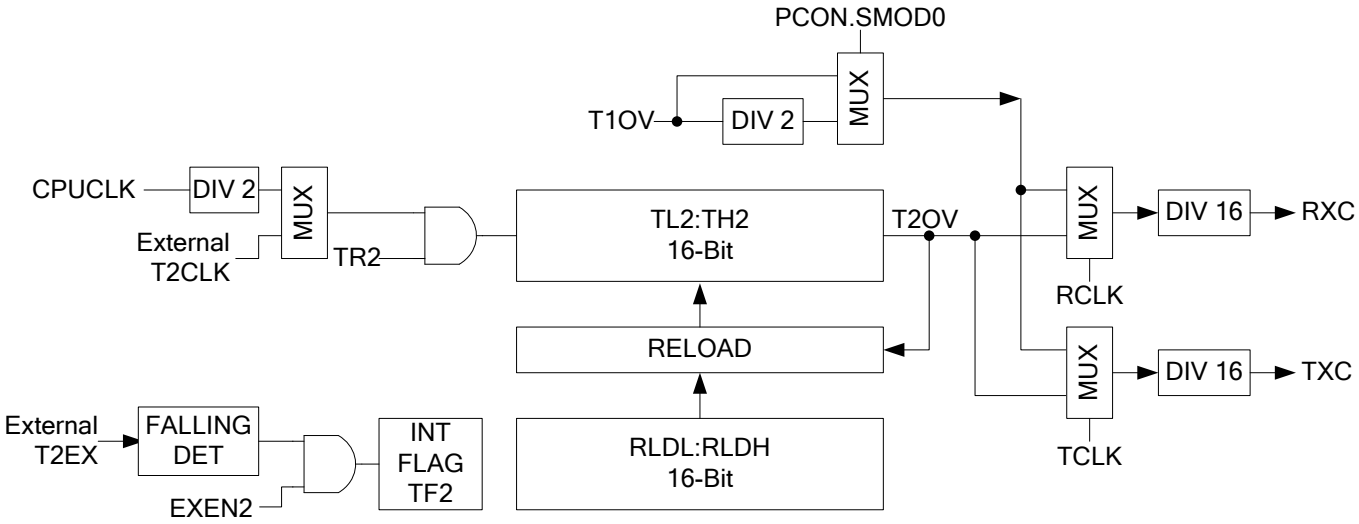
Timer 2 can be configured in three modes of operations – auto-reload counter, or capture timer, or baud rate generator. These modes are defined by RCLK, TCLK, CPRL2 and TR2 bits of T2CON registers. The definition is illustrated in the following table.

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's 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.

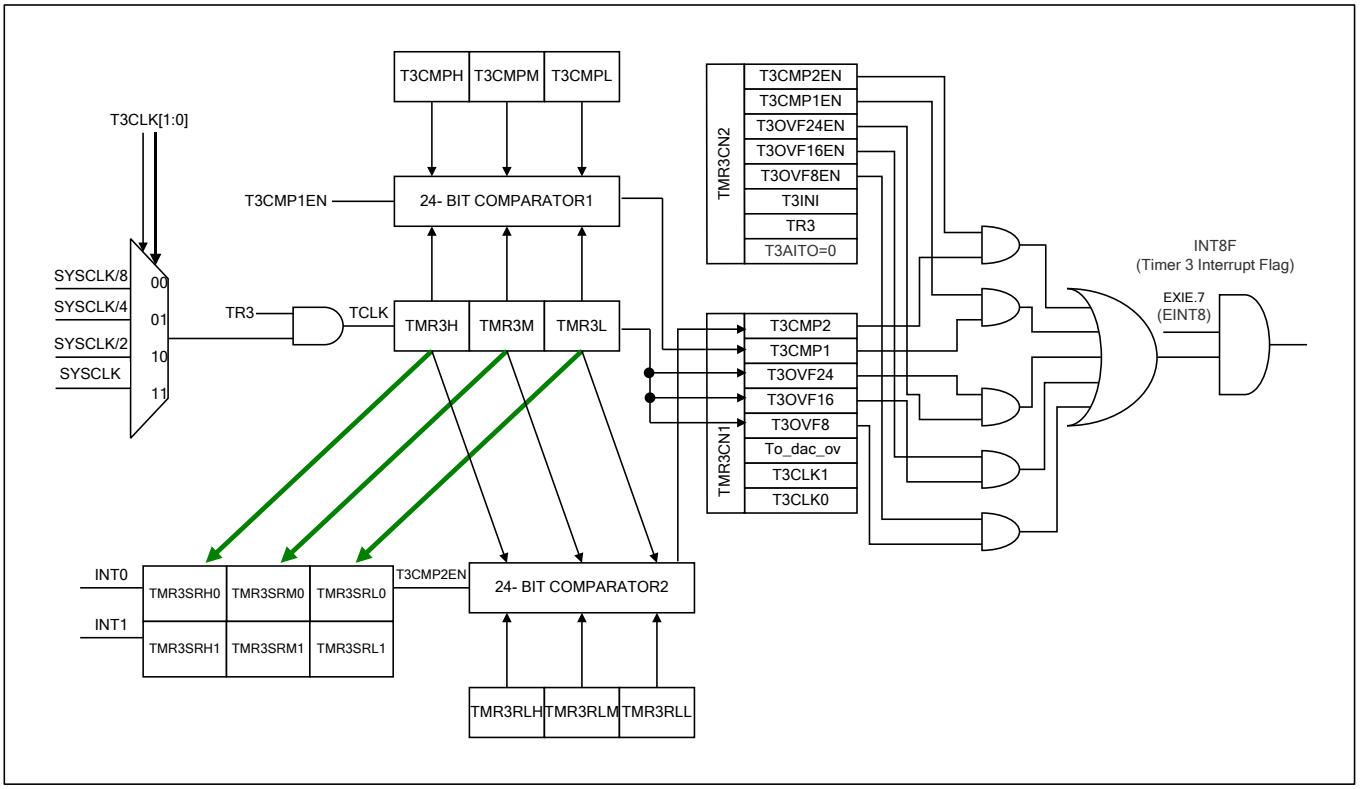


The block diagram of the Timer 2 operating in Baud Rate Generator is shown in the following diagram.

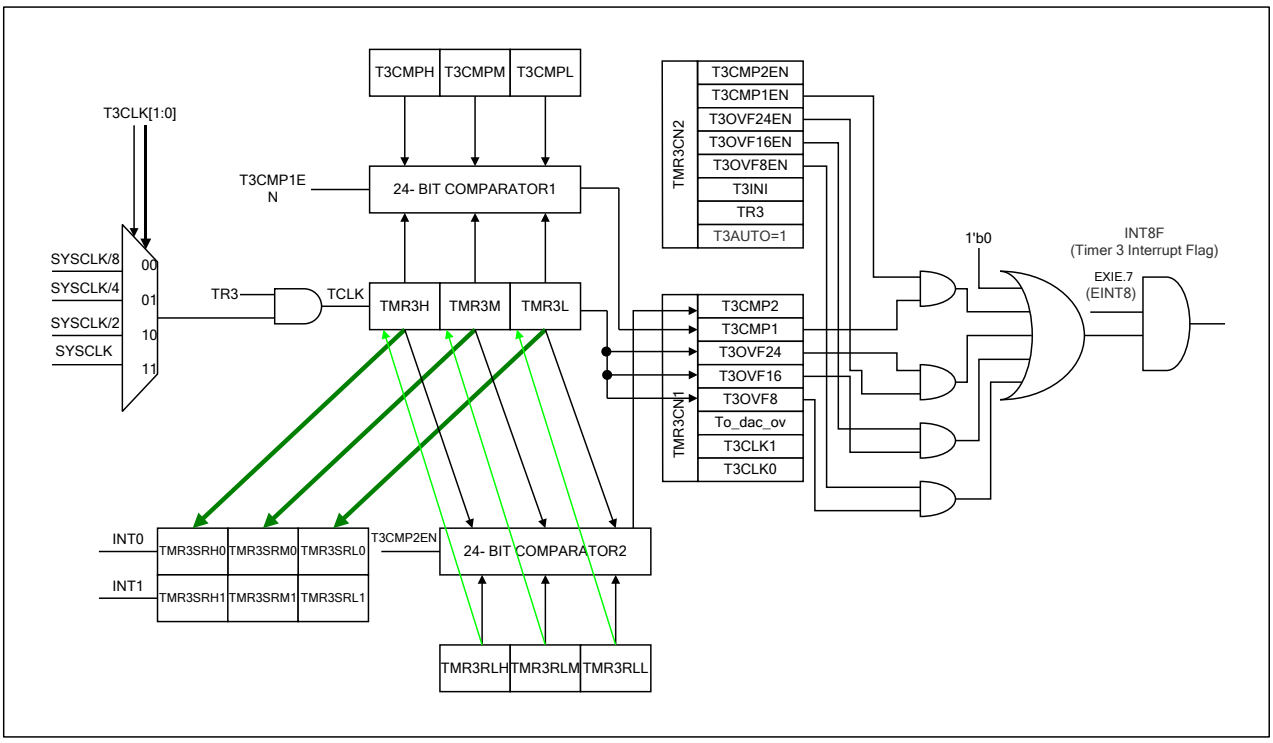


1.12 System Timer – T3

Timer 3 is a 24-bit timer formed by three 8-bit counter registers - TMR3H, TMR3M and TMR3L. The counter can operate in either FREE-RUN or AUTO-RELOAD mode. The clock of TMR3 can be configured by T3CLK[1-0] and to assign either of SYSCLK, SYSCLK/2, SYSCLK/4, or SYSCLK/8 to work. If T3AUTO=0, it operates in FREE-RUN mode. TMR3[23-0] counter increments is defined by T3 clock and overflows from 0xFFFFFF to 0x000000. The following figure shows FREE-RUN mode. In this mode, the reload register TMR3RL is used for another comparator and triggers T3CMP2.



When T3AUTO is 1, T3 operates at AUTO-RELOAD mode. TMR3[23-0] counter increments and overflows from 0xFFFFFF then reload from the register TMR3RL[23-0]. TheAUTO-RELOAD operation is shown in the following diagram.



TMR3CN1 (A100h) Timer 3 Control Register One R/W 00000000

	7	6	5	4	3	2	1	0
RD	T3CMP2F	T3CMP1F	T3OVF24	T3OVF16	T3OVF8	-	T3CLK1	T3CLK0
WR	T3CMP2F	T3CMP1F	T3OVF24	T3OVF16	T3OVF8	WG	T3CLK1	T3CLK0

T3CMP2F — Timer 3 Compare 2 Match Flag bit
T3CMP2F is set to 1 by hardware, WhenT3AUTO=0, T3CMP2EN=1 and TMR3 counter reaches the same value asTMR3RL,a TMR3 interrupt is generated. This bit must be cleared by software.

T3CMP1F — Timer 3 Compare 1 Match Flag bit
T3CMP1F is set to 1 by hardware, WhenT3CMP1EN=1 and TMR3 counter reaches the same value as T3CMP register, a TMR3 interrupt is generated. This bit must be cleared by software.

T3OVF24 — Timer 3 24-bit Overflow Flag bit
T3OVF24 is set to 1 by hardware, When T3OVF24EN=1 and TMR3 counter overflows from 0xFFFFFF to 0x000000, a TMR3 interrupt is generated. This bit must be cleared by software.

T3OVF16 — Timer 3 16-bit Overflow Flag bit
T3OVF16 is set to 1 by hardware, When T3OVF16EN=1 and TMR3 counter overflows from 0xXXFFFF to 0xXX0000, a TMR3 interrupt is also generated. This bit must be cleared by software.

T3OVF8 — Timer 3 8-bit Overflow Flag bit
T3OVF8 is set to 1 by hardware, When T3OVF8EN=1 and TMR3 counter overflows from 0xXXXXFF to 0xXXXX00, a TMR3 interrupt is also generated. This bit must be cleared by software.

WG — Write Gate Flag bit
If WG = 1 when written, then only bits [7-3] are written.
If WG = 0 when written, then bits [7-0] are written.

T3CLK[1:0] — Timer 3 Clock Source Select bit.
These bits select one out of four input clock sources for Timer 3. The table is shown as follows:

T3CLK[1]	T3CLK[0]	TMR3 Clock Source
0	0	SYSCLK/8
0	1	SYSCLK/4
1	0	SYSCLK/2
1	1	SYSCLK

TMR3CN2 (A101h) Timer 3 Control Register Two R/W 00000000

	7	6	5	4	3	2	1	0
RD	T3CMP2EN	T3CMP1EN	T3OVF24EN	T3OVF16EN	T3OVF8EN	T3INI	TR3	T3AUTO
WR	T3CMP2EN	T3CMP1EN	T3OVF24EN	T3OVF16EN	T3OVF8EN	T3INI	TR3	T3AUTO

T3CMP2EN — Timer 3 Comparator 2 Enable bit
T3CMP2EN=1 configures TMR3RL register as a comparator and enables the corresponding flag and interrupt. This bit can be set to 1 only if T3AUTO=1. If T3AUTO=0, TMR3RL is always used as the reload register.

T3CMP1EN — Timer 3 Comparator 1 Enable bit
T3CMP1EN=1 enables T3CMP comparator and the corresponding flag and interrupt.

T3OVF24EN — Timer 3 24-bit Overflow Flag Enable bit.
T3OVF24EN=1enables T3OVF24 flag and the corresponding interrupt.

T3OVF16EN — Timer 3 16-bit Overflow Flag Enable Control bit.
T3OVF16EN=1 enables the T3OVF16 flag and the corresponding interrupt.

T3OVF8EN — Timer 3 8-bit Overflow Flag Enable bit.
T3OVF8EN=1 enables T3OVF8 flag and the corresponding interrupt.

T3INI	T3MR Initialization bit. This bit is used to configure the initial content of TMR3. TMR3 can be initialized by the of the writing to TMR3L. T3INI=0 loads TMR3=0 when writing into TMR3L. T3INI=1 loads TMR3=TMR3RL when writing into TMR3L.
TR3	Timer 3 Run Control bit which controls the counting of TMR3. TR3=0 holds TMR3 counting TR3=1 enables TMR3 normal counting
T3AUTO	Timer 3 Auto-reload Mode Enable bit. T3AUTO=0 configures TMR3 in Timer 3 in FREE-RUN mode T3AUTO=1 configures TMR3 in AUTO-RELOAD mode

TMR3L (A102h) Timer 3 Counter Low Byte Register RO/W 0000000

	7	6	5	4	3	2	1	0
RD	TMR3[7-0]							
WR	-							

The TMR3L register contains the low byte of the 24-bit Timer 3. The count value is read-only and when read it reflects the current count value. The counter can be initialized by a write operation into this location and TMR3 may be loaded with either value 0 or TMR3RL depending on T3INIT setting.

TMR3M (A103h) Timer 3 Counter Middle Byte Register RO 0000000

	7	6	5	4	3	2	1	0
RD	TMR3[15-7]							
WR	-							

The TMR3M register contains the middle byte of the 24-bit Timer 3. The reading reflects the current counter value.

TMR3H (A104h) Timer 3 Counter High Byte Register RO 0000000

	7	6	5	4	3	2	1	0
RD	TMR3[24-16]							
WR	-							

The TMR3M register contains the high byte of the 24-bit Timer 3. The reading reflects the current counter value.

TMR3RLL (A105h) Timer 3 Reload and Comparison 2 Register Low Byte R/W 0000000

	7	6	5	4	3	2	1	0
RD	TMR3RL[7-0]							
WR	TMR3RL[7-0]							

This register is an auto-reload register which can also be used as a compare 2 register.

TMR3RLM (A106h) Timer 3 Reload and Comparison Register Middle Byte R/W 0000000

	7	6	5	4	3	2	1	0
RD	TMR3RL[15-8]							
WR	TMR3RL[15-8]							

This register is an auto-reload register and can be used as a compare 2 register.

TMR3RLH (A107h) Timer 3 Reload and Comparison Register High Byte R/W 0000000

	7	6	5	4	3	2	1	0
RD	TMR3RL[24-16]							
WR	TMR3RL[24-16]							

This register is an auto-reload register and can be used as a compare 2 register.

TMR3CMPL (A018h) Timer 3 Comparison Register Low Byte R/W 00000000

	7	6	5	4	3	2	1	0
RD	TMR3CMP[7-0]							
WR	TMR3CMP[7-0]							

TMR3CMPM (A019h) Timer 3 Comparison Register Middle Byte R/W 00000000

	7	6	5	4	3	2	1	0
RD	TMR3CMP[15-8]							
WR	TMR3CMP[15-8]							

TMR3CMPH (A01Ah) Timer 3 Comparison Register High Byte R/W 00000000

	7	6	5	4	3	2	1	0
RD	TMR3CMP[24-16]							
WR	TMR3CMP[24-16]							

In Timer 3, there are two snapshot registers that hold the snapshot counting value of TMR3. The triggering of snapshot can be configured as either hardware or software-driven. TMR3SSR0 can be triggered either by software or by the INT0 interrupt event. TMR3SSR1 can be triggered either by software or by INT1 interrupt event.

TMR3SSC (0xA117) Timer 3 Software Snapshot Control Register R/W (0b0000XXXX)

	7	6	5	4	3	2	1	0
RD	-	-	SSEN0	SSEN1	-	-	-	-
WR	SSEN0TG	SSEN1TG	SSEN0	SSEN1	-	-	-	-

SSEN0TG Software Trigger of TMR3 Snapshot 0.
When the bit is set to 1 and SSEN0=1, the snapshot of TMR3 to TMR3SR0 is triggered.
This bit is write-only and self-cleared by hardware.

SSEN1TG Software Trigger of TMR3 Snapshot 0.
When the bit is set to 1 and SSEN1=1, the snapshot of TMR3 to TMR3SR1 is triggered.
This bit is write-only and self-cleared by hardware.

SSEN0 Enable Control of Software Trigger of Snapshot 0.
SSEN0=1 enables the software triggering using SSEN0TG and disables the INT0 hardware triggering of snapshot 0.
SSEN0=0 enables the INT0 hardware snapshot.

SSEN1 Enable Control of Software Trigger of Snapshot 1.
SSEN1=1 enables the software triggering using SSEN1TG and disables the INT1 hardware triggering of snapshot 1.
SSEN1=0 triggers the INT1 hardware snapshot.

TMR3SR0L (A028h) Timer 3 INT0 Triggered Snapshot Register Low Byte RO 00000000

	7	6	5	4	3	2	1	0
RD	TMR3SR0[7-0]							
WR	-							

TMR3SR0M (A029h) Timer 3 INT0 Triggered Snapshot Register Middle Byte RO 00000000

	7	6	5	4	3	2	1	0
RD	TMR3SR0[15-8]							
WR	-							

TMR3SR0H (A02Ah) Timer 3 INT0 Triggered Snapshot Register High Byte RO 00000000

	7	6	5	4	3	2	1	0
RD	TMR3SR0[23-16]							
WR	-							

TMR3SR1L (A02Bh) Timer 3 INT1 Triggered Snapshot Register Low Byte RO 00000000

	7	6	5	4	3	2	1	0
RD	TMR3SR1[7-0]							
WR	-							

TMR3SR1M A02Ch) Timer 3 INT1 Triggered Snapshot Register Middle Byte RO 00000000

	7	6	5	4	3	2	1	0
RD	TMR3SR[15-8]							
WR	-							

TMR3SR1H (A02Dh) Timer 3 INT1 Triggered Snapshot Register High Byte RO 00000000

	7	6	5	4	3	2	1	0
RD	TMR3SR1[24-16]							
WR	-							

1.13 Multiplication and Division Unit (MDU)

MDU provides acceleration on unsigned integer operations of 16-bit multiplications, 32-bit division, and shifting and normalizing operations. The following table shows the execution characteristics of these operations. The MDU does not contain the operation completion status flag. Therefore the most efficient utilization of MDU uses NOP delay for the required clock time of the MDU operation types. The number of the clock cycles required for each operation is shown in the following table and it is counted from the last write of the writing sequence.

Operations	Result	Reminder	# of Clock Cycle
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 the operation is controlled by 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 MDx being written before the previous operation completes. MDEF is automatically cleared after reading ARCON.

MDOV — MDU Overflow Flag bit. MDOV is set by hardware if dividend is zero or the result of multiplication is greater than 0x0000FFFFh

SLR — Shift Direction Control bit. SLR = 1 indicates a shift to the right and SLR =0 indicates a shift to the left.

SC4-0 — Shift Count Control and Result bit. If SC0-4 is written with 00000, the normalization operation 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 a non-zero value, then the shift operation is performed by MDU with the number of shift specified by SC4-0 value.

MD0 (0xF9) MDU Data Register 0 R/W 00000000

	7	6	5	4	3	2	1	0
RD	MD0[7-0]							
WR	MD0[7-0]							

MD1 (0xFA) MDU Data Register 1 R/W 00000000

	7	6	5	4	3	2	1	0
RD	MD1[7-0]							
WR	MD1[7-0]							

MD2 (0xFB) MDU Data Register 2 R/W 00000000

	7	6	5	4	3	2	1	0
RD	MD2[7-0]							
WR	MD2[7-0]							

MD3 (0xFC) MDU Data Register 3 R/W 00000000

	7	6	5	4	3	2	1	0
RD	MD3[7-0]							
WR	MD3[7-0]							

MD4 (0xFD) MDU Data Register 4 R/W 00000000

	7	6	5	4	3	2	1	0
RD	MD4[7-0]							
WR	MD4[7-0]							

MD5 (0xFE) MDU Data Register 5 R/W 00000000

	7	6	5	4	3	2	1	0
RD	MD5[7-0]							
WR	MD5[7-0]							

MDU operation consists of three phases.

Loading MD0 to MD5 data registers in an appropriate order depending on the operation.

Execution of the operation.

Reading result from MD0 to MD5 registers.

The following list shows the MDU read and write sequences. Each operation has its unique writing sequence and reading sequence of MD0 to MD5 registers therefore a precise access sequence is required.

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

Follow the following write-sequence. The first write of MD0 resets the MDU and initiates the MDU error flag mechanism. The last write incites calculation of MDU.

Write MD0 with Dividend LSB byte

Write MD1 with Dividend LSB+1 byte

Write MD2 with Dividend LSB+2 byte (ignore this step for 16-bit divide by 16-bit)

Write MD3 with Dividend MSB byte (ignore this step for 16-bit divide by 16-bit)

Write MD4 with Divisor LSB byte

Write MD5 with Divisor MSB byte

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

Read MD0 with Quotient LSB byte

Read MD1 with Quotient LSB+1 byte

Read MD2 with Quotient LSB+2 byte (ignore this step for 16-bit divide by 16-bit)

Read MD3 with Quotient MSB byte (ignore this step for 16-bit divide by 16-bit)

Read MD4 with Remainder LSB byte

Read MD5 with Remainder MSB byte

Read ARCON to determine error or overflow condition

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

1.13.2 Multiplication – 16-bit multiply by 16-bit

Follow the following write sequence.

Write MD0 with Multiplicand LSB byte
Write MD4 with Multiplier LSB byte
Write MD1 with Multiplicand MSB byte
Write MD5 with Multiplier MSB byte

Then follow the following read sequence.

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

1.13.3 Normalization – 32-bit

Normalization is obtained with integer variables 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 restored on MD0 to MD3. The number of the shift of the normalization can be used as exponents. The following write sequence should be followed. The last write to ARCON initiates the normalization operations by MDU.

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

Then follow the following read sequence.

Read MD0 with Result LSB byte
Read MD1 with Result LSB+1 byte
Read MD2 with Result LSB+2 byte
Read MD3 with Result MSB byte
Read SC[4-0] from ARCON for normalization count or error flag

1.13.4 Shift – 32-bit

Shift is done with integer variables 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 initiates the normalization operations by MDU.

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

Then follow the following read sequence.

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

1.13.5 MDU Flag

The error flag (MDEF) of MDU indicates improperly performed operations. The error mechanism starts at the first MD0 write and finishes with the last read of MD result register. MDEF is set if current operation is interrupted or restarted by improper write of MD register before the operation completes. MDEF is cleared if the operations and proper write/read sequences successfully complete. The overflow flag (MDOV) of MDU indicates an error of operations. MDOV is set if

the divisor is zero

Multiplication overflows

Normalization operation is performed on already normalized variables (MD3.7 =1)

1.14 Serial Port – UART0

UART0 is full duplex and fully compatible with the standard 8052 UART. The receive path of the UART0 is double-buffered that can commence reception of second byte before previously received byte is read from the receive register. Writing to SBUF0 loads the transmit register while reading SBUF0, reads a physically separate receive register. The UART0 can operate in four modes: one synchronous (Mode 0) and three asynchronous modes (Mode 1, 2, and 3). Mode 2 and Mode 3 share a special provision for multi-processor communications. This feature is enabled by setting SM2 bit in SCON0 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: 1 defines an address byte whereas 0 defines a data byte. When SM2 is set to 1, no slave can be interrupted by a data byte. An address byte can interrupt slaves. The addressed slave clears its SM2 bit and prepares to receive the following incoming data bytes. The slaves that are not addressed leave their SM2 set and ignore the incoming data. The UART0-related registers are SBUF0, SCON0, PCON, IE, and IP.

SCON0 (0x98h) UART0 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, SM1 UART Operation Mode

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 1 or Timer 2 overflow rate. This is selected in T2CON registers.
2	1	0	9-Bit UART Mode, fix baud rate Baud rate = CPUCLK/64 (PCON.SMOD0 = 0) or CPUCLK/32 (PCON.SMOD0 = 1)
3	1	1	9-Bit UART Mode, variable baud rate Baud rate = Timer 1 or Timer 2 overflow rate. This is selected in TCON registers.

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

REN Set REN=1 to enable UART PMM switch back function. REN=0 will disable this function. In PMM mode, if REN=1, then any transition on RX of UART will trigger the exit of PMM mode into normal mode.

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.

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) UART0 Data Buffer Register

	7	6	5	4	3	2	1	0
RD	RB[7-0]							
WR	TB[7-0]							

SBUF0 is used for both transmit and receive. Writing a data byte into SBUF0 put this data in UART0's transmit buffer and starts a transmission. Reading a byte from SBUF reads data from the UART0's receive buffer.

1.14.1 Mode 0

Mode 0 is a simple synchronous shift register mode. TXD0 outputs the shift clock which is fixed at CPUCLK/12. RXD0 is a bidirectional I/O port that serves as a data-shifting port. To utilize this mode, TXD0 pin must be enabled as an output pin, while RXD0 needs to be configured as an open-drain type of I/O port. The shift data changes at the rising edge of the shift clock and is valid at the falling edge of the shift clock. The transmission starts when a new byte is written in SBUF0 as TI is cleared to 0. When the byte is transmitted, TI is set and the UART0 waits for the 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 UART0.

1.14.2 Mode 1

8-bit UART mode. RXD0 is the serial input and TXD0 is the serial output. To utilize this mode, the corresponding RXD0 and TXD0 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.T2CON, TCLK.T2CON. 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.T2CON, or TCLK.T2CON is set, the Timer 2 overflow rate is selected and overwrites the SMOD0 setting.

1.14.3 Mode 2

9-bit UART mode. RXD0 is the serial input and TXD0 is the serial output. To utilize this mode, the corresponding RXD0 and TXD0 pins should be configured correctly. 11-bit data including a Start bit (always 0), 8 data bits, a programmable 9^{th} bit, and a Stop bit (always 1) are transferred. The 9^{th} bit can be configured as a parity bit configured by software through TB8 in SCON0. The received 9^{th} bit can be read from TB8. The software determines 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 in PCON register.

1.14.4 Mode 3

Similar to Mode 2 (9-bit UART mode). RXD0 is the serial input and TXD0 is the serial output. To utilize this mode, the corresponding RXD0 and TXD0 pins should also be configured properly. 11-bit data including a Start bit (always 0), 8 data bits, a programmable 9^{th} bit, and a Stop bit (always 1) are transferred. The 9^{th} bit can serve as a parity bit configured by software through TB8 in SCON0. The received 9^{th} bit can be read from TB8. The software determines the correctness of the parity check. The mechanism of the baud rate control in Mode 3 is similar to which in Mode 1 that is determined by Timer 1 or Timer 2 overflow and is set by SMOD0, and T2CON.

1.15 I²C Master

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	I2CMTP[7-0]							
WR	I2CMTP[7-0]							

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. The minimum value of I2CMTP[7-0] shall be 0x01. In other words, the maximum SCL frequency is system clock divide by 16.

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

	7	6	5	4	3	2	1	0
RD	SA[6-0]							RS
WR	SA[6-0]							RS

SA[6-0] Slave Address. SA[6-0] defines the slave address the I²C master uses to communicate.
RS Receive/Send Bit. RS determines if the following operation will be a RECEIVE (RS=1) or SEND (RS=0).

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

	7	6	5	4	3	2	1	0
RD	RD[7-0]							
WR	TD[7-0]							

I2CMBUF functions as a transmit data register when written and as a receive data register when read. When written, TD will be sent on the bus by the next SEND or BURST SEND operations. TD[7] is sent first. When read, RD 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	I2CMRST	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.

I2CMRST Writing 1 to this bit forces the I2CM to perform reset and clear its internal state machine. At the end of the initialization, all SFRs will return to the default value. This bit is cleared automatically by hardware.
INFILEN Input Noise Filter Enable. When IFILEN is set, pulses shorter than 50 nsec on inputs of SDA and SCL are filtered out.
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.

START, STOP, RUN and HS, RS, ACK bits are used to set the actually I²C Master to initiate and terminate a transaction. 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.

2. FLASH CONTROLLER

The flash controller connects the CPU to the on-chip embedded FLASH memory. The FLASH memory functions 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 sends commands to the FLASH controller through the XFR registers. And when the FLASH controller processes these commands, CPU is held idle until the command is executed. The embedded Flash memory contains two blocks – Main Memory and Information Block (IFB). The Main Memory is 128KX8 with uniform 1024 Byte page (sector) size. The Information Block is 256 Byte and sits in a separate sector.

The commands performed by a Flash Controller are defined in FLSHCMD registers. The defined operations allow the user program to use on-chip flash as a program memory, and a non-volatile data memory in In-System-Programming as well as In-Application-Programming. The maximum flexibility of the on-chip flash memory can be achieved through user program. The manufacturer provides a default ISP boot program located on the top sectors of the flash. The preset ISP boot program can be used or modified or replaced based on application requirements

FLSHCMD (A020h) Flash Controller Command Register R/W 10000000 TB Protected

	7	6	5	4	3	2	1	0
RD	WRVFY	BUSY	FAIL	CMD4	CMD3	CMD2	CMD1	CMD0
WR	CYC[2-0]			CMD4	CMD3	CMD2	CMD1	CMD0

WRVFY Byte Write Result Verify. At the end of a write cycle, hardware reads back the data and compares it with which should be written to the flash. If there is a mismatch, this bit represents 0. It is reset to 1 by hardware when another ISP command is executed.

BUSY Flash command is in processing. This bit indicates that Flash Controller is executing the Flash Read, Write, or Sector Erase and other commands are not valid.

FAIL Command Execution Result. It is set if the previous command execution fails due to any reasons. It is recommended that the program should verify the command execution after issuing a command to the Flash controller. It is not cleared by reading but when a new command is issued. Possible causes of FAIL include address over range, or address falls into protected region.

CYC[2-0] Flash Command Time Out

CYC[2-0] defines command time out cycle count. Cycle period is defined by ISPCLK, which is SYSCLK/256/(ISPCLKF[7-0]+1). The number of cycles is tabulated as following.

CYC[2-0]			WRITE	ERASE
0	0	0	55	5435
0	0	1	60	5953
0	1	0	65	6452
0	1	1	69	6897
1	0	0	75	7408
1	0	1	80	7906
1	1	0	85	8404
1	1	1	89	8889

For normal operations, CYC[2-0] should be set to 111.

CMD4 – CMD0 Flash Command

These bits define commands for the Flash controller. The valid commands are listed in the following table. Any invalid command will not be executed and returned with FAIL bit set.

CMD4	CMD3	CMD2	CMD1	CMD0	COMMAND
1	0	0	0	0	Main Memory Byte Read
0	1	0	0	0	Main Memory Sector Erase
0	0	1	0	0	Main Memory Sector Byte Write
0	0	0	1	0	IFB Byte Read
0	0	0	0	1	IFB Byte Write
0	0	0	1	1	-
1	0	0	1	0	-

For all commands, the address of the flash is composed from FLSHADM:FLSHADH:FLSHADL and the data is referred at FLSHDAT registers. The erase command operation is sector-based, the address of the sector is determined from the high order address bits. For example, to point to the sector of 0x0C000-0xCFFF, the upper 8 bits "0C" are used. And the erase command erases the whole addressed sector contents. For Erase and Write command, the Flash Controller also checks if the destination address falls within the protection zone defined by CNTPCTL and CNTPCTH registers. If it is protected, the Flash Controller does not execute the command and return with FAIL result bit. For IFB Byte-write, the Flash Controller does not execute the command and return with Fail result bit if the byte address falls into manufacturer data range. Please also note the Fuse block is used for manufacturer to store manufacturing related and calibration data and thus can only be read and not writable or erasable. Fuse block can only be erased or written under writer mode.

ISPCLKF (A024h) Flash Command Clock Scaler R/W 00100101 TB Protected

	7	6	5	4	3	2	1	0
RD	ISPCLKF[7-0]							
WR	ISPCLKF[7-0]							

ISPCLKF[7-0] configures the clock time base for generation of Flash erase and write timing. ISPCLK = SYSCLK * (ISPCLKF[7-0]+1)/256. For correct timing, ISPCLK should be set to approximately at 2MHz.

FLSHDAT (A021h) 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							

FLSHADL (A023h) 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 ADDR[7-0]							
WR	Flash Address Low Byte Register ADDR[7-0]							

FLSHADH (A022h) Flash Controller High Address Data Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	Flash Address High Byte Register ADDR[15-8]							
WR	Flash Address High Byte Register ADDR[15-8]							

FLSHADM (A012h) Flash Controller MSB Address Data Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	ADDR[16]
WR	-	-	-	-	-	-	-	ADDR[16]

A very common problem of embedded flash memory is when being used as both data and program storage which leads to content loss due to software or other problems caused by program flow or noise. It induces executions of modifying stored contents. The design of Flash controller takes into considerations of these events and provides further protection to avoid accidental erasure or modifications of critical information or software codes. When a command is sent to the Flash Controller through FLSHCMD register, the controller 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 message. Two protections zones are defined by 00000 to CNTPCTL and CNTPCTH to 1FFFF.

CNTPCTL (A025h) Flash Content Protection Low Zone 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 00000h of the flash which is protected against erasure or modifications. The data is processed in 512 Byte increments. The protected region is defined from 00 to (CNTPCTL1 -1). Note that CNTPCTL defaults to FFh which protects the whole 128KB of flash memory. User program needs to write the appropriate data into CNTPCTL to enable erase and write access.

CNTPCTH (A026h) Flash Content Protection High Zone 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 1FFFFh of the flash to be protected against erasure or modifications. The data is processed in the increments of 512 Byte. The protected region is defined from (CNTPCTH+1) to FF. Note that CNTPCTH defaults to 00h which means the protection of 128KB of flash memory is on. User program needs to write the appropriate data into CNTPCTH by reading IFB-protected information to protect boot code and expand the protection zone under application considerations.

There is an additional content protection against internal program. This protects sensitive data from unauthorized access. The protection range is from 0x2000 to 0x1FFFF of embedded flash memory. The protection is achieved by two special registers, INTPCT1 (0xA013) and INTPCT2 (0xA014). After any reset condition such as power-up, RSTN, LVR, or WDT reset, INTPCT1 and INTPCT2 are initialized to 0x00. The bits in INTPCT1 and INTPCT2 can only be written to "1". When the embedded flash memory has been protected, this means accessing this protected range returns with 0x00 either by program instruction such as "MOVC" or by Flash Main Memory Byte Read access. The internal protection is by default not turned on after reset because both INTPCT1 and INTPCT2 are 0x00. Both registers are protected by TB. To turn on the internal protection, INTPCEN must be enabled by writing a "0x80" into enable the Internal Protection function. When INTPCEN is set, user can define the protection range by program INTPCT1 or INTPCT2. The INTPCEN will be cleared and stuck-on zero after setting any protection range. In other words, the protection range is single time programmable. Once the protection is turned on, it can't be turned off or modified because INTPCEN is stuck-on zero. To restore unprotected state, the chip must go through a reset. The internal protection should be enabled with extreme cautiousness. It is important that once it is turned on, program execution should not reach the protected zone, otherwise unpredicted program errors may occur.

INTPCT1 (A013h) Internal Protection Enable Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	INTPCT1[7-0]							
WR	INTPCT1[7-0]							

This register can be written to "1" only. Writing "0" into any bit of this register does not alter the content. This register is cleared to 0x00 after reset. And the value of this register can be cleared only by a reset.

INTPCT2 (A014h) Internal Protection Enable Register R/W 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	INTPCEN	INTPCT2[6-0]						
WR	INTPCEN	INTPCT2[6-0]						

This register can be written to "1" only. Writing "0" into any bit of this register does not alter the content. This register is set to 0x00 after a reset. And the value of this register can be set only by a reset.

INTPCEN INTPCT1/INTPCT2 updates enable. This bit is single time programmable. After setting any protection range, this bit will be stuck-on zero and prohibit another protection updating. Because the INTPCEN is disabled in the initiation, a "0x80" must be programmed into INTPCT2 to enable INTPCEN. When INTPCEN is set, the expected program protection can be defined by setting INTPC1 or INTPC2[6-0].

The following table summarizes the internal program protection with different INCTPCT1 and INTPCT2 settings and protected range from program read access.

Executed Priority	Register	Protected Region of Embedded Flash Memory
1	INTPCT1[0]=1	0x02000 ~ 0x1FFFF
2	INTPCT1[1]=1	0x04000 ~ 0x1FFFF
3	INTPCT1[2]=1	0x06000 ~ 0x1FFFF
4	INTPCT1[3]=1	0x08000 ~ 0x1FFFF
5	INTPCT1[4]=1	0x0A000 ~ 0x1FFFF
6	INTPCT1[5]=1	0x0C000 ~ 0x1FFFF
7	INTPCT1[6]=1	0x0E000 ~ 0x1FFFF
8	INTPCT1[7]=1	0x10000 ~ 0x1FFFF
9	INTPCT2[0]=1	0x12000 ~ 0x1FFFF
10	INTPCT2[1]=1	0x14000 ~ 0x1FFFF
11	INTPCT2[2]=1	0x16000 ~ 0x1FFFF
12	INTPCT2[3]=1	0x18000 ~ 0x1FFFF
13	INTPCT2[4]=1	0x1A000 ~ 0x1FFFF
14	INTPCT2[5]=1	0x1C000 ~ 0x1FFFF
15	INTPCT2[6]=1	0x1E000 ~ 0x1FFFF

3. I²C Slave Controller 1 (I2CS1)

The I²C Slave Controller 1 is a regular I²C Slave controller with enhanced functions such as clock-stretching and programmable hold time. These enhancements provide significant improvement on compatibilities. I2CS1 shares the SCL/SDA pins with the I2CM1. I2CS1 also can be configured to respond to two I²C addresses – I2CADR1 and I2CADR3. These two addresses can be enabled separately.

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 either forces 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 occurs, 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 the new byte is written to the I2CSDAT. If TXBI is not cleared, it indicates lack of new data and the slave controller holds SCL line low to stretch the current clock cycle if CLKSTREN is set. If the clock stretching is not enabled, the slave controller takes the old byte into the shift register and replies with NACK, thus causing 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 releases the data line to allow the master to generate a STOP or REPEAT START.

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 spikes under 1/2 EPPCLK period on the input of SDA and SCL lines are filtered out. If INFILEN is low, no input filtering is done. The following registers are related to I²C Slave Controller.

I2CSCON1 (0xEB) I2CS1 Configuration Register R/W (0x00)

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

I2CSRST	I²C Slave Reset bit. Set this bit causes the Slave Controller to reset all internal state machine. Clear this bit for normal operations. Setting this bit clears the I2CSADR1 (I²C slave address x).
EADDRMI	ADDRMI interrupt Enable bit. Set this bit to set 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 to set STOPI interrupt as the I²C slave interrupt.
ERPSTARTI	RPTSTARTI Interrupt Enable Bit. Set this bit to set RPTSTARTI interrupt as the I²C slave interrupt.
ETXBI	TXBI Interrupt Enable bit Set this bit to allow TXBI interrupt as the I²C slave interrupt.
ERCBI	RCBI Interrupt Enable bit. Set this bit to allow RCBI interrupt as the I²C slave interrupt.
CLKSTREN	Clock Stretching Enable bit. Set to enable the clock stretching function of the slave controller. Clock stretching is an optional feature defined in I²C specification. If the clock stretching option is enabled (for slave I²C), the data written into transmit buffer is shifted out only after the occurrence of clock stretching, and the data cannot be loaded to transmit shift register. The programmer must write the same data again to the transmit buffer.
INFILEN	Input Noise Filter Enable bit. Set this bit to enable the input noise filter of SDA and SCL lines. When the filter is enabled, it filters out the spike of less than 50nsec.
START	Start Condition. This bit is set when the slave controller detects 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 read-only bit is cleared when STOP condition is detected.
XMT	This bit is set by the controller when the I²C slave is in transmit operation; is clear when the I²C slave controller is in receive operation.

I2CSST1 (0xEC) I2CS1 Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	FIRSTBT	ADDRMI	STOPI	RPSTARTI	TXBI	RCBI	SADR3M	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 as the first byte received after address match. This bit is cleared after the first byte of the transaction is read. The bit is read only and generated by the slave controller.

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

STOPI — Stop Condition Interrupt Flag bit. 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 bit. 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.

RCBI — Receiver Buffer Interrupt Flag bit. This bit is set when the slave controller puts new data in the I2CSDAT and ready for software-reading. This bit is cleared after the software reads I2CSDAT.

SARD3M — Slave Address Match Flag bit. This bit is meaningful only when SDDRMI is set.
SARD3M=0 indicates the received I²C address matches with I2CSADR1.
SARD3M=1 indicates the received I²C address matches with I2CSADR3.
This bit is cleared when ADDRMI is cleared.

NACK — NACK Condition bit. This bit is set when the host responds 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 does not upload new data into the shift register. And the slave transmits the old data again as the next transfer, and this re-transmission continues if NACK is repeated until the transmission is successful and returned with ACK. This bit is cleared when a new ACK is detected or it can be cleared by software.

HOLDT[3-0] — These four bits define the hold time of the peripheral clock (EPPCLK) cycles between SDA to SCL. The I²C specification requires for minimum of 300nsec hold time, so the condition of "TEPPCLK*(HOLDT[3:0]+3) ≧300nsec hold time" equation must be met. For example, if the peripheral clock cycle (EPPCLK) is 20MHz, then HOLD[3-0] should be set to ≧3.

I2CSADR1 (0xED) I2CS1 Slave Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	I2CSEN	ADDR[6-0]						

I2CSEN — Set this bit to enable the I²C slave controller and ADDR[6-0] for address matching

ADDR[6-0] — 7-bit slave address.

I2CSDAT1 (0xEE) I2CS1 Data Register R/W (0x00)

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

I2CSADR3 (0x9E) I2CS1 2nd Slave Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	I2CSEN	ADDR[6-0]						

I2CSEN — Set this bit to enable the I²C slave controller and ADDR[6-0] for address matching. Please note this can coexist with I2CSADR1.

ADDR[6-0] — 7-bit slave address.

4. I²C Slave Controller 2 (I2CS2)

The I²C Slave Controller 2 has dual functions – as a debug port for communication with host or as a regular I²C slave port. Please note both functions can coexist. Also the I²C Slave controller supports the clock stretching functions.

The debug accessed by the host is through I²C slave address defined by SI2CSDBGID register and enabled by DBGSI2C2EN=1. When I2CS2 received this address match, a DBG interrupt is generated. This is described in the Debug and ISP sections. If DBGSI2C2EN=0, then I2CS2 functions as a regular I²C slave. The address of the slave is set by I2CSADR2 register. The MSB in I2CSADDR2 is the enable bit for the I²C slave controller and I2CSADR2[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 either forces 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 occurs, 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 the new byte is written to the I2CSDAT. If TXBI is not cleared, it indicates lack of new data and the slave controller holds SCL line low to stretch the current clock cycle if CLKSTREN is set. If the clock stretching is not enabled, the slave controller takes the old byte into the shift register and replies with NACK, thus causing 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 releases the data line to allow the master to generate a STOP or REPEAT START.

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 spikes under 1/2 EPPCLK period on the input of SDA and SCL lines are filtered out. If INFILEN is low, no input filtering is done. The following registers are related to I²C Slave Controller.

I2CSCON2 (0xDB) I2CS2 Configuration Regiter 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 causes the Slave Controller to reset all internal state machine. Clear this bit for normal operations. Setting this bit clears the I2CSADR2 (I²C slave address x).
EADDRMI	ADDRMI interrupt Enable bit. Set this bit to set 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 to set STOPI interrupt as the I²C slave interrupt.
ERPSTARTI	RPTSTARTI Interrupt Enable Bit. Set this bit to set RPTSTARTI interrupt as the I²C slave interrupt.
ETXBI	TXBI Interrupt Enable bit Set this bit to allow TXBI interrupt as the I²C slave interrupt.
ERCBI	RCBI Interrupt Enable bit. Set this bit to allow RCBI interrupt as the I²C slave interrupt.
CLKSTREN	Clock Stretching Enable bit. Set to enable the clock stretching function of the slave controller. Clock stretching is an optional feature defined in I²C specification. If the clock stretching option is enabled (for slave I²C), the data written into transmit buffer is shifted out only after the occurrence of clock stretching, and the data cannot be loaded to transmit shift register. The programmer must write the same data again to the transmit buffer.
INFILEN	Input Noise Filter Enable bit. Set this bit to enable the input noise filter of SDA and SCL lines. When the filter is enabled, it filters out the spike of less than 50nsec.
XMT	This bit is set by the controller when the I²C slave is in transmit operation; is clear when the I²C slave controller is in receive operation.

I2CSST2 (0xDC) I2CS2 Status Register 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 as the first byte received after address match. This bit is cleared after the first byte of the transaction is read. The bit is read only and generated by the slave controller.

ADDRMI — Slave Address Match Interrupt Flag bit. This bit is set when the received address matches the address defined in I2CSADR2. If EADDMI is set, this generates an interrupt. This bit must be cleared by software.

STOPI — Stop Condition Interrupt Flag bit. 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 bit. 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.

RCBI — Receiver Buffer Interrupt Flag bit. This bit is set when the slave controller puts new data in the I2CSDAT and ready for software-reading. This bit is cleared after the software reads I2CSDAT.

START — Start Condition. This bit is set when the slave controller detects 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 read-only bit is cleared when STOP condition is detected.

NACK — NACK Condition. This bit is set when the host responds 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 does not upload new data into the shift register. And the slave transmits the old data again as the next transfer, and this re-transmission continues if NACK is repeated until the transmission is successful and returned with ACK. This bit is cleared when a new ACK is detected or it can be cleared by software.

HOLDT[3-0] — These four bits define the hold time of the peripheral clock (EPPCLK) cycles between SDA to SCL. The I²C specification requires for minimum of 300nsec hold time, so the condition of "TEPPCLK*(HOLDT[3:0]+3) ≧300nsec hold time" equation must be met. For example, if the peripheral clock cycle (EPPCLK) is 20MHz, then HOLD[3-0] should be set to ≧3.

I2CSADR2 (0xDD) I2CS2 Slave Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	-
WR	I2CSEN	ADDR[6-0]						

I2CSENT — Set this bit to enable the I²C slave controller.

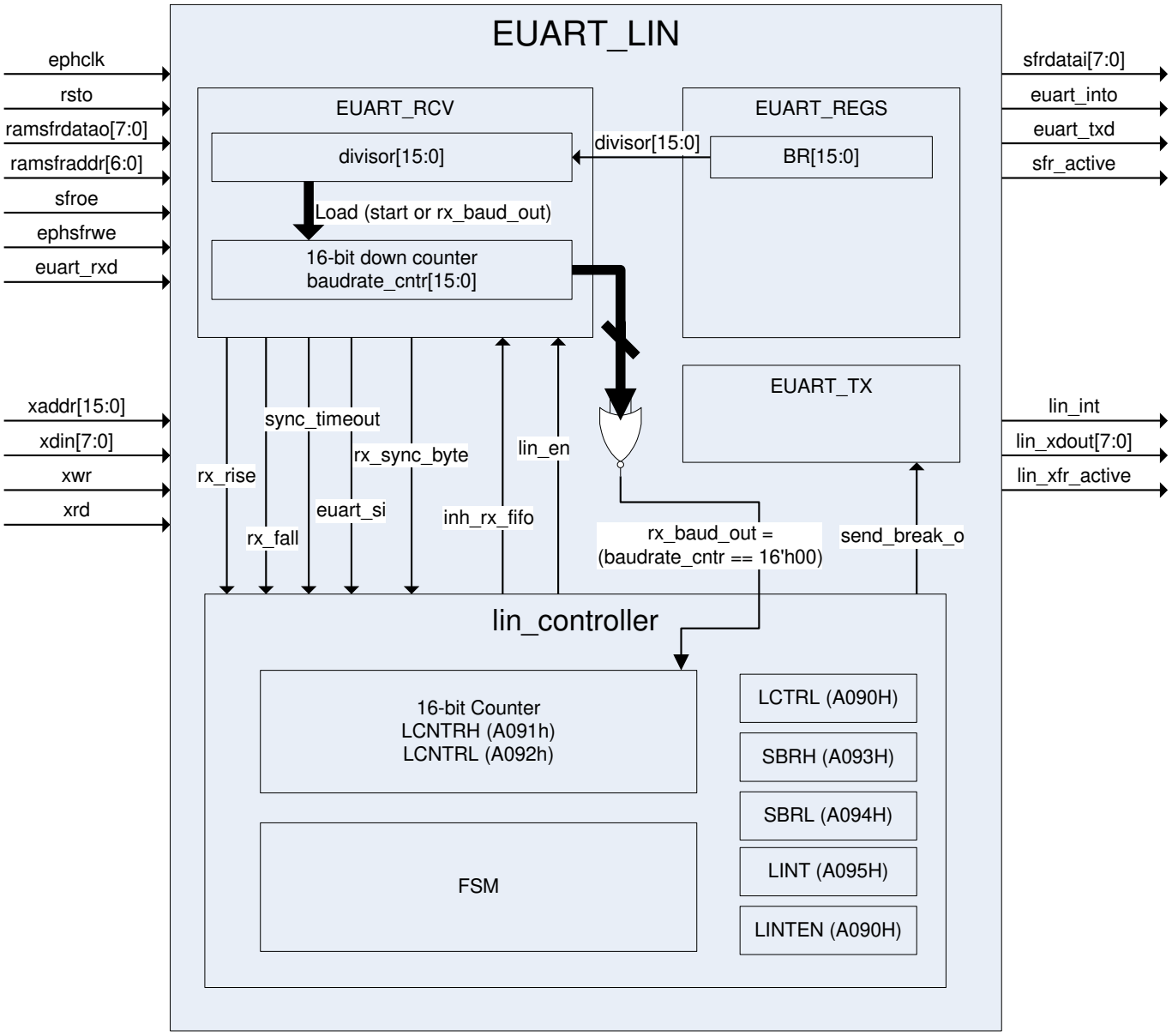
ADDR[6-0] — 7-bit slave address.

I2CSDAT2 (0xDE) I2CS2 Data Register R/W (0x00)

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

5. EUART2 with LIN Controller (EUART2)

LIN-capable 16550-like EUART2 is an enhanced UART controller (EUART) with separate transmit and receive FIFO. Both transmit and receive FIFO are 15-bytes deep and can be parameterized for interrupt triggering. The addition of FIFO significantly reduces the CPU load to handle high-speed serial interface. Transmit FIFO and receive FIFO have respective interrupt trigger levels that can be set based on optimal CPU performance adjustment. The EUART2 also has dedicated 16-bit Baud Rate generator and thus provides accurate baud rate under wide range of system clock frequency. The EUART2 also provides LIN extensions that incorporate message handling and baud-rate synchronization. The block diagram of EUART2 is shown in the following.



The following registers are used for configurations of and interface with EUART2.

SCON2 (0xC2) UART2 Configuration Register 00000000, R/W

	7	6	5	4	3	2	1	0
RD	EUARTEN	SB	WLS[1]	WLS[0]	BREAK	OP	PERR	SP
WR	EUARTEN	SB	WLS[1]	WLS[0]	BREAK	OP	PE	SP

EUARTEN — Transmit and Receive Enable bit
Set to enable EUART2 transmit and receive functions: To transmit messages in the TX FIFO and to store received messages in the RX FIFO.

SB — Stop Bit Control
Set to enable 2 Stop bits, and clear to enable 1 Stop bit.

WLS[1-0] — The number of bits of a data byte. This does not include the parity bit when parity is enabled.
00 - 5 bits
01 - 6 bits
10 - 7 bits
11 - 8 bits

BREAK — Break Condition Control Bit.
Set to initiate a break condition on the UART interface by holding UART output at low until BREAK bit is cleared.

OP — Odd/Even Parity Control Bit

PE/PERR — Parity Enable / Parity Error status
Set to enable parity and clear to disable parity checking functions. If read, PERR=1 indicates a parity error in the current data of RX FIFO.

SP — Parity Set Control Bit
When SP is set, the parity bit is always transmitted as 1.

SFIFO2 (0xA5) UART2 FIFO Status/Control Register 00000000 R/W

	7	6	5	4	3	2	1	0
RD	RFL[3-0]				TFL[3-0]			
WR	RFLT[33-0]				TFLT[3-0]			

RFL[3-0] — Current Receive FIFO level. This is read only and indicate the current receive FIFO byte count.

RFLT[3-0] — Receive FIFO trigger threshold. This is write-only. RDA interrupt will be generated when RFL[3-0] is greater than RFLT[3-0].

RFLT[3-0]	Description
0000	RX FIFO trigger level = 0
0001	RX FIFO trigger level = 1
0010	RX FIFO trigger level = 2
0011	RX FIFO trigger level = 3
0100	RX FIFO trigger level = 4
0101	RX FIFO trigger level = 5
0110	RX FIFO trigger level = 6
0111	RX FIFO trigger level = 7
1000	RX FIFO trigger level = 8
1001	RX FIFO trigger level = 9
1010	RX FIFO trigger level = 10
1011	RX FIFO trigger level = 11
1100	RX FIFO trigger level = 12
1101	RX FIFO trigger level = 13
1110	RX FIFO trigger level = 14
1111	Reserved

TFL[3-0] — Current Transmit FIFO level. This is read only and indicate the current transmit FIFO byte count.

TFLT[3-0] Transmit FIFO trigger threshold. This is write-only. TRA interrupt will be generated when TFL[3-0] is less than TFLT[3-0].

TFLT[3-0]	Description
0000	Reserved
0001	TX FIFO trigger level = 1
0010	TX FIFO trigger level = 2
0011	TX FIFO trigger level = 3
0100	TX FIFO trigger level = 4
0101	TX FIFO trigger level = 5
0110	TX FIFO trigger level = 6
0111	TX FIFO trigger level = 7
1000	TX FIFO trigger level = 8
1001	TX FIFO trigger level = 9
1010	TX FIFO trigger level = 10
1011	TX FIFO trigger level = 11
1100	TX FIFO trigger level = 12
1101	TX FIFO trigger level = 13
1110	TX FIFO trigger level = 14
1111	TX FIFO trigger level = 15

Receive and transmit FIFO can be reset by clear FIFO operation. This is done by setting BR[11-0]=0 and EUARTEN=0. This also clears RFO, RFU and TFO interrupt flags without writing the interrupt register. The LIN counter LCNTR is also cleared.

SINT2 (0xA7) UART2 Interrupt Status/Enable Register 00000000 R/W

	7	6	5	4	3	2	1	0
RD	INTEN	TRA	RDA	RFO	RFU	TFO	FERR	TI
WR	INTEN	TRAEN	RDAEN	RFOEN	RFUEN	TFOEN	FERREN	TIEN

INTEN Interrupt Enable bit. Write only
Set to enable UART2 interrupt. Clear to disable interrupt. Default is 0.

TRA/TRAEN Transmit FIFO is ready to be filled.
This bit is set when transmit FIFO has been emptied below FIFO threshold. Write “1” to enable interrupt. The flag is automatically cleared when the condition is absent.

RDA/RDAEN Receive FIFO is ready to be read.
This bit is set by hardware when receive FIFO has been filled above the FIFO threshold. Write “1” to enable interrupt. RDA will also be set when RFL < RFLT for bus idle duration longer than RFLT * 16 * Baud Rate. This is to inform software that there are still remaining unread received bytes in the FIFO.
This flag is cleared when RFL < RFLT and writing “0” on the bit (the interrupt is disabled simultaneously).

RFO/RFOEN Receive FIFO Overflow Enable bit
This bit is set when overflow condition of receive FIFO occurs. Write “1” to enable interrupt. The flag can be cleared by software, writing “0” on the bit (the interrupt is disabled simultaneously), or by FIFO reset action.

RFU/RFUEN Receive FIFO Underflow Enable bit
This bit is set when underflow condition of receive FIFO occurs. Write “1” to enable interrupt. The flag can be cleared by software, writing “0” on the bit (the interrupt is disabled simultaneously), or by FIFO reset action.

TFO/TFOEN Transmit FIFO Overflow Interrupt Enable bit
This bit is set when overflow condition of transmit FIFO occurs. Write “1” to enable interrupt. The flag can be cleared by software, writing “0” on the bit (the interrupt is disabled simultaneously), or by FIFO reset action.

FERR/FERREN Framing Error Enable bit
This bit is set when framing error occurs as the byte is received. Write “1” to enable interrupt. The flag must be cleared by software, writing “0” on the bit (the interrupt is disabled simultaneously).

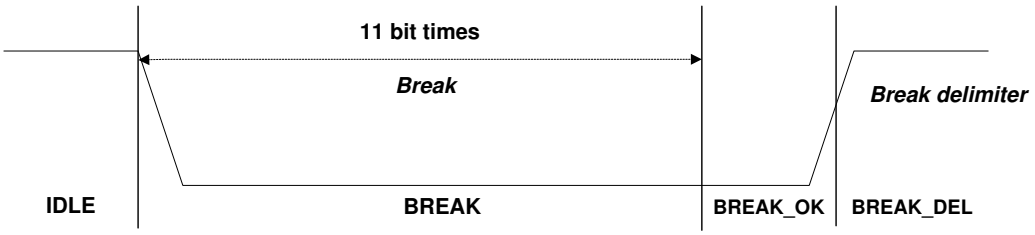
TI/TIEN Transmit Message Completion Interrupt Enable bit
This bit is set when all messages in the TX FIFO are transmitted and thus the TX FIFO becomes empty. Write “1” to enable interrupt. The flag must be cleared by software, writing “0” on the bit (the interrupt is disabled simultaneously).

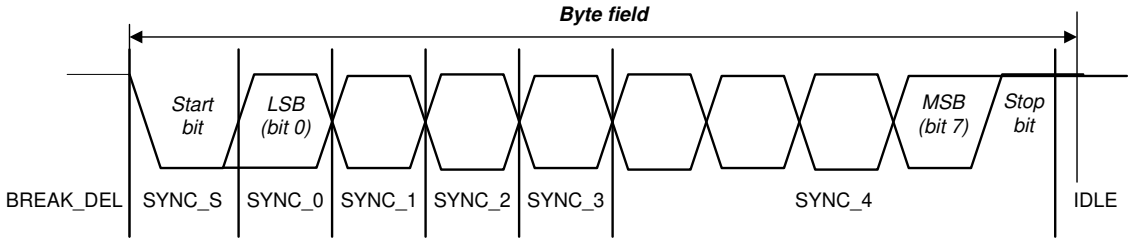
SBUF2 (0xA6) UART2 Data Buffer Register 0x00 R/W

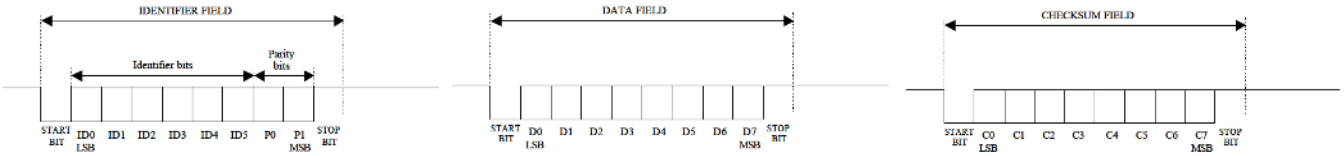
	7	6	5	4	3	2	1	0
RD	EUART2 Receive Data Register							
WR	EUART2 Transmit Data Register							

This register is the virtual data buffer register for both receive and transmit FIFO. When being read, it reads out the top byte of the RX FIFO; when being written, it writes into the top byte of the TX FIFO.

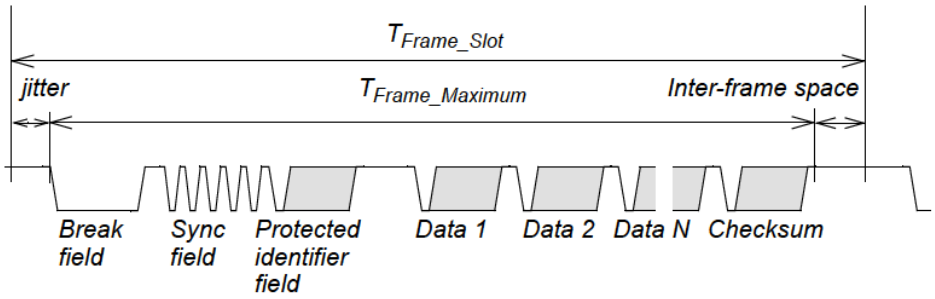
EUART2 can be configured to add LIN capability. The major enhancement of LIN includes master/slave configurations, auto baud-rate synchronization, and frame based protocol with header. Under LIN extension mode, all EUART2 registers and functions are still effective and operational. LIN is a single-wire bus and it requires external components to combine RX and TX signals externally. LIN is frame based and consists of message protocols with master/slave configurations. The following diagram shows the basic composition of a header message sent by the master. It starts with BREAK field, the SYNC field, PID field, DATA bytes, and CRC bytes. The LIN controller can handle the BREAK field and the SYNC field by hardware, for the PID field, DATA bytes and CRC bytes must be processed by software.







A LIN frame structure is shown and the frame time matches the number of bits sent and has a fixed timing.



LIN bus protocol is based on frame. Each frame is partitioned into several parts as shown above. For master to initiate a frame, the software follows the following procedure.

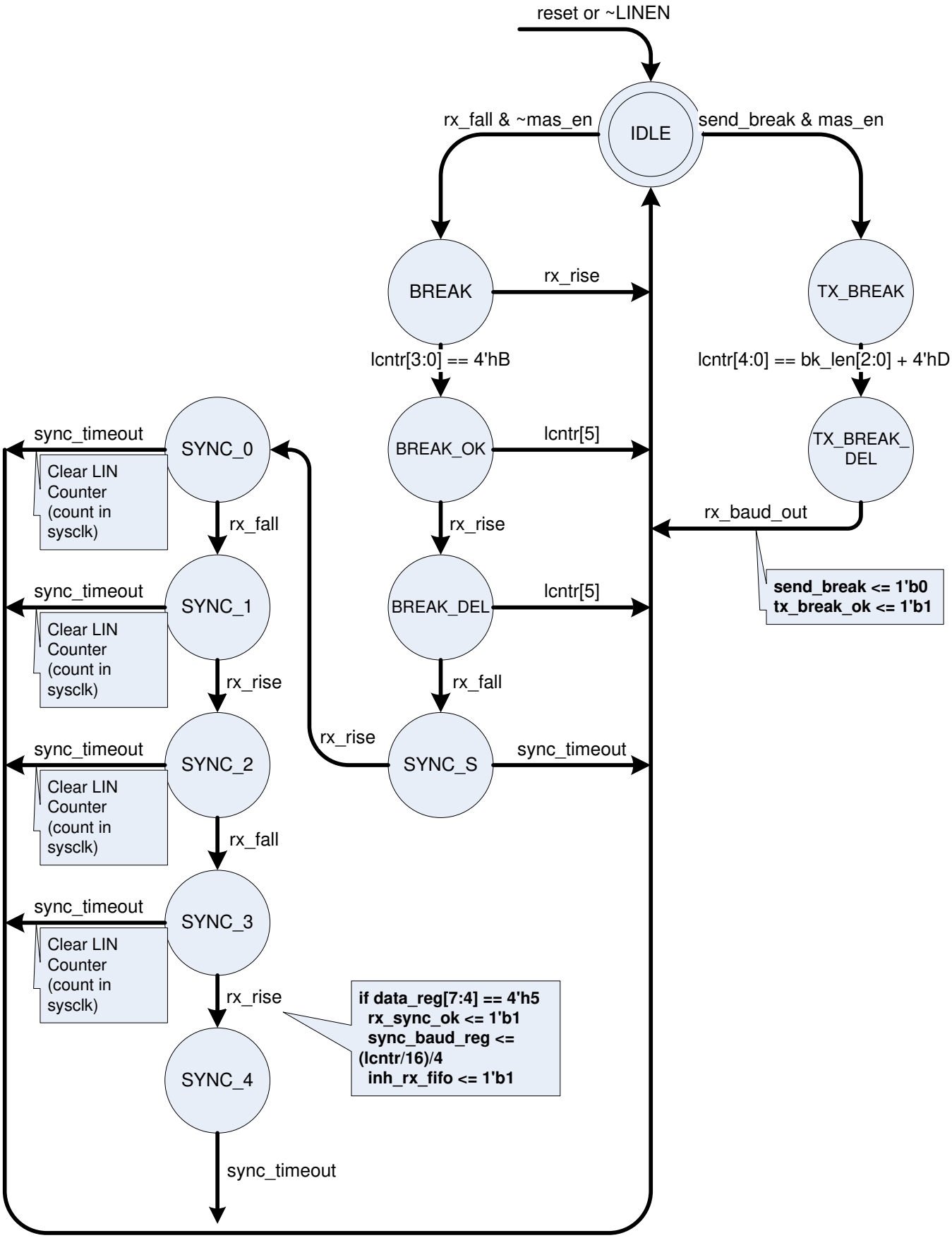
Initiate a SBK command. (SW needs to check if the bus is in idle state, and there is no pending transmit data).

Write “55” into TFIFO.

Write “PID” into TFIFO.

Wait for SBK to complete interrupts and then write the following transmit data if applicable. (This is optional).

The following diagram shows Finite State Machine (FSM) of the LIN extension and is followed by registers within EUART2.


reset or ~LINEN
IDLE
rx_fall & ~mas_en
send_break & mas_en
BREAK
rx_rise
TX_BREAK
lcntr[3:0] == 4'hB
lcntr[4:0] == bk_len[2:0] + 4'hD
BREAK_OK
lcntr[5]
TX_BREAK_DEL
rx_baud_out
rx_rise
send_break <= 1'b0
tx_break_ok <= 1'b1
BREAK_DEL
lcntr[5]
rx_fall
SYNC_S
sync_timeout
rx_rise
SYNC_0
sync_timeout
Clear LIN Counter (count in sysclk)
rx_fall
SYNC_1
sync_timeout
Clear LIN Counter (count in sysclk)
rx_rise
SYNC_2
sync_timeout
Clear LIN Counter (count in sysclk)
rx_fall
SYNC_3
sync_timeout
Clear LIN Counter (count in sysclk)
rx_rise
SYNC_4
if data_reg[7:4] == 4'h5
rx_sync_ok <= 1'b1
sync_baud_reg <= (lcntr/16)/4
inh_rx_fifo <= 1'b1
sync_timeout

LCTRL (0xA090) LIN Status/Control Register 0x00 R/W

	7	6	5	4	3	2	1	0
RD	LINEN	MASEN	ASU	MASU	SBK	BL[2:0]		
WR	LINEN	MASEN	ASU	MASU	SBK	BL[2:0]		

LINEN — LIN Enable (1: Enable / 0: Disable)
LIN header detection / transmission is functional when LINEN = 1.
※ Before enabling LIN functions, the EUART2 registers must be set correctly：0xB0 is recommended for SCON2.

MASEN — Master Enable bit (1: Master / 0: Slave) LIN operating mode selection. This bit is changeable only when LINEN = 0 (must clear LINEN before changing MASEN).

ASU — Auto-Sync Update Enable (1: Enable / 0: Disable), Write Only
If ASU is 1, the LIN controller will automatically overwrite BR[15-0] with SBR[15-0] and issue an ASUI interrupt when received a valid SYNC field.
If ASU is 0, the LIN controller will only notice the synchronized baud rate in SBR[15-0] by issuing an RSI interrupt.
Please note, ASU should not be set under UART mode. ASU capability is based on the message containing BREAK and SYNC field in the beginning.
When ASU=1, the auto sync update is performed on every receiving frame, and is updated frame by frame.

MASU — Message Auto Sync Update Enable.
MASU is meaningful only if ASU=0. MASU=1 will enable the auto sync update on the next received frame only. It is self-cleared when the sync update is completed. The software must set MASU again if another auto sync operation is desired.

SBK — Send Break (1: Send / 0: No send request)
LINEN and MASEN should be set before setting SBK. When LINEN and MASEN are both 1, set SBK to send a bit sequence of 13+BL[2:0] consecutive dominant bits and 1 recessive bit (Break Delimiter). Once SBK is set, this bit represents the “Send Break” status and CANNOT be cleared by writing to “0”; instead, clearing LINEN cancels the “Send Break” action. In normal cases, SBK is cleared automatically when the transmission of Break Delimiter is completed.

BL[2:0] — Break Length Setting
Break Length = 13 + BL[2:0]. Default BL[2:0] is 3’b000.

LINTMRH (0xA091) LIN Timer Register High (0xFF) R/W

	7	6	5	4	3	2	1	0
RD	LCNTR15-8]							
WR	LINTMR[15-8]							

LINTMRL (0xA092) LIN Time Register Low (0xFF) R/W

	7	6	5	4	3	2	1	0
RD	LCNTR[7-0]							
WR	LINTMR[7-0]							

LCNTR[15-0] is read only and is an internal 16-bit counter clocked by the baud rate clock. LINTMR[15-0] is write only and is the timer limit for LCNTR[15-0]. If MASEN=1 as LIN master mode, this timer is used to generate Frame time base. The internal counter LCNTR[15-0] is cleared whenever a “SEND BREAK” command is executed, and when the counter reaches LINTMR [15-0] (LCNTR[15-0] >= LINTMR[15-0]), a LCNTRO interrupt is generated. Thus the software can write a Frame Time value into LINTMR and use interrupts to initiate frames. If MASEN=0 as LIN slave mode, this timer is used for determining the accumulated bus idle time. The internal counter is cleared whenever a RX transition occurs. When the internal counter reaches LINTMR[15-0], an LCNTRO interrupt is generated. The software can use this interrupt to enter sleep mode by writing the required bus idling time into LINTMR[15-0].

SBAUDH (0xA093) EUART/LIN Baud Rate Register High byte (0x00) RO

	7	6	5	4	3	2	1	0
RD	SBR[15-8]							
WR	BR[15-8]							

SBAUDL (0xA094) EUART/LIN Baud Rate Register Low byte (0x00) RO

	7	6	5	4	3	2	1	0
RD	SBR[7:0]							
WR	BR[7-0]							

SBR[15-0] The acquired Baud Rate under LIN protocol. This is read-only.
SBR[15-0] is the acquired baud rate from last received valid sync byte. SBR is meaningful only in LIN-Slave mode.

BR[15-0] The Baud Rate Setting of EUART/LIN. This is write-only. BR[15-0] can not be 0.
BUAD RATE = SYSCLK/BR[15-0].

When a slave receives a BREAK followed by a valid SYNC field, an RSI interrupt is generated and the acquired baud rate from SYNC field is stored in SBR[15-0]. The acquired baud rate is BAUD RATE = SYSCLK/SBR[15-0]. The software can just update this acquired value into BR[15-0] to achieve synchronization with the master. If Auto-Sync Update (ASU) register bit is enabled under LIN slave mode, LIN controller will automatically perform the update of BR[15-0] with SBR[15-0] and issue another ASUI interrupt when received a valid SYNC field.

LININTF (0xA095) LIN Interrupt Flag Register (0x00) R/W

	7	6	5	4	3	2	1	0
RD	-	-	LSTAT	LIDLE	ASUI	SBKI	RSI	LCNTRO
WR	-	-	-	-	ASUI	SBKI	RSI	LCNTRO

LSTAT LIN Bus Status bit (1: Recessive / 0: Dominant), Read only.
LSTAT = 1 indicates that the LIN bus (RX pin) is in recessive state.

LIDLE LIDLE is 1 when LIN bus is idle and not transmitting/receiving LIN header or data bytes. This bit read only. It is 1 when LINEN = 0.

ASUI Auto-Sync Updated completion Interrupt (1: Set / 0: Clear)
This flag is set when auto baud rate synchronization has been completed and BR[15-0] has been updated with SBR[15-0] by hardware. It must be cleared by writing “1” on the bit.

SBKI Send Break Completion Interrupt bit (1: Set / 0: Clear)
This flag is set when Send Break completes. It must be cleared by writing “1” in the bit.

RSI Receive Sync Completion Interrupt bit (1: Set / 0: Clear)
This flag is set when a valid Sync byte is received following a Break. It must be cleared by writing “1” in the bit.

LCNTRO LIN Counter Overflow Interrupt bit (1: Set / 0: Clear).
This flag is set when the LIN counter reaches 0xFFFF. It must be cleared by writing “1” in the bit.

LININTEN (0xA096) LIN Interrupt Enable Register (0x00) R/W

	7	6	5	4	3	2	1	0
RD	LINTEN	-	-	-	ASUIE	SBKIE	RSIE	LCNTRIE
WR	LINTEN	-	-	-	ASUIE	SBKIE	RSIE	LCNTRIE

LINTEN LIN Interrupt Enable (1: Enable / 0: Disable)
Set to enable all LIN interrupts. LINT flags should be checked before setting or modifying.

ASUIE Auto-Sync Update Interrupt Enable (1: Enable / 0: Disable)

SBKIE Send Break Completion Interrupt Enable (1: Enable / 0: Disable)

RSIE Receive Sync Completion Interrupt Enable (1: Enable / 0: Disable)

LCNTRIE LIN Counter Overflow Interrupt Enable (1: Enable / 0: Disable)

6. Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) is an enhanced synchronous serial hardware which is compatible with Motorola's SPI specifications. The SPI Controller includes 4-bytes FIFO for both transmit and receive. SPI Interface uses P3.5 as Master-Out-Slave-In (MOSI), P3.6 as Master-In-Slave-Out (MISO), P3.7 as SPICLK. When configured as slave it also can use P3.4 as Slave Enable signal for multi-slave SPI configurations.

SPICR (0xA1) SPI Configuration Register R/W (0b001000xx)

	7	6	5	4	3	2	1	0
RD	SPIE	SPEN	MSTR	CPOL	CPHA	SCKE	-	-
WR	SPIE	SPEN	MSTR	CPOL	CPHA	SCKE	-	-

SPIE SPI interface Interrupt Enable bit.

SPEN SPI interface Enable bit.

MSTR SPI Master/Slave Switch.(set as a master; clear as a slave)

CPOL SPI interface Polarity bit: Set to configure the SCK to stay HIGH while the SPI interface is idling and clear to keep it LOW.

CPHA Clock Phase Control bit: If CPOL=0, set to shift output data at rising edge of SCK, and clear to shift output data at falling edge of SCK. If CPOL=1, set to shift output data at falling edge of SCK and clear to shift output data at rising edge of SCK.

SCKE Clock Selection bit in Master Mode: Set to use rising edge of SCK to sample the input data. Clear to use falling edge of SCK to sample the input data.

In Slave mode, the sampling phase is determined by the combinations of CPOL and CPHA setting shown in the following table.

CPOL	CPHA	(Slave mode) SCK edge used for sampling input data	Data shift out
0	0	Rising edge	Falling edge
0	1	Falling edge	Rising edge
1	0	Falling edge	Rising edge
1	1	Rising edge	Falling edge

SPIMR (0xA2) SPI Mode Control Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	ICNT1	ICNT0	FCLR	-	SPR[2]	SPR[1]	SPR[0]	DIR
WR	ICNT1	ICNT0	FCLR	-	SPR[2]	SPR[1]	SPR[0]	DIR

ICNT1, ICNT0 FIFO Byte Count Threshold.
This sets the FIFO threshold for generating SPI interrupts.
00 – the interrupt is generated after 1 byte is sent or received;
01 – the interrupt is generated after 2 bytes are sent or received;
10 – the interrupt is generated after 3 bytes are sent or received;
11 – the interrupt is generated after 4 bytes are sent or received.

FCLR FIFO Clear/Reset
Set to clear and reset transmit and receive FIFO

SPR[2-0] SPI Clock Rate Setting. This is used to control the SCK clock rate of SPI interface.
000 – SCK = SYSCLK/6;
001 – SCK = SYSCLK/8;
010 – SCK = SYSCLK/16;
011 – SCK = SYSCLK/32;
100 – SCK = SYSCLK/64;
101 – SCK = SYSCLK/128;
110 – SCK = SYSCLK/256;
111 – SCK = SYSCLK/512.

DIR Transfer Format
DIR=1 uses MSB-first format.
DIR=0 uses LSB-first format.

SPIST (0xA3) SPI Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SSPIF	ROVR	TOVR	TUDR	RFULL	REMPT	TFULL	TEMPT
WR	SSPIF	ROVR	TOVR	TUDR	-	-	-	-

SSPIF	SPI Interrupt Flag bit. Set by hardware to indicate the completion of data transfer. Clear by assigning this bit to 0 or disabling SPI.
ROVR	Receive FIFO-overrun Error Flag bit. When Receiver FIFO Full Status occurs and SPI receives new data, ROVR is set and generates an interrupt. Clear by assigning this bit to 0 or disabling SPI.
TOVR	Transmit FIFO-overrun Error Flag bit. When Transfers FIFO Full Status occurs and new data is written, TOVR is set and generates an interrupt. Clear by assigning this bit to 0 or disabling SPI.
TUDR	Transmit Under-run Error Flag bit. When Transfers FIFO Empty Status and new data transmission occur, TOVR is set and generates an interrupt. Clear by written 0 to this bit or disable SPI.
RFULL	Receive FIFO Full Status bit . Set when receiver FIFO is full. Read only.
REMPT	Receive FIFO Empty Status bit . Set when receiver FIFO is empty. Read only.
TFULL	Transmitter FIFO Full Status bit . Set when transfer FIFO is full. Read only.
TEMPT	Transmitter FIF0 Empty Status bit . Set when transfer FIFO is empty. Read only.

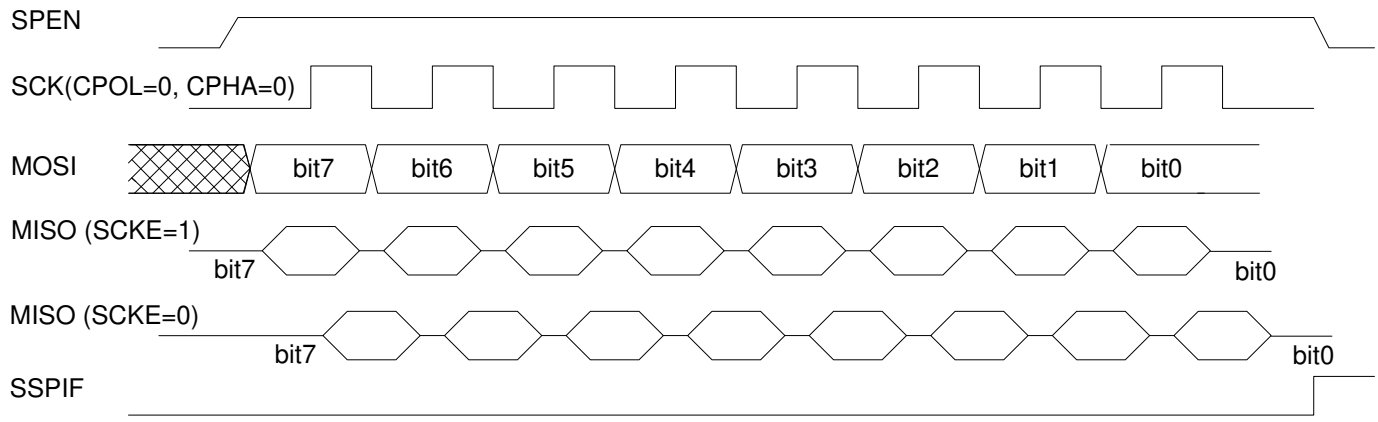
SPIDATA (0xA4) R/W (0xXX) SPI Data Register

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

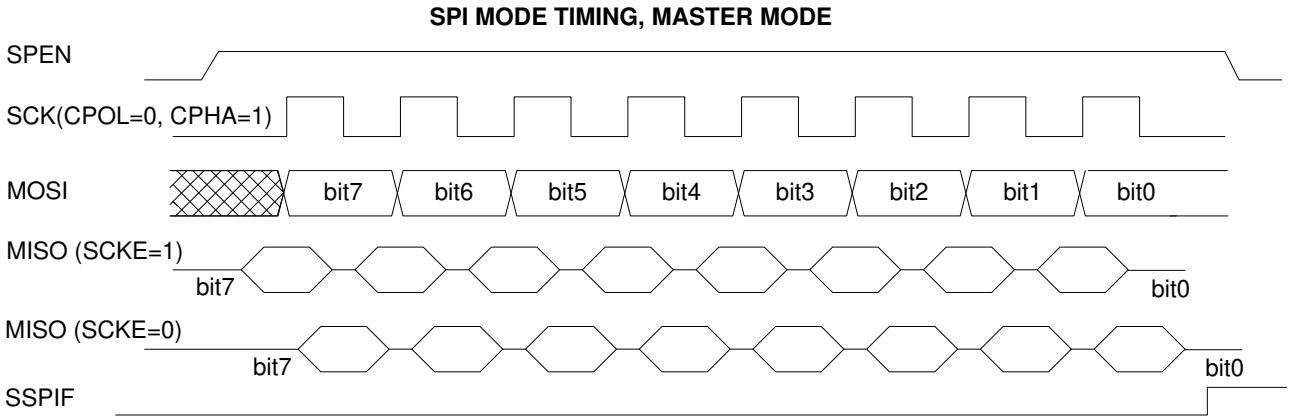
6.1 SPI Master Timing Illustration

6.1.1 CPOL=0 CPHA=0

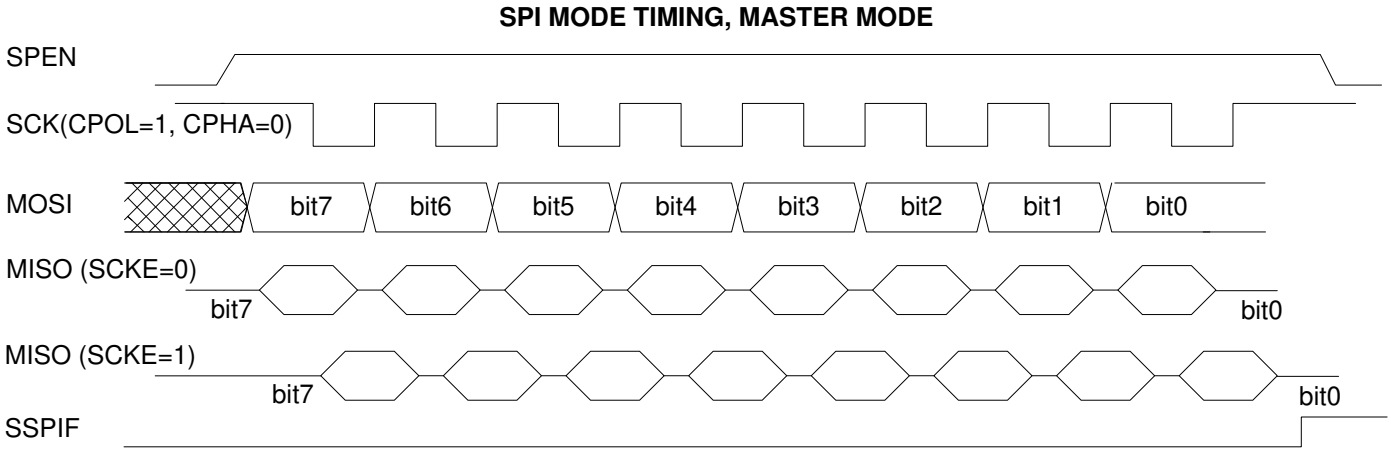
SPI MODE TIMING, MASTER MODE



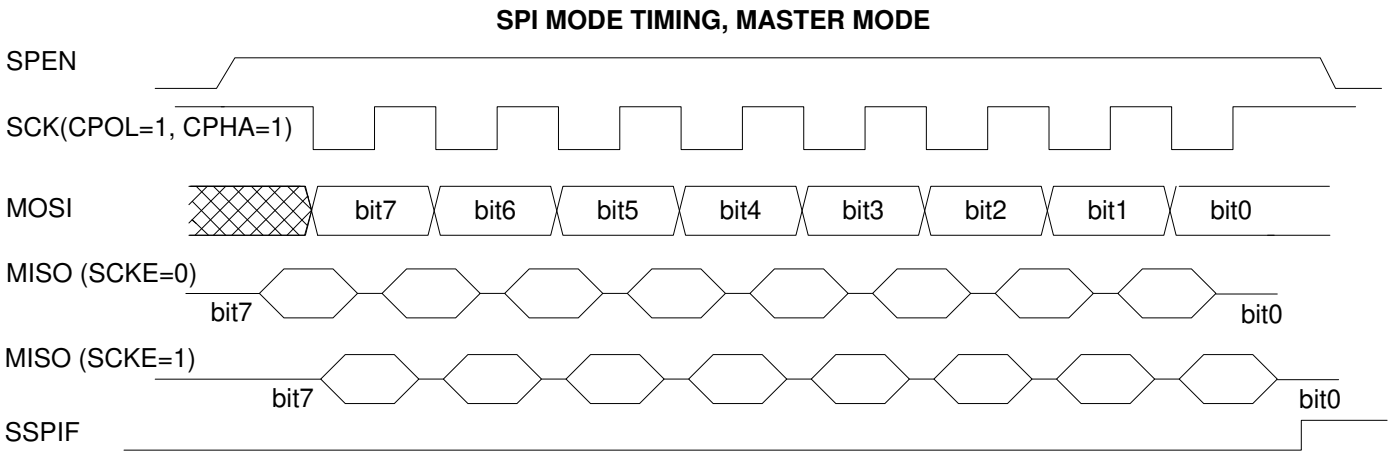
6.1.2 CPOL=0 CPHA=1



6.1.3 CPOL=1 CPHA=0

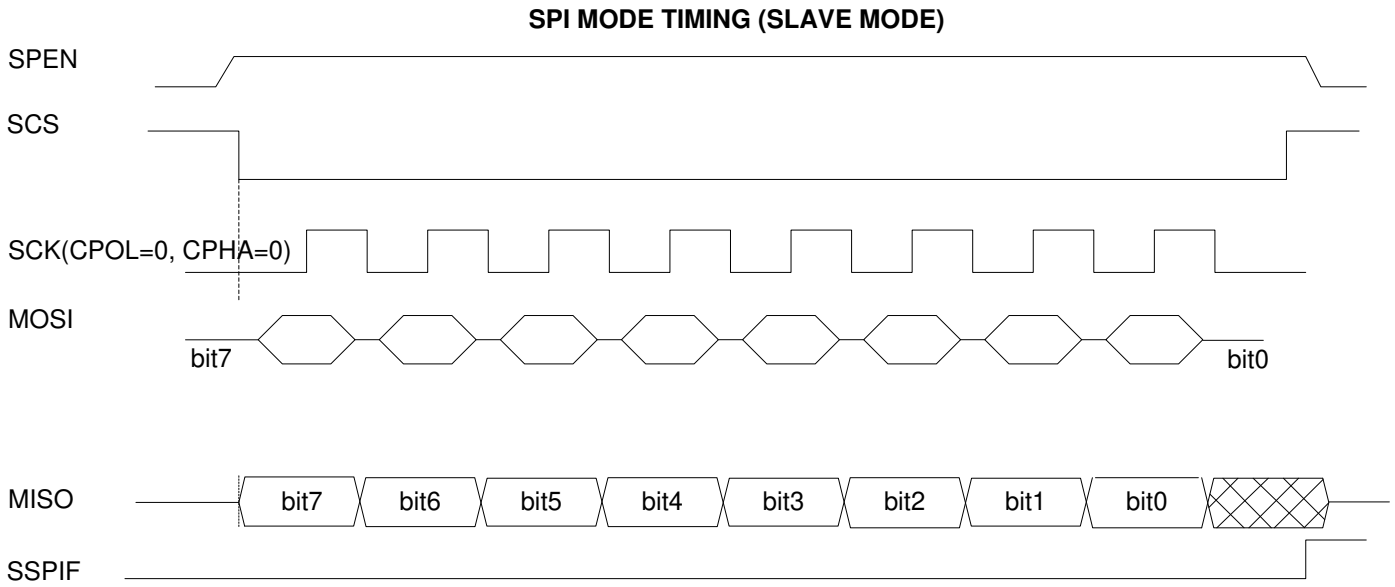


6.1.4 CPOL=1 CPHA=1

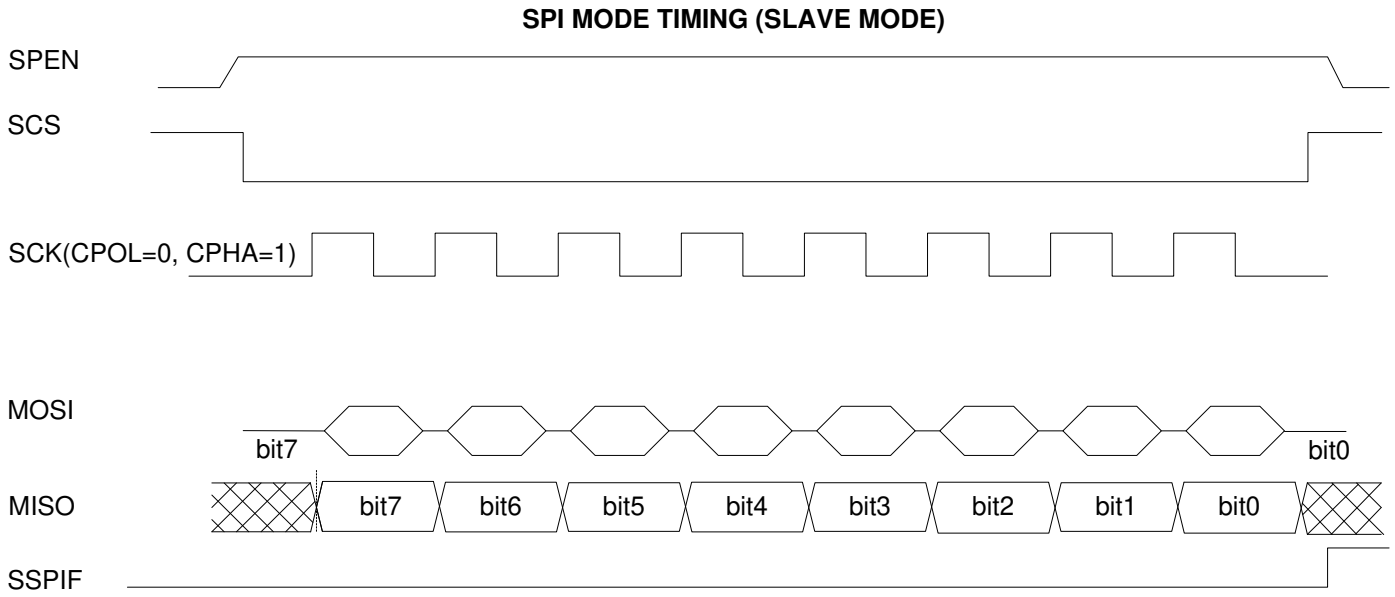


6.2 SPI Slave Timing Illustration

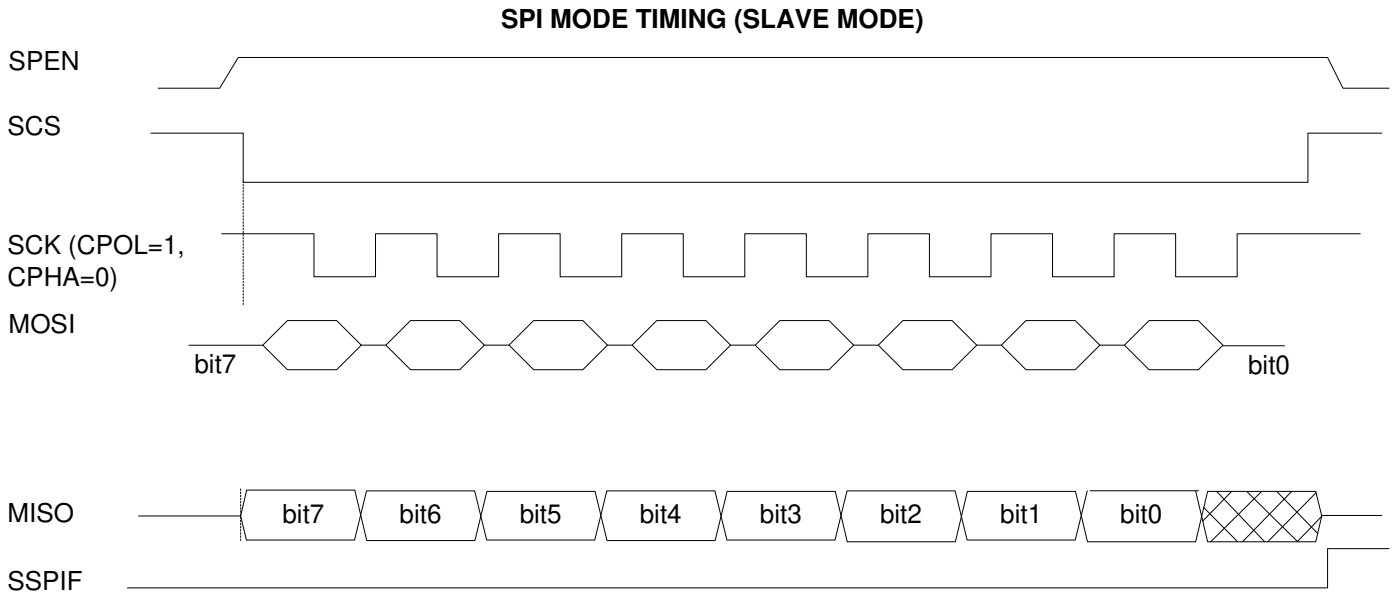
6.2.1 CPOL=0 CPHA=0



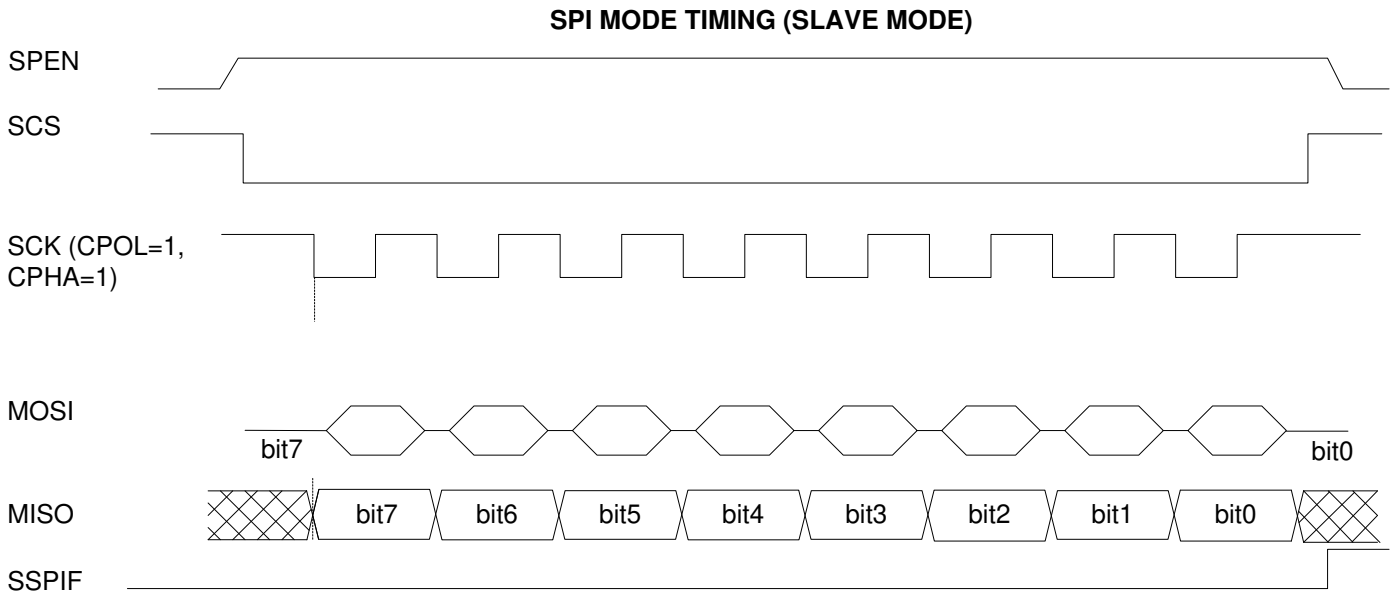
6.2.2 CPOL=0 CPHA=1



6.2.3 CPOL=1 CPHA=0

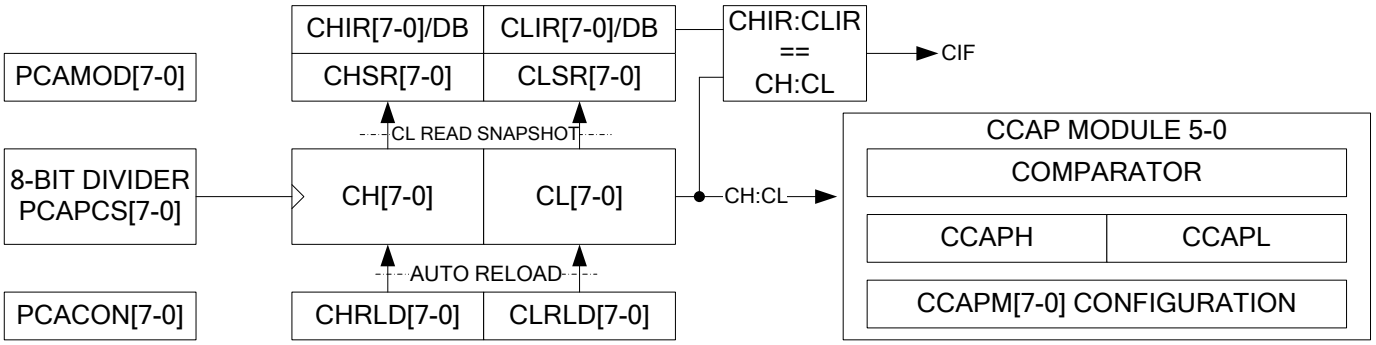


6.2.4 CPOL=1 CPHA=1



7. Programmable Counter Array (PCA) and Compare/Capture/PWM (CCP)

The PCA provides enhanced timing functions with less CPU intervention than the standard 8051 timers T0, T1, and T2. The PCA is partitioned in three parts. The main PCA Counter consists of CH and CL. There are 6 channels of Compare/Capture/PWM modules



The MAIN COUNTER (CH and CL) is configured and controlled by two registers, CMOD and PCACON. The counter value is accessed by CH and CL registers. The counter can be configured as either FREE-RUN or AUTO-RELOADED mode. The counter values of CH and CL can be captured in CHSR and CLSR triggered by software or hardware. There is also a counter compare register CHIR and CHLR. An interrupt can be enabled at CH:CL == CHIR:CLIR. This allows the PCA to easily synchronize with the software control. CHIR and CLIR are double-buffered.

PCAPCS (0xA0A5) PCA Counter Clock Prescal Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PCACPS[7-0]							
WR	PCACPS[7-0]							

PCACPS sets the clock input to the PCA at SYSCLK/(PCACS[7-0]+1).

PCAMOD (0xD1) PCA Mode Control Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CIDL	RLDEN	COUNT8	OVF8EN	PCAEN	ECF	CIFEN	CMPTRIG
WR	CIDL	RLDEN	COUNT8	OVF8EN	PCAEN	ECF	CIFEN	CMPTRIG

CIDL — Counter Control bit in IDLE mode
When CIDL=1, PCA counting is disabled in IDLE mode.
When CIDL=0, normal counting of PCA in IDLE mode persists. PCAEN needs to be 1 for counting.

RLDEN: — AUTO-RELOAD Mode Enable bit
Set RLDEN=1to enable AUTO-RELOAD mode. At overflow, the main counter is reloaded with CHRL and CHRH in 16-bit mode or CHRL in 8-bit mode.
When RLDEN=0: FREE-RUN mode.

COUNT8 — 8-Bit or 16-Bit Counter Mode Select bit
When COUNT8=1 the PCA is configured as an 8-Bit counter. In 8-bit counter mode, OVF8EN must be set as 1.
COUNT8=0 the PCA is configured as a16-Bit counter

OVF8EN — 8-Bit Overflow Enable bit
When OVF8EN=1, the PCA overflow condition occurs at 0xXXFF to 0xXX00. In other words, the overflow condition (CF flag) is set every 256 count. This overflow condition also applies to the 16-bit counter mode.
When OVF8EN=0 the PCA overflow condition occurs at 0xFFFF to 0X0000. This does not work in 16-bit counter mode.

PCAEN — PCA counter Enable bit
Set PCAEN=1 to enable the PCA counter
Set PCAEN=0 to disable the PCA and also clears the counter value. When PCAEN=0, all double buffer is loaded with reload values.

ECF — Counter Overflow Interrupt Flag bit
When ECF=1 the overflow condition interrupt is enabled.

When ECS=0 the overflow interrupt is disabled.

CIFEN — Count Compare Interrupt Enable
Set IFEN=1 to enable CHIR:CLIR == CH:CL interrupt
Set IFEN=0 to disable this interrupt

CMPTRIG — Comparator Trigger Enable
CMPTRIG=1 enables the snapshot of PCA count value into CHSR by analog comparator interrupt.
CMPTRIG=0 disables the triggering.

PCACON (0xE1) PCA Configuration Register R/W (0x00)

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

CF — Counter Overflow Flag bit
CF is set to 1 by hardware when overflow condition occurs. The overflow condition occurs at either of 0xFFFF to 0x0000 (OVF8EN=0) or 0xXXFF to 0xXX00 (OVF8EN=1). This bit must be cleared by software.

CIF — Count Compare Flag bit
CIF is set by hardware when CH:CL == CHIR:CLIR. This bit must be cleared by software.

CCF5 - CCF0 — Module Interrupt Flag 5-0
This is set by hardware as its corresponding module generates an interrupt. These bits must be cleared by software.

CH (0xE9) PCA Main Counter High Byte R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CH[7-0]							
WR	-							

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

CL (0xD9) PCA Main Counter Low Byte R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CL[7-0]							
WR	-							

CL holds the lower 8-bit of the main counter value. Reading CL will trigger a snapshot action to copy CH:CL to CHSR:CLSR.

CHRLD (0xA0A7) PCA Counter CH Reload Value Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CHRLD[7-0]							
WR	CHRLD[7-0]							

This register holds the reload value for CH in AUTO-RELOAD mode.

CLRLD (0xA0A6) PCA Counter CL Reload Value Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CLRLD[7-0]							
WR	CLRLD[7-0]							

This register holds the reload value for CL in AUTO-RELOAD mode.

CHSR (0xF3) PCA Snapshot Register of CH RW (0x00)

	7	6	5	4	3	2	1	0
RD	CHSR[7-0]							
WR	CHIR[7-0]							

CHSR[7-0] CH Snapshot Register. It is read-only.
CHIR[7-0] CH Counter Compare Interrupt . The compare value is double- buffered.

CLSR (0xF2) PCA Snapshot Register of CL RW (0x00)

	7	6	5	4	3	2	1	0
RD	CLSR[7-0]							
WR	CLIR[7-0]							

CLSR[7-0] CL Snapshot Register. It is read-only.
CLIR[7-0] CL Counter Compare Interrupt The compare value is double-buffered.

The Compare/Capture modules receive the 16-bit count value from the main counter as the time base. Each module is configured by its mode register CCAPMn and contains two 8-bit registers used for comparing value holder or capturing value in 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 CCP Module Configuration Register (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 configured as PWM mode, OF is ignored. Set TOG=1 to enable CEX output to high regardless of PWM value.

ECOM Comparator Enable bit. Set to enable comparator function. Clear to disable the comparator.
CAPP Positive Edge Capture bit. Set to use a positive edge as the capture edge. Clear to disable positive edge capture.
CAPN Negative Edge Capture bit. Set to use a 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 causes CCF to be set and generates an interrupt. It also enables a compare edge interrupt in WPWM mode.
PWM Pulse Width Modulation bit. Set to enable PWM function. CEX is the PWM output.
ECCF Enable Capture/Compare/PWM Interrupt bit. Set to enable the CCP module n (n = number of the designated module; there are 6 modules in this case) to generate the interrupt.

CCAPnL CCP Compare Value Low Register (0xD2, 0xD4, 0xD6, 0xE2, 0xE4, 0xE6) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CCAPnL[7-0]							
WR	CCAPnL[7-0]							

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

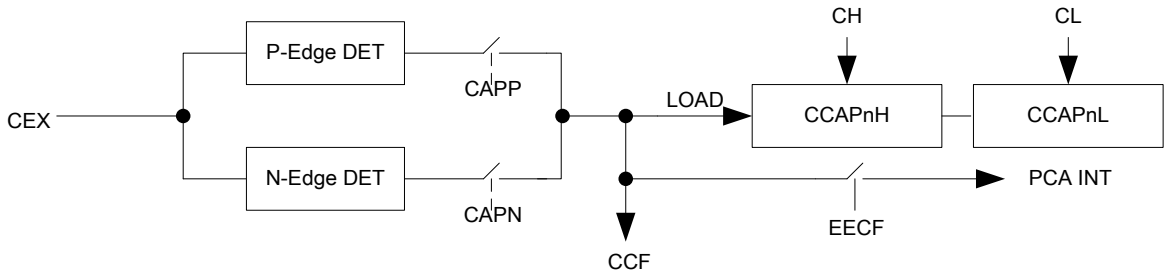
CCAPnH CCP Compare Value High Register (0xD3, 0xD5, 0xD7, 0xE3, 0xE5, 0xE7) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CCAPnH[7-0]							
WR	CCAPnH[7-0]							

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

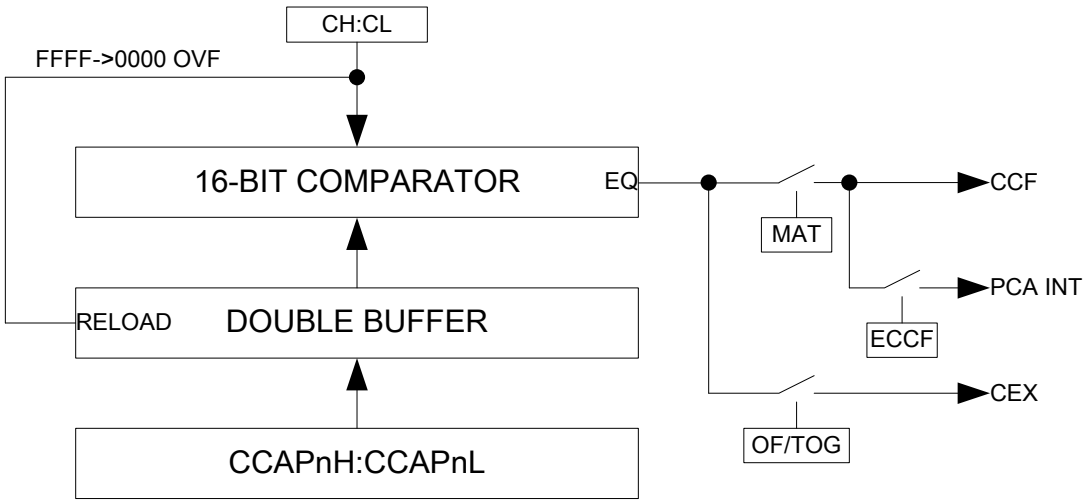
7.1 16-Bit CAPTURE MODE

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.



7.2 16-Bit COMPARE TIMER MODE

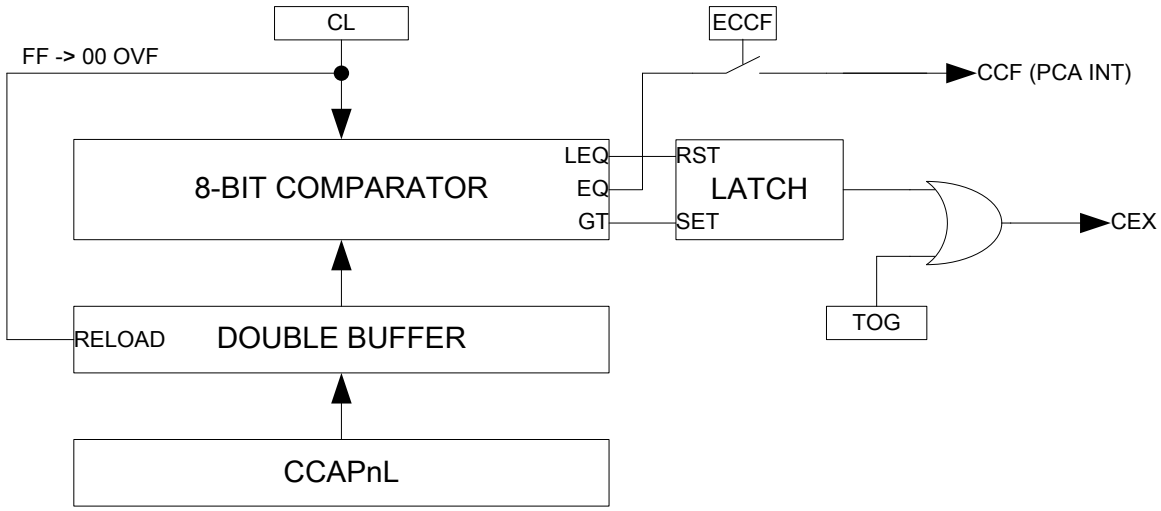
The COMPARE TIMER mode can be used as a software timer or to generate a PWM output. This mode is enabled when ECOM is set and CAPP CAPN are set to low. To allow the compare result to be used, MAT/ECCF also needs to be set. The CCAPnH and CCAPnL hold the 16-bit Timer value and are compared against the incrementing value CH and CL from the main counter. The compare value is double-buffered and is updated when the main counter overflows. This prevents any unexpected comparator output during updating a new value to CCAPnH and CCAPnL. When a match occurs, CCF is set and an 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 glitches or a match condition when updating the CCAPnH and CCAPnL registers, when ECOM is set and the writing to CCAPnL causes ECOM to clear. Writing to CCAPnH sets 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.

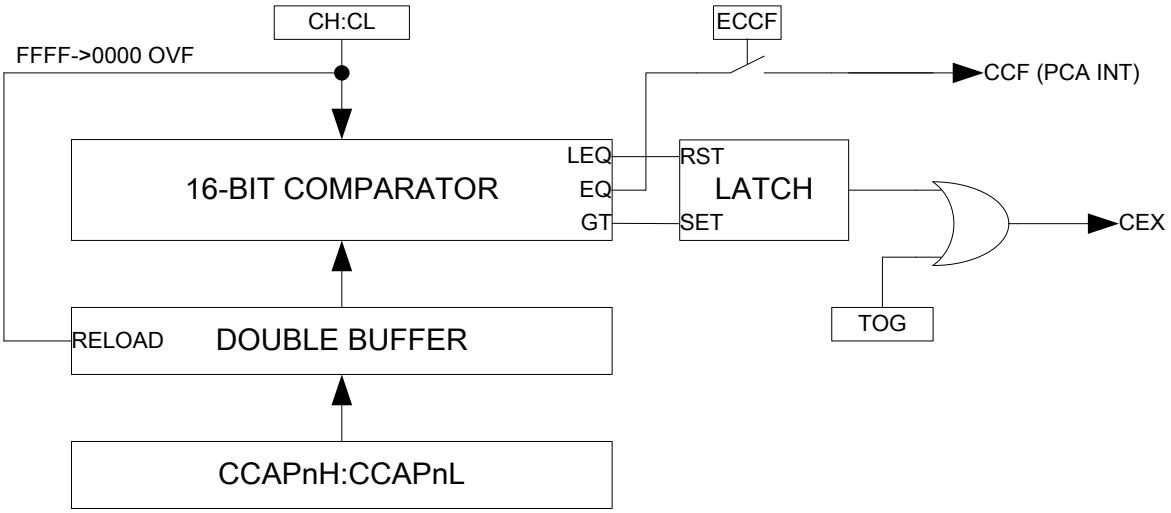
7.3 8-Bit Pulse Width Modulator Mode

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. CCAPnL is used for compare value. When CL ≦ CCAPnL, the output is 0 and when CL > CCAPnL, the output is 1. The compare value is double-buffered and is updated when CL overflows from FF to 00. 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, the maximum CEX duty cycle is 255/256. If TOG is set to 1 in this mode, CEX is forced high to provide 256/256 with full high duty cycle. If ECCF bit is set, then when CCAPnL=CL (i.e. the output change), CCF is also set to 1 by hardware and triggers a PCA interrupt. The following block diagram shows the PWM mode operation.



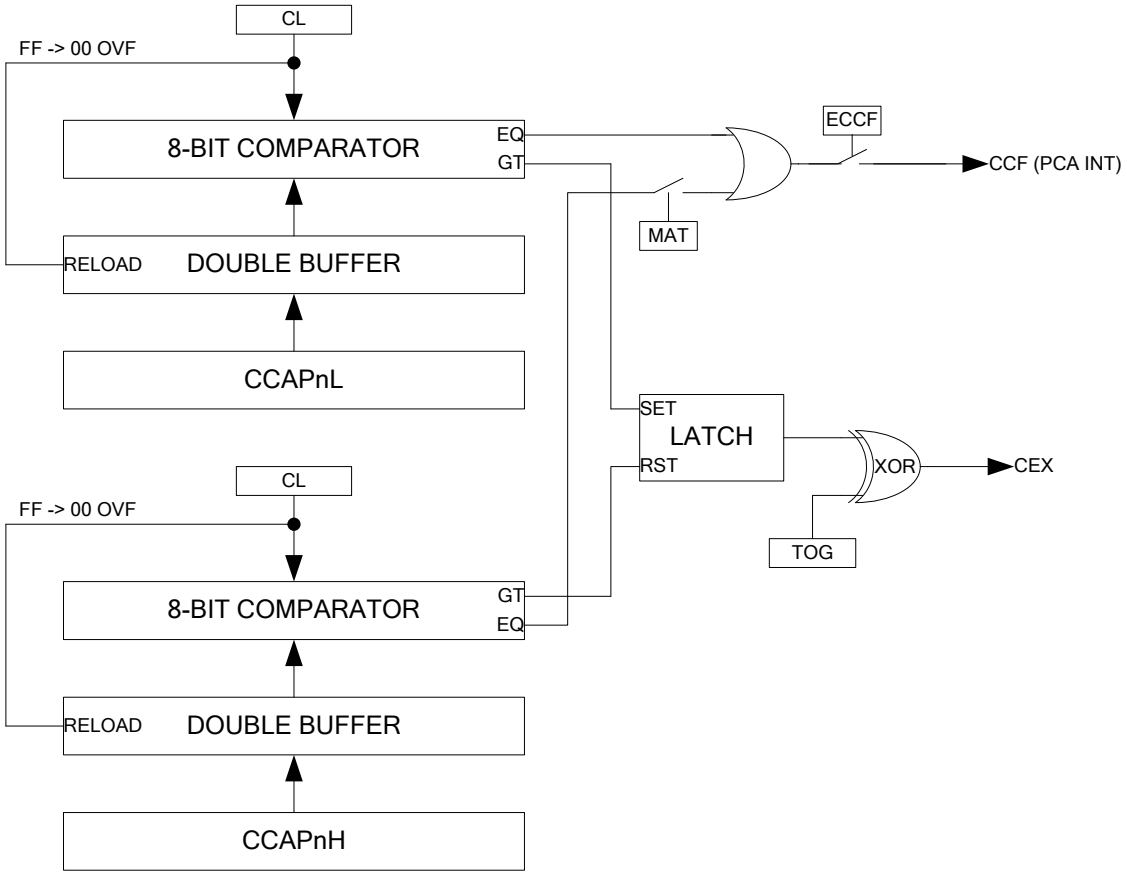
7.4 16-Bit Pulse Width Modulator Mode

This mode is similar to the 8-bit PWM mode except it uses the 16-bit CH:CL count value for the time base of the PWM. The compare value is composed of CCAPnH:CCAPnL and is double-buffered. When CH:CL ≦ CCAPnHL:CCAPnL, the output is 0; when CH:CL > CCAPnHL:CCAPnL, the output is 1. The output can be forced to 1 by setting TOG=1. The PWM mode is enabled when both ECOM and PWM bits and CAPP are set while CAPN is low. An interrupt is enabled by ECCF and triggered when CH:CL==CCAPnH:CCAnL.



7.5 8-BIT Wnidowed Pulse Width Modulator (WPWM) Mode

This mode is used to generate 8-bit 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. The compare values are double-buffered. Therefore the value in CCAPnH must be larger than CCAPnL to prevent abnormal operations. The output of CEX can be inverted by setting TOG to 1. An interrupt can be enabled by ECCF, if MAT=0, then CL=CCAPnL generates an interrupt. Setting MAT to 1 and CL=CCAPnH also generates an interrupt.



7.6 CCP FUNCTION SUMMARY

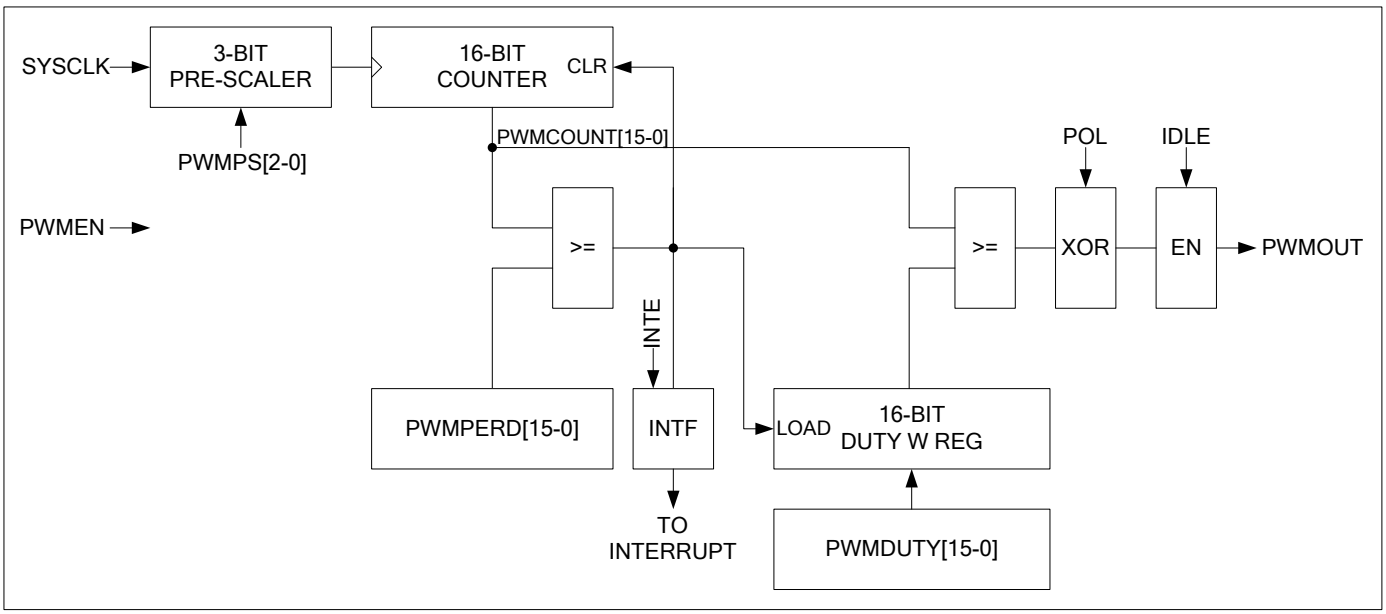
CCP Function			ECOM	CAPP	CAPN	PWM	OF	TOG
No operation *(Note 1)*			0	0	0	0	X	X
16-bit Capture	Triggered by positive edge of CEX		X	0	1	0	X	X
	Triggered by negative edge of CEX			1	0			
	Triggered by both edges of CEX			1	1			
16-bit Compare *(Note 2)*	CH:CL == CCAPnH:CCAPnL	CEX is unchanged	1	0	0	0	0	0
		CEX toggles					0	1
		CEX = 0 (or stay 0)					1	0
		CEX = 1 (or stay 1)					1	1
8-bit PWM	CEX = 0 when CL <= CCAPnL CEX = 1 when CL > CCAPnL		1	0	0	1	X	0
	CEX = 1							1
16-bit PWM	CEX=0 if CH:CL =< CCAPnH:CCAPnL CEX=1 if CH:CL > CCAPnH:CCAPnL		1	1	0	1	X	0
	CEX=1							1
8-bit WPWM	CEX = 0 when CL > CCAPnH CEX = 1 when CL > CCAPnL		1	1	1	1	X	0
	CEX = 1 when CL > CCAPnH CEX = 0 when CL > CCAPnL							1

Note:

1. *ECOM cannot be set to 1 by hardware (when writing to CCAPnH) if all bits (OF, ECOM, CAPP, CAPN, MAT, TOG, PWM) in CCAPM are set to 0 (NO OPERATION mode).*
2. *In 16-bit compare mode, ECOM can be set to 1 by hardware (when writing to CCAnPH) or software, and can be cleared to 0 by hardware (when writing to CCAnPL) or software. When ECOM is cleared to 0 in this mode, the CCP function enters NO OPERATION mode. The compare value is CCAPnH:CCAPnL and is double-buffered.*

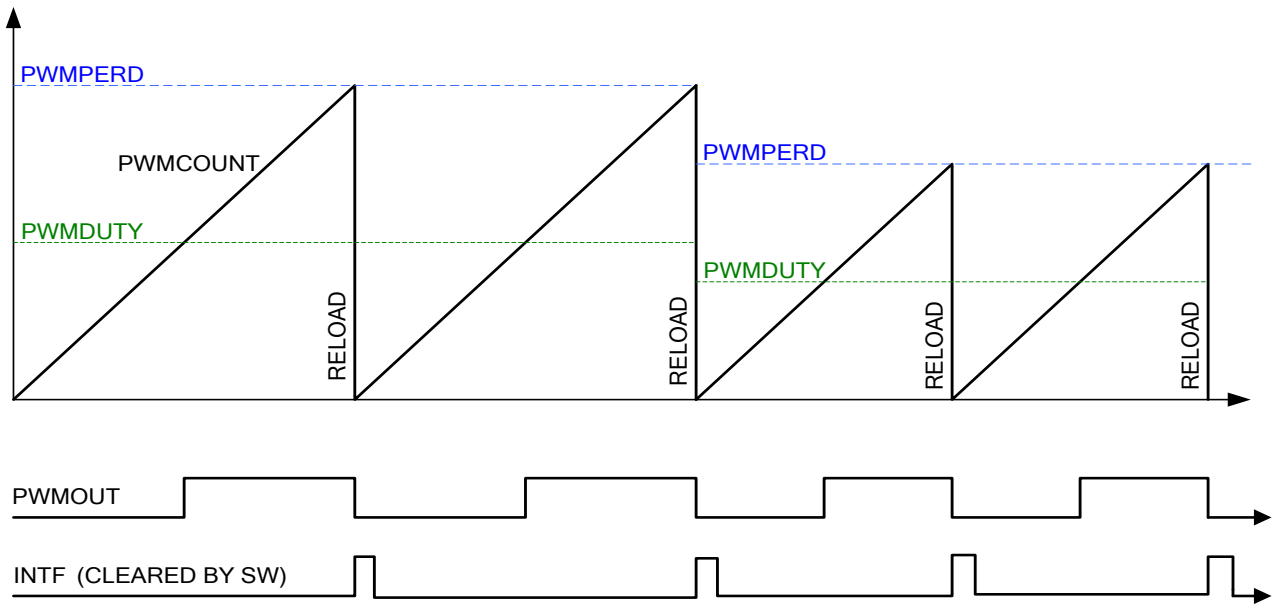
8. 16-Bit PWM Controller

This PWM controller provides high precision up to 16-bit Pulse-Width Modulation output with programmable PWM base frequency. The block diagram of PWM control is shown in the following block diagram.



The PWM control includes a 16-bit counter that receives SYSCLK as the basic clock source with a 3-bit pre-scaler divider. The PWM base period is set by PWMPERD[15-0]. When the 16-bit counter, PWMCOUNT[15-0], reaches PWMPERD[15-0], the counter is reloaded with 0. The period of each PWM cycle is thus SYSCLK * (PWMPS[2-0] +1) * (PWMPERD[15-0]+1). The duty cycle or the on-duration of the PWM cycle is controlled by PWMDUTY[15-0]. When PWMCOUNT[15-0] is less than PWMDUTY[15-0], the output of the PWM is set to 0, and when it is equal or greater, the output is set to 1. There is also a polarity control of the final PWM output.

There is a double buffer for DUTY control. The compared DUTY value is uploaded only when the PWM counter is reloaded. Therefore, there is no spurious effect when changing the duty value. Each time when PWMCOUNT is cleared to 0, it also generates an interrupt to indicate a new PWM cycle is started. Thus MCU can prepare a new data to for the duty value when it receives the interrupt. The interrupt of this PWM module goes into INT4, which is shared with ADC's interrupt. The operation of the PWM can also be described in the following timing diagram.



The following register definitions specify each control register of the PWM module.

PWM16CTL (0xA028) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PWMEN	PWMPS[2]	PWMPS[1]	PWMPS[0]	IDLE	INTEN	POL	INTF
WR	PWMEN	PWMPS[2]	PWMPS[1]	PWMPS[0]	IDLE	INTEN	POL	INTF

PWMEN Set this bit to 1 to enable the PMW module. Clear this bit to 0 will disable the module, the output of the PMW is equal to IDLE setting.

PWMPS[2-0] These two bits determine the clock pre-scaling factor for the PWM counter. The PWM clock is SYSCLK/(PWMPS[2-0]+1).

IDLE This bit determines the value of PWMOUT during disable state of the PWM module.

INTEN Set this bit to enable the PWM interrupt. The interrupt occurs at each PWM counter reload

POL This bit controls the PMWOUT polarity. When POL=0, PWMOUT=0 when PWMCOUNT < PWMDUTY, and PWMOUT=1 when PWMCOUNT >= PWMDUTY. When POL=1, the output is reversed.

INTF This bit is the interrupt flag. INTF is set when the PWM counter is reloaded. The software must clear the interrupt flag by setting this bit. INTF is also sent interrupt INT4 which is shared with ADC interrupt.

PWMPERDH (0xA029) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PWMPERD[15-8]							
WR	PWMPERD[15-8]							

This register defines the period count high byte.

PWMPERDL (0xA02A) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PWMPERD[7-0]							
WR	PWMPERD[7-0]							

This register defines the period count low byte.

PWMDUTYH (0xA02B) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	PWMDUTY[15-8]							
WR	PWMDUTY[15-8]							

This register defines the DUTY high byte.

PWMDUTYL (0xA02C) R/W (0x00)

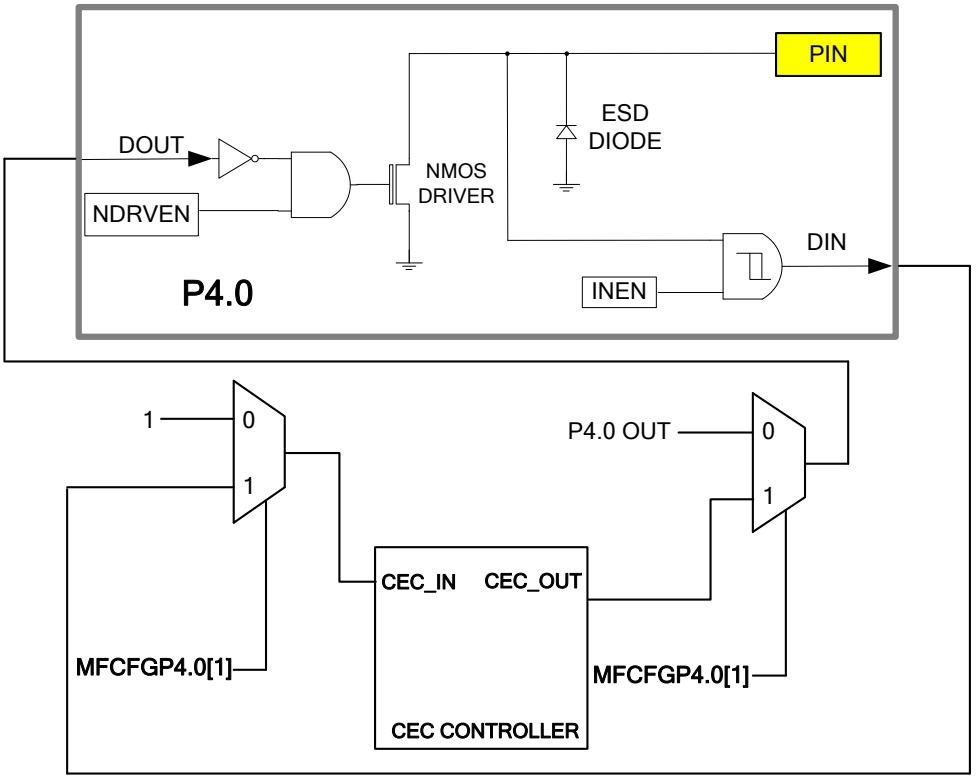
	7	6	5	4	3	2	1	0
RD	PWMDUTY[7-0]							
WR	PWMDUTY[7-0]							

This register defines the DUTY low byte.

Note: When modify the PWM duty and the PWM period, the updating PWMDUTY[15-0] and PWMPERD[7-0] (i.e. PWMPERDL) values will be kept into their corresponding temporary buffers. These values will be reloaded into the PWM counter when MCU programs the PWMPERD[15-8] (i.e. PWMPERDH) register.

9. CEC Controller

The CEC Controller can be configured as one of CEC initiator or follower. The CEC Controller uses P4.0 as CEC interface pin. Thus MFCFGP4.0[1] (XFR address A0D0H) must be set to “1”, so that P4.0 is configured as CEC function. In addition, IOCFGP4.0 (XFR address A0C0H) must be configured to 82h, so that P4.0 can work in open-drain mode, which is required by CEC protocol. Diagram below shows the configuration for CEC module and P4.0:



CECCTL (0xAC) CEC Control Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	SENDBYTE	STPRDT	BACKDEN	SENDNTY	BFSEND	BFREC	CEC_IN	SENDSBYTE
WR	SENDBYTE	STPRDT	BACKDEN	SENDNTY	-	-	-	SENDSBYTE

SENDSBYTE	It sets this bit to initiate the following operations in sequence 1. Start arbitration (on error, automatically set TXERRIF) 2. Transmit a byte 3. Receive Acknowledge bit from the follower and store it in ACK This bit will be cleared automatically when the CEC byte is transmitted.
STPRDT	Start Bit Period Tolerance Range Adjust When STPRDT=1, CEC receiver judges the receiving Start Bit with higher timing tolerance. STPRDT=0: The timing tolerance = ± CECBPRD[7:0] / 16 STPRDT=1: The timing tolerance = ± CECBPRD[7:0] / 8
BACKDEN	Check ACK bit enable when sending broadcast frame as an initiator. When BACKDEN=1, initiator checks ACK signal and issues an interrupt for error detection When BACKDEN=0, initiator disables check ACK at first byte transmit in broadcast mode
SENDNTY	It sets this bit to transmit a NOTIFICATION. This bit will be cleared automatically when the NOTIFICATION is transmitted.
BFSEND	Internal state, it indicates the current state is set to “send broadcast frame” if “1” is read
BFREC	Internal state, it indicates that the current state is set to “receive broadcast frame” if “1” is read
CEC_IN	Indicate the current CEC input value.
SENDBYTE	It sets this bit to transmit a byte and receives ACKNOWLEDGE bit from the follower and stores it in ACK This bit will be cleared automatically when the CEC byte is transmitted.

CECCFG (0xA080) CEC Configuration Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	TXMODE	TXCONT	ESTI	ETMOUTI	ELNERRI	ETXERRI	ETXI	ERXI
WR	TXMODE	TXCONT	ESTI	ETMOUTI	ELNERRI	ETXERRI	ETXI	ERXI

TXMODE — CEC Transmission Mode Enable bit.
TXMODE = 0: CEC in Receive mode
TXMODE = 1: CEC in Transmission mode

TXCONT — Continuous Transmit Enable bit.
Set both TXMODE and TXCONT to enable the continuous transmit mode.

ESTI — Start Interrupt Enable bit
Write 1 to enable the STIF interrupt. Write 0 to disable the STIF interrupt.

ETMOUTI — Time-out Interrupt Enable bit.
Write 1 to enable the TMOUTIF interrupt. Write 0 to disable the TMOUTIF interrupt.

ELNERRI — Line Error Interrupt Enable bit
Write 1 to enable the LNERRIF interrupt. Write 0 to disable the LNERRIF interrupt.

ETXERRI — Arbitration and Transmission Error Interrupt bit.
Write 1 to enable the TXERRIF interrupt. Write 0 to disable the TXERRIF interrupt.

ETXI — Transmission Complete Interrupt Enable bit.
Write 1 to enable the TXIF interrupt. Write 0 to disable the TXIF interrupt.

ERXI — Receiving Complete Interrupt Enable bit.
Write 1 to enable the RXIF interrupt. Write 0 to disable the RXIF interrupt.

CECADR (0xA081) CEC Address Register R/W (0x40)

	7	6	5	4	3	2	1	0
RD	CECEN	AUTOSW	ARBLF	FIRSTBF	CECADR[3-0]			
WR	CECEN	AUTOSW	-	-	CECADR[3-0]			

CECEN — Global CEC Function Enable bit. (1: Enable / 0: Disable)

AUTO_SW — CEC Controller Auto Switch: set as an initiator to follower when arbitration lost. The bit's default setting is 1
AUTO_SW = 1: the initiator will switch to follower automatically when arbitration lost.
AUTO_SW = 0: the initiator will wait next bus idle state by aborting the transmission.

ARBLF — Arbitration Lost Flag bit
ARBLF=1 indicates that the initiator has lost the arbitration. This bit is set by hardware when the arbitration lost happened during transmission and is cleared when bus idle. This bit is read only.

FIRSTBF — First Byte Flag bit
FIRSTBF=1 indicates current byte is the 1st byte received. This bit is set by hardware and is cleared when bus idle. This bit is read only.

CECADR[3:0] — CEC Address Device
If the CEC controller is configured as follower which detects a START event and its associated destination address, it sends the nACK onto the CEC line only if the destination address is the same as CECADR[3:0] or 1111_b.

CECSTS (0xAD) CEC Status Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	EOM	ACK	STIF	TMOUTIF	LINERRIF	TXERRIF	TXIF	RXIF
WR	EOM	ACK	STIF	TMOUTIF	LINERRIF	TXERRIF	TXIF	RXIF

EOM — EOM Control / Status Bit
Write this bit to transmit the EOM bit to CEC line as an initiator.
Read this bit to check out the EOM status from CEC line as a follower.

ACK — ACK Control / Status Bit
Write 1 to enable the follower issue the ACK bit during receiving, write 0 to disable.
Read this bit to check the ACK status during transmission.

STIF — Start Condition Detection Flag bit
Write 0 to clear.

TMOUTIF	Time-out Flag bit (CEC controller is waiting for the next bit.) This bit is set when the bus idle time > CECTMOUTLMT. This bit is cleared by writing 0 or a new transmission / receiving is issued.
LINERRIF	Line Error Flag bit (the CEC controller is monitoring the CEC line.) Write 0 to clear.
TXERRIF	Arbitration or Transmission Error Flag bit Write 0 to clear.
TXIF	Transmission Complete Flag bit Write 0 to clear.
RXIF	Receiving Complete Flag bit Write 0 to clear.

CECTXD (0x9A) CEC Transmit Data Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECTXD[7:0]							
WR	CECTXD[7:0]							

CECTXD[7:0] contains the byte data waiting to be transmitted.

CECRXD (0x9C) CEC Receive Data Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECRXD[7:0]							
WR	CECRXD[7:0]							

CECRXD[7:0] contains the byte data received from CEC line.

CECCKPRS (0xA082) CEC Clock Pre-Scaler Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECCKPRS[7:0]							
WR	CECCKPRS[7:0]							

CECCKPRS[7:0] sets the clock divider for the CEC controller. The CEC Time Unit (CTU) is expressed as “CTU = SYSCLK / (CECCKPRS[7-0] X 2)” if “HCKPRS = 0” or “CTU = SYSCLK / (CECCKPRS[7-0])” if “HCKPRS = 1”. All CEC timing is based on CTU.

The time period of CTU shall be configured around 19us. For example, if SYSCLK is 16MHz, the suggested CECCKPS[7:0] is 0x98 with “HCKPRS = 0”.

CECBPRD (0xA083) CEC Bit Period Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECBPRD[7:0]							
WR	CECBPRD[7:0]							

CECBPRD[7:0] determines the bit period of the CEC data transmission and receiving. This is in CTU unit. For the nominal data bit period, “CECBPRD[7:0] x CTU” shall be configured to 2.4ms.

CEC0LDTY (0xA084) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CEC0LDTY[7:0]							
WR	CEC0LDTY[7:0]							

CEC0LDTY[7:0] configures the LOW duty to encode a logical “0” state. This is in CTU unit. For the nominal data bit period, “CEC0LDTY[7:0] x CTU” shall be configured to 1.5ms.

CEC1LDTY (0xA085) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CEC1LDTY[7:0]							
WR	CEC1LDTY[7:0]							

CEC1LDTY[7:0] configures the LOW duty to encode a logical “1” state. This is in CTU unit. For the nominal data bit period, “CEC1LDTY[7:0] x CTU” shall be configured to 0.6ms.

CECSTPRD (0xA086) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECSTPRD[7:0]							
WR	CECSTPRD[7:0]							

CECSTPRD[7:0] configures the period of START transmission and START matching. This is in CTU unit. For the nominal data bit period, “CECSTPRD[7:0] x CTU” shall be configured to 4.5ms.

CECSTLDTY (0xA087) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECSTLDTY[7:0]							
WR	CECSTLDTY[7:0]							

CECSTLDTY[7:0] configures the LOW duty of START matching. The valid CEC LOW duty of START matching is larger than “CECSTXPNT[7:0] x CTU” and less than ”CECSTLDTY[7:0] x CTU”. This is in CTU unit. For the nominal data bit period, “CECSTLDTY[7:0] x CTU” shall be configured more than 3.9ms.

CECSTRBPNT (0xA088) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECSTRBPNT[7:0]							
WR	CECSTRBPNT[7:0]							

CECSTRBPNT[7:0] configures strobe point for decoding a bit. This is in CTU unit. For the nominal data bit period, “CECSTRBPNT[7:0] x CTU” shall be configured to 1.05ms.

CECTXFREETM (0xA089) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECFREECNT[7:0]							
WR	CECTXFREETM[7:0]							

CECFREECNT Current bus free time period based on frequency “CTU x 8”.
The counter will be stopped when CECFREECNT = 0xFF and be cleared by CEC falling.

CECTXFREETM CEC TXD Bus Free Comparator Time.
When the TXD Bus Free Time reach “8 x CECTXFREETM[7:0] x CTU”, the status flag “BUSFREEETX” and the interrupt flag “INTTXFG” will be issued to notice the users.

CECNTYLDTY (0xA08A) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECNTYLDTY[7:0]							
WR	CECNTYLDTY[7:0]							

CECNTLDTY[7:0] configures LOW-duty of NOTIFICATION. This is in CTU unit.

CECSTXPNT (0xA08B) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECSTXPNT[7:0]							
WR	CECSTXPNT[7:0]							

CECSTXPNT[7:0] set the ignored point for START event matching. The valid LOW duty of START matching shall be larger than “CECSTXPNT[7:0] x CTU” and less than ”CECSTLDTY[7:0] x CTU”. This is in CTU unit. For the nominal data bit period, “CECSTXPNT[7:0] x CTU” shall be configured less than 3.5ms.

CECMINEVDUR (0xA08C) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECMINEVDUR[7:0]							
WR	CECMINEVDUR[7:0]							

CECMINEVDUR[7:0] configures minimum duration of two contiguous falling edges for LINE_ERROR determination. If any incoming bit period is less than “CECMINEVDUR[7:0] x CTU”, the line error flag “LINERRIF” will be set to notice the users. This is in CTU unit. For the nominal data bit period, “CECMINEVDUR[7:0] x CTU” shall be configured less than 2.05ms.

CECTMOUTLMT (0xA08D) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CECTMOUTLMT[7:0]							
WR	CECTMOUTLMT[7:0]							

CECTMOUTLMT[7:0] configures maximum time limitation of two contiguous falling edges for TIMEOUT determination. Before achieving 9 bits, the CEC controller waits for the CEC Line and judges if there is TIMEOUT according to the setting. When the period of 9 bits completes, the TIMEOUT detection stops. This is in CTU unit. For the nominal data bit period, “CECTMOUTLMT[7:0] x CTU” shall be configured more than 2.75ms.

If the timeout counter reaches “CECTMOUTLMT[7:0] x CTU” when the CEC bus keeps low, the Time-out Flag “TMOUTIF” will be set and abort current transmission.

CECRXFREETM (0xA08E) WO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-						
WR	RXFREESBE	CECRXFREETM[6:0]						

RXFREESBE Bus-Free Receive Switch Enable bit. If this bit is set, the system returns to IDLE mode and prepares for the next receive automatically.

CECRXFREETM[6:0] CEC Receive Bus Free Time.

When CEC bus keeps high for the period of “8 × CECRXFREETM[6:0] × CTU”, BUSFREERX is set.

CECSTLDTTX (0xA08F) WO (0x00)

	7	6	5	4	3	2	1	0
RD	-							
WR	CECSTLDTTX[7:0]							

CECSTLDTTX[7:0] configures dedicated TXD start low duty. This is in CTU unit.

CECBFCTRL (0xA097) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	BUSFREETX	EINTTX	BUSFREERX	EINTRX	INTTXFG	INTRXFG	STRBERR	HCKPRS
WR	-	EINTTX	BUSFREERX	EINTRX	INTTXFG	INTRXFG	STRBERR	HCKPRS

BUSFREEETX CEC TXD Bus Free Flag bit.
This bit is set when the TXD Bus Free Time reach “8 x CECTXFREETM[7:0] x CTU”, and is cleared by hardware when detect a CEC falling.

EINTTX CEC TXD Bus Free Interrupt Enable bit. (1: Enable / 0: Disable)

BUSFREERX CEC RXD Bus Free Flag bit.
This bit is set when the RXD bus free time reach “8 x CECRXFREETM[6:0] x CTU”, and is cleared by writing “0” or any CEC bus transition.

EINTRX CEC RXD Bus Free Interrupt Enable bit. (1: Enable / 0: Disable)

INTTXFG CEC TXD Bus Free Interrupt Flag bit.
This bit is set when the TXD bus free time reach “8 x CECTXFREETM[7:0] x CTU”. It must be cleared by writing “0” on the bit.

INTRXFG CEC RXD Bus Free Interrupt Flag bit.
This bit is set when the RXD bus free time reach “8 x CECRXFREETM[6:0] x CTU”. It must be cleared by writing “0” on the bit.

STRBERR CEC Start Bit Error Flag
This bit is set when CEC low period reach “CECSTLDTY[7:0] x CTU” during the START transmission. It must be cleared by writing “0” on the bit.

HCKPRS CEC Clock Pre-Scale Resolution Selection
0: CTU = SYSCLK / (CECCKPRS[7-0] x 2)
1: CTU = SYSCLK / (CECCKPRS[7-0])

9.1 CEC Initiator

While configured as an initiator, the CEC controller provides users with the following features:

SENDBYTE associated with a status report and interrupt option
SENDFIRSTBYTE associated with a status report and interrupt option of the first byte
SENDNOTIFICATION associated with an interrupt option of notification

9.1.1 Send Byte

This function transmits 9 bit-data onto the CEC line, and strobe san Acknowledge bit into an internal user-access register. During the transmission, the CEC engine concurrently verifies if the electronic value of the CEC line is the same as the value being transmitted, and finally marks a status flag indicating success or failure of transmission. An interrupt is issued to the CPU.

The following registers are used by SENDBYTE procedure in sequence.

CECTXD[7:0] - CEC Data Transmission. It is the leading 8 bits of data to be transmitted.

EOM - the 9th bit transmitted which indicates the EOM(End Of Message)

CECBPRD[7:0] - CEC Bit Period Register. the bit period based on the CEC clock.

CEC0LDUTY[7:0] - the forced LOW duty indicating logical “0”.

CEC1LDUTY[7:0] - the forced LOW duty indicating logical “1”.

CECSTRBPNT[7:0] – CEC Strobe Point. It is the relative strobe point during a bit period which is invoked by the CEC engine while it is cross verifying the out-sending value and the responded value from the CEC line.

The procedure needs to check the following values for the return status.

ACK - the 10th bit received from the CEC line which indicates Acknowledge bit for CEC Specification.

TXIF – Transmission Complete Flag. a flag which indicates the CEC engine finishes the byte transmission. It is updated when the 10th bit ends.

TXERR - Transmission Error. It is a flag which indicates verification failure during byte transmission. The cause may be the CEC line which is interfered by invalid devices or noises, or come from arbitration procedure errors.

The procedure is controlled and initiated by the following register bits.

TXMODE – Transmission Mode Enable bit. It is used to set the CEC engine to work as a transmitter. By users' definition, the CEC engine can be as a transmitter if set, or as a receiver if cleared

SENDBYTE – Byte Sending Enable bit. It is the trigger to initiate the byte-sending procedure. After the user sets this bit, the CEC engine copies the CECTXD[7:0] into an internal shift register, and starts transmitting data from the shift register. The user can update the CECTXD as soon as the trigger is enabled.

ETXI – End of Transmission Interrupt bit. It is an interrupt enabling bit that generates an interrupt as soon as the CEC engine completes a transmission.

TXCONT - Continuous Transmission Enable bit. If this bit is set, the CEC engine automatically copies CECTXD[7:0] into the active shift register, and starts contiguous byte transmission unless one of the following conditions occurs.

((ACK==1) && (DADR[3:0] != 1111_b)) || ((ACK==0)&& (DADR[3:0] == 1111_b))

(EOM==1)

(TXERR==1)

This mechanism is helpful in deducing the latency between two contiguous bytes. The user should fill the next byte into CECTXD[7:0] as soon as SENDBYTE is triggered.

9.1.2 Send First Byte

This procedure performs the following functions and is used to obtain arbitrations of the bus.

Test if CEC line is free for a specific duration.

Send START waveform followed with 4-bit data (initiator address). If no error happens, continue to send the other 4-bit data (follower address) and the bit EOM, also strobes Acknowledge bit from the CEC line into ACK and sets interrupt flag TXIF.

The procedure uses the following registers for operations.

CECSTPRD[7:0] - It is the period of the Start bit based on CEC engine clock period.

CECSTLDUTY[7:0] - the forced LOW duty of the Start bit.

CECTXD[7:0] – CEC Data Transmission. It is the leading 8 bits of data to be transmitted.

EOM - the 9th bit transmitted which indicates the EOM (End Of Message)

CECBPRD[7:0] - CEC Bit Period Register. the bit period based on the CEC clock.

CEC0LDUTY[7:0] - the forced LOW duty indicating logical "0".

CEC1LDUTY[7:0] - the forced LOW duty indicating logical "1".

CECSTRBPNT[7:0] – CEC Strobe Point. It is the relative strobe point during a bit period which is invoked by the CEC engine while it is cross verifying the out-sending value and the responded value from the CEC line.

The procedure needs to check the following values for the return status.

ACK - the 10^{th} bit received from the CEC line which indicates Acknowledge bit for CEC Specification.

TXIF – Transmission Complete Flag. a flag which indicates the CEC engine finishes the byte transmission. It is updated when the 10^{th} bit ends.

TXERR - Transmission Error. It is a flag which indicates verification failure during byte transmission. The cause may be the CEC line which is interfered by invalid devices or noises, or come from arbitration procedure errors.

The procedure is controlled and initiated by the following register bits.

TXMODE – Transmission Mode Enable bit. It is used to set the CEC engine to work as a transmitter. By users' definition, the CEC engine can be as a transmitter if set, or as a receiver if cleared

SENDBYTE – Byte Sending Enable bit. It is the trigger to initiate the byte-sending procedure. After the user sets this bit, the CEC engine copies the CECTXD[7:0] into an internal shift register, and starts transmitting data from the shift register. The user can update the CECTXD as soon as the trigger is enabled.

ETXI – End of Transmission Interrupt bit. It is an interrupt enabling bit that generates an interrupt as soon as the CEC engine completes a transmission.

9.1.3 Send Notification

This function forces the CEC line LOW for a specific duration and generates an interrupt after releasing the CEC line. It is designed to implement a NOTIFICATION activity.

The procedure uses the following registers for operations.

CECNTYPRD[7:0] – CEC Notification Period Register. It is the period of the NOTIFICATION after which the CEC engine sets the interrupt flag.

CECNTYLDUTY[7:0] - CEC forced LOW duty of the NOTIFICATION.

The procedure needs to check the following values for the return status.

TXIF - Transmission Complete Flag. It is a flag which indicates the CEC engine completes the NOTIFICATION transmission.

The procedure is controlled and initiated by the following register bits.

SENDNTY – Send Notification bit. It is the trigger to initiate the NOTIFICATION-sending procedure.

ETXI – End of Transmission Interrupt bit. It is the interrupt-enabling bit while the CEC engine completes a transmission.

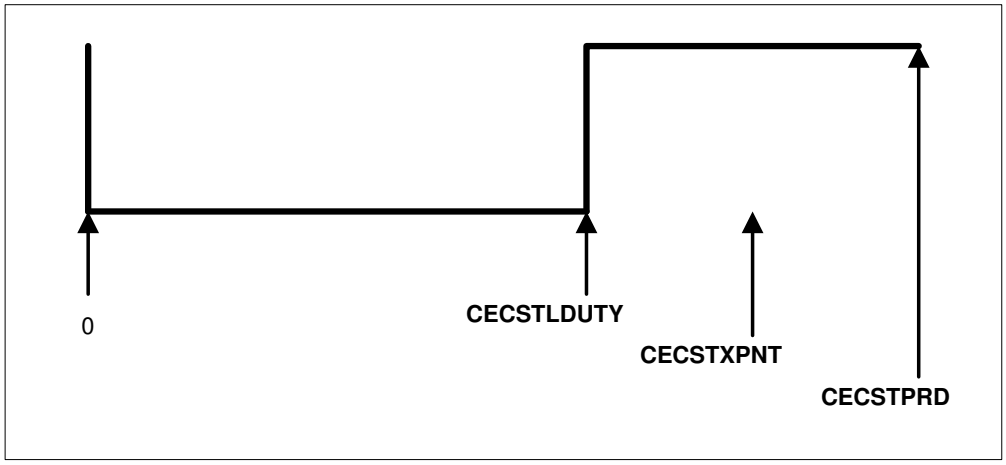
9.2 CEC Follower

While configured as a follower, the CEC controller provides user with the following capabilities.

START DETECTION
BYTE RECEIPTION
LINE ERROR DETECTION

9.2.1 Start Detection

The waveform is taken as a Start bit is shown in the following diagram - if it keeps LOW from the leading falling edge to CECSTLDUTY[7:0] or keeps HIGH impendency from STXPNT[7:0] to CECSTPRD[7:0]



The following registers are used for START detection.

CECSTPRD[7:0] – CEC Start Period

CECSTLDUTY[7:0] –CEC forced LOW duty of the Start bit

STXPNT[7:0] – CEC Start Transmission Point. CEC Start Uncertainty Time

The procedure needs to check the following flags for the return status.

STIF – CEC Start Transmission Flag. It is a flag which indicates as the CEC engine detects a Start bit.

The procedure is controlled and initiated by the following register bits.

TXMODE - Transmission Mode Enable bit. It is used to set the CEC engine to work as a transmitter. By users' definition, the CEC engine can be as a transmitter if set,or as a receiver if cleared

ESTI - Start Transmission Interrupt Enable bit. Interrupt-enabling bit for STIF. .

9.2.2 Byte Reception

This is for receive byte condition as a follower. After a byte is received then an ACK is replied to the initiator and an interrupt is generated.

The following registers are used for BYTE RECEPTIONT.

CECSTRBPNT[7:0] - It is the strobe point from the last falling edge while the CEC engine is reading a bit.

CECBPRD[7:0] - It is the period for coming acknowledge respond.

ACK - It is the bit to be echoed onto the CEC line after the CEC engine receives 9 data bits.

CEC0LDUTY[7:0] - It is the forced LOW duty while respond condition ACK==0.

CEC1LDUTY[7:0] - It is the forced LOW duty while respond condition ACK==0.

The procedure needs to check the following flags for the return status.

CECRXD[7:0] - It is the leading 8-bit data received from CEC line.

EOM - It is the 9th data bit received from the CEC line.

RXIF - It is a flag which indicates the CEC engine has 9 data bits and transmits ACK out.

The procedure is controlled and initiated by the following register bits.

TXMODE - Transmission Mode Enable bit. It is used to set the CEC engine to work as a transmitter. By users' definition, the CEC engine can be as a transmitter if set, or as a receiver if cleared.

ERXI – Receiving Complete Enable bit. It is the interrupt-enabling bit to RXIF.

9.2.3 Line Error Detection

This is used to detect bus line error. The line error is defined as the following two conditions.

The duration between two contiguous falling edges is shorter than specific time after the CEC engine detects a START pattern.

After the CEC engine detects a START pattern and gets a falling edge, it cannot achieve the next falling edge for more than a specific period of time, except the falling edge is numbered as the 10th falling edge.

The following registers are used for byte reception: .

MINEVDUR[7:0] - It means the minimum duration between two events (falling edges). If any two contiguous falling edges get shorter duration than MINEVDUR[7:0], an error flag LINERRIF is set.

TMOUTLMT[7:0] - It means the maximum duration from a falling edge to the next falling edge. After the CEC engine detects a START pattern and gets a falling edge, another coming falling edge is expected by TMOUTLMT[7:0], or timeout status is assumed and an error flag TMOUTIF is set.

The procedure needs to check the following flags for the return status.

LINERRIF - Line Error flag.

TMOUTIF - Time-out flag.

The procedure is controlled and initiated by the following register bits.

ELNERRI - Line Error Interrupt Enable bit. It is the interrupt-enabling gate of the LINERRIF.

ETMOUTI - Time-out Interrupt Enable bit. It is the interrupt-enabling gate of the TMOUTIF.

9.3 STOP Mode Switch Back

MFCFGP4.0[2] - CEC switches back from STOP mode enable bit on pad P4.0

PINT0EN[2] - Enable bit of CEC switch back function

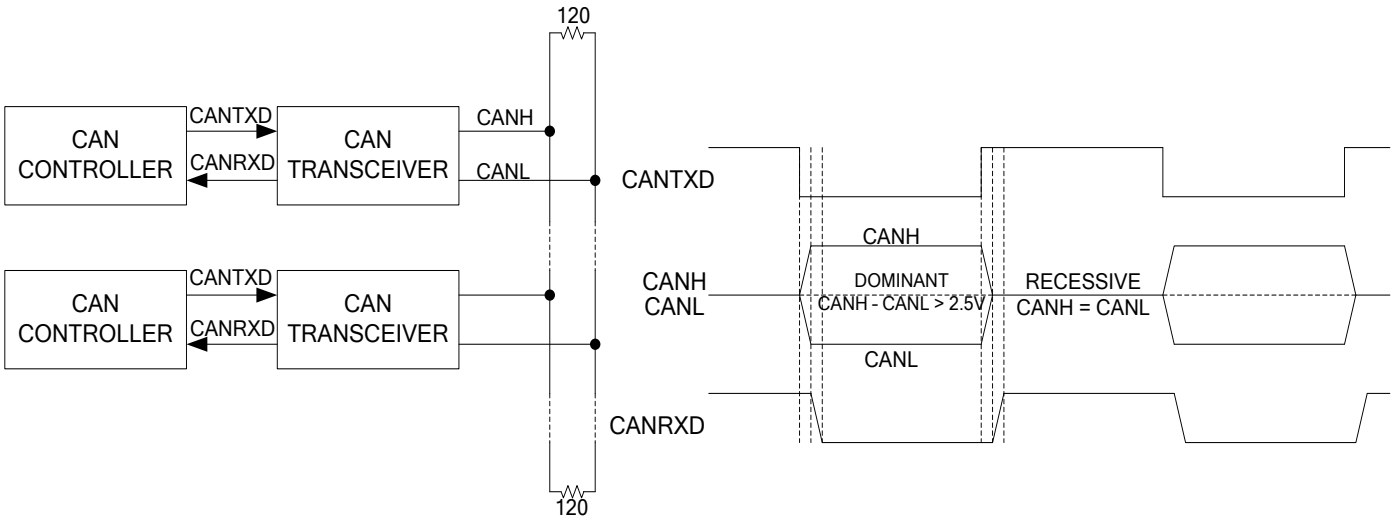
If these 2 bits are set to 1, a "low" on P4.0(CEC line) turns on the IOSC in STOP mode and makes the CPU exit STOP mode.

10. CAN Controller

10.1 Brief Introduction to CAN

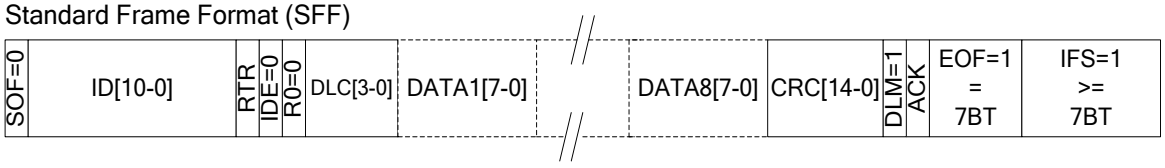
The Controller Area Network (CAN) is a serial asynchronous bus allowing multi-master communications. The CAN bus uses a single terminated twisted pair with maximum data rate of 1MBps. Typical length of the bus can be up to 40 meters. One unusual aspect of the CAN protocol is its message based nature. Nodes on the bus do not have specific address instead the messages have unique identifiers which are used for determining its priority. Each node depending on its functionality transmits specific types of messages and also responds to specific related messages. CAN is especially popular in connecting electronic control modules, sensors, actuators in automotive and industrial applications because its simple bus structure and fault-tolerant capability from extensive error checking.

The physical media of typical CAN bus is a twisted pair terminated with 120Ohm on both ends. The twisted pair has two signals, CANH and CANL. CANH and CANL forms a 5V pseudo differential signal thus have reliable transmissions even under high noise environment. There are two possible logic states on the bus. Dominant state (logic 0) is when CANH is actively pulled to 5V and CANL is actively pulled to 0V. Recessive state (logic 1) is when the bus is not actively driven and both CANH and CANL are pulled toward 2.5V by the termination resistors. The bus also exhibits the wire-AND characteristics, that is bus is in recessive state if only if all nodes are recessive, and bus is in dominant state if one or more nodes are dominant. Because of this pseudo differential nature, it is possible to maintain correct data transmission even if one of the wires is broken. To drive the CAN bus, typically an external transceiver is used to provide isolated drive.

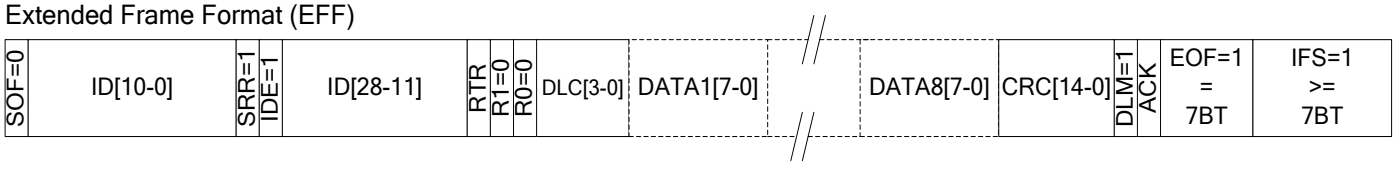


The CAN protocol is a modified version of Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) similar to Ethernet. If two nodes are trying to send at the same time, instead of collision avoidance, the ID field of message will resolve itself which allows the higher priority message to continue and lower priority message been postponed. The typical standard CAN frame format and extended format are shown in the following.

Standard Frame Format (SFF)



Extended Frame Format (EFF)



The frame starts with a bit 0 of Start of Frame (SOF), and followed by an 11-bit ID field. Because the dominant state transmitted by one node will overwrite any other nodes transmitting recessive state, the message ID with lowest number will win the bus arbitration. After arbitration, RTR bit indicate whether this is a transmit message (RTR=0), or a Remote Transmission Request (RTR=1) message. R0 and R1 are two reserve bits which should both be 0. The Data Length Code (DLC) indicates the byte count of following data. DLC must be less than 9. After data field, a 15-bit CRC is appended, and a CRC delimiter is the last bit of CRC field. The receive node will acknowledge the successful reception by sending 0 during the ACK slot. An ACK delimiter and along with minimum 7 bit time of EOF field is

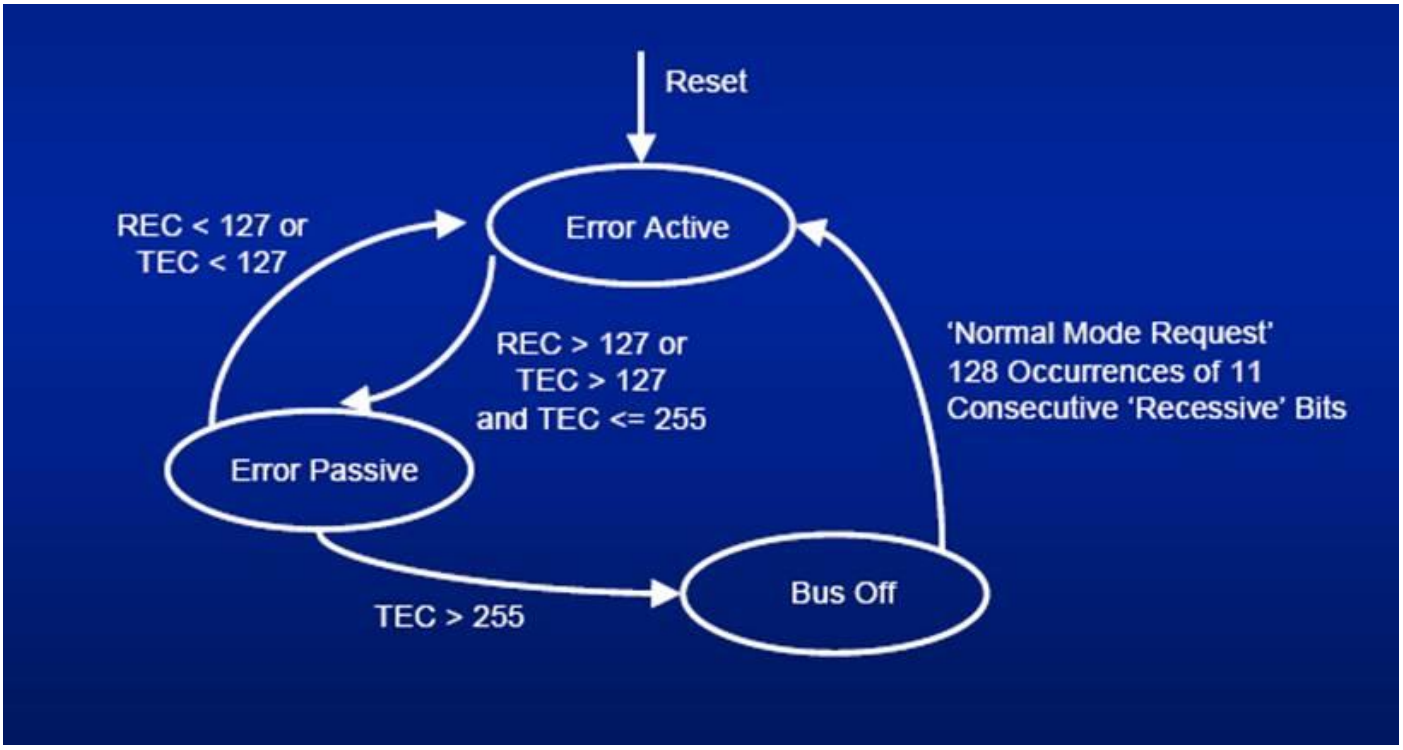
ensured for the bus to enter into bus IDLE state. And a minimum 3 BT (typically implemented as 7 BT) of recessive state and bus IDLE state should be used for inter-frame spacing.

The message format can be further categorized into standard (SFF) or extended format (EFF). The difference in SFF and EFF is in IDE bit. If IDE=dominant, the message is SFF, and if IDE=recessive, the message is EFF. The extended format provides 29-bit ID addressing whit the additional 18 bit ID following the IDE bit. In terms of purpose of message, the message can be considered as either data frame or remote frame. A data frame carries data sent by the transmitter to receivers (plural here). A remote frame is transmitted by a node to request other nodes to respond with data frame using same message ID. Both data frame and remote frame can take either standard or extended format.

CAN bus uses NRZ coding. To ensure enough edges for synchronization, Bit-Stuffing is used. An opposite polarity bit is inserted after 5 consecutive bits of the same level. Bit-Stuffing starts with SOF and ends with CRC (before CRC delimiter). The initial falling edge of SOF provides a hard synchronization on the start of the message.

CAN bus include many error checking mechanisms to ensure reliable operations. This includes CRC checking built-in in the message frame. CRC is calculated over the non-stuffed bit stream data starting with SOF and ends with last bit of data field. In case of CRC mismatch, the frame is discarded, and ACK bit is not responded. Nodes on the bus will also detect frame format error, bit stuffing errors. All these error been detected, a node will transmit an error frame. The error frame consists of 6 consecutive dominant bits as error flag and may be extended to 12 bits when multiple nodes are responding to the same error. This violation of bit-stuffing (more than 5 consecutive same bits) will ensure all nodes on the bus will detect the error and the error flag thus might be further extended up 12 bit times by those nodes detecting the same error. After error flag, the error frame is terminated with minimum 3 bit time of recessive state as error delimiter. Another special frame with the same format as error frame is the overload frame which is sent by a receiving node in response to a remote request message when this node can not meet the response requirement. Overload frame is also sent by nodes when detecting a violation of frame intermission rule (each frame must be separated by 3 bit time of recessive bits).

There are three fundamental states for error handling in each CAN node. This is shown in the following state diagram. REC is receive error count, and TEC is transmit error count.



The node is in normal operation when in Error Active state. In this state, the node can send all frames including error frame (active error flag with six consecutive dominant bits). When error count reaches 127, it enters Error Passive states, and the node can send all frames except error frame. When error count exceeds 255, the node enters Bus Off state and is isolated from the bus. This mechanism ensures a faulty node does cause bus dead-lock. The exit of Bus Off state can only be done by re-initialization or if auto recovery is enabled when 128 occurrence of 11 consecutive recessive bit.

10.2 Features of CAN Controller

The CAN controller is compatible with CAN 2.0A and 2.0B standards. The controller can be configured as normal operating mode or listen-only mode. A self-test mode can also be enabled to perform self loop-back test of the

CAN controller through external loopback. The receiver includes four acceptance filters that are used for ID filtering. The matched messages are stored in receive FIFO shared with CPU XRAM. The transmitter includes a 13-byte transmit buffer in XFR consisting of frame information, 4 bytes of message ID, and 8 bytes of message data. The transmitter also includes three dispatchers. The dispatcher is used for automatically transmit a pre-determined message at a fixed programmable time interval without CPU intervention. Each dispatcher also include a 13-byte transmit buffer in XFR similar to the main transmit path.

After reset, the CAN controller is put in reset mode with all state-machines forced in the initialization states. The user must exit this reset mode by setting CANRST=0 to enter into operating modes.

10.3 Register Map

The registers for CAN Controller resides from 0xAB00 to 0xACFF in XFR space. Some of the registers have different read/write permissions as well as definition under normal mode and silent mode.

ADDR	Operating Mode		Reset Mode	
	read	write	read	write
AB00H	Mode Register	Mode Register	Mode Register	Mode Register
AB01H	00H	Command Register	00H	——
AB02H	Status Register	——	Status Register	——
AB03H	Interrupt Register	——	00H	——
AB04H	Interrupt Enable Register	Interrupt Enable Register	Interrupt Enable Register	Interrupt Enable Register
AB05H	CAN Receive Buffer Start Address[12:5]	CAN Receive Buffer Start Address[12:5]	CAN Receive Buffer Start Address[12:5]	CAN Receive Buffer Start Address[12:5]
AB06H	CAN Receive Buffer Stop Address[12:5]	CAN Receive Buffer Stop Address[12:5]	CAN Receive Buffer Stop Address[12:5]	CAN Receive Buffer Stop Address[12:5]
AB07H	Current Write Address [11:4]	——	Current Write Address [11:4]	——
AB08H	Current Write Address [12]	——	Current Write Address [12]	——
AB09H	Bus Timing Register 0	——	Bus Timing Register 0	Bus Timing Register 0
AB0AH	Bus Timing Register 1	——	Bus Timing Register 1	Bus Timing Register 1
AB0BH | AB0EH	Reserved			
AB0FH	Filter Enable Register 0	——	Filter Enable Register 0	Filter Enable Register 0
AB10H	Arbitration Lost Capture Register	——	Arbitration Lost Capture Register	——
AB11H	Error Code Capture Register	——	Error Code Capture Register	——
AB12H	Error Warning Limit Register	——	Error Warning Limit Register	Error Warning Limit Register
AB13H	Receive Error Counter Register	——	Receive Error Counter Register	Receive Error Counter Register
AB14H	Transmit Error Counter Register	——	Transmit Error Counter Register	Transmit Error Counter Register
AB15H		——		
AB16H		——		——
AB17H		Transmit Frame Information Register		——
AB18H		Transmit ID Register 3		——

ADDR	Operating Mode		Reset Mode	
	read	write	read	write
AB19H		Transmit ID Register 2		——
AB1AH		Transmit ID Register 1		——
AB1BH		Transmit ID Register 0		——
AB1CH		Transmit DATA Register 1		——
AB1DH		Transmit DATA Register 2		——
AB1EH		Transmit DATA Register 3		——
AB1FH		Transmit DATA Register 4		——
AB20H		Transmit DATA Register 5		——
AB21H		Transmit DATA Register 6		——
AB22H		Transmit DATA Register 7		——
AB23H		Transmit DATA Register 8		——
AB24H	——	——	Dispatch 1 Time Register 3 (Note5)	Dispatch 1 Time Register 3
AB25H	——	——	Dispatch 1 Time Register 2	Dispatch 1 Time Register 2
AB26H	——	——	Dispatch 1 Time Register 1	Dispatch 1 Time Register 1
AB27H	Filter Status Register 0	——	Dispatch 1 Time Register 0	Dispatch 1 Time Register 0
AB28H			Filter 0 Acceptance Code 3	Filter 0 Acceptance Code 3
AB29H			Filter 0 Mask Code 3	Filter 0 Mask Code 3
AB2AH			Filter 0 Acceptance Code 2	Filter 0 Acceptance Code 2
AB2BH			Filter 0 Mask Code 2	Filter 0 Mask Code 2
AB2CH			Filter 0 Acceptance Code 1	Filter 0 Acceptance Code 1
AB2DH			Filter 0 Mask Code 1	Filter 0 Mask Code 1
AB2EH			Filter 0 Acceptance Code 0	Filter 0 Acceptance Code 0
AB2FH			Filter 0 Mask Code 0	Filter 0 Mask Code 0
AB30H			Filter 1 Acceptance Code 3	Filter 1 Acceptance Code 3
AB31H			Filter 1 Mask Code 3	Filter 1 Mask Code 3
AB32H			Filter 1 Acceptance Code 2	Filter 1 Acceptance Code 2
AB33H			Filter 1 Mask Code 2	Filter 1 Mask Code 2
AB34H			Filter 1 Acceptance Code 1	Filter 1 Acceptance Code 1

ADDR	Operating Mode		Reset Mode	
	read	write	read	write
AB35H			Filter 1 Mask Code 1	Filter 1 Mask Code 1
AB36H			Filter 1 Acceptance Code 0	Filter 1 Acceptance Code 0
AB37H			Filter 1 Mask Code 0	Filter 1 Mask Code 0
AB38H			Filter 2 Acceptance Code 3	Filter 2 Acceptance Code 3
AB39H			Filter 2 Mask Code 3	Filter 2 Mask Code 3
AB3AH			Filter 2 Acceptance Code 2	Filter 2 Acceptance Code 2
AB3BH			Filter 2 Mask Code 2	Filter 2 Mask Code 2
AB3CH			Filter 2 Acceptance Code 1	Filter 2 Acceptance Code 1
AB3DH			Filter 2 Mask Code 1	Filter 2 Mask Code 1
AB3EH			Filter 2 Acceptance Code 0	Filter 2 Acceptance Code 0
AB3FH			Filter 2 Mask Code 0	Filter 2 Mask Code 0
AB40H			Filter 3 Acceptance Code 3	Filter 3 Acceptance Code 3
AB41H			Filter 3 Mask Code 3	Filter 3 Mask Code 3
AB42H			Filter 3 Acceptance Code 2	Filter 3 Acceptance Code 2
AB43H			Filter 3 Mask Code 2	Filter 3 Mask Code 2
AB44H			Filter 0 Acceptance Code 1	Filter 3 Acceptance Code 1
AB45H			Filter 3 Mask Code 1	Filter 3 Mask Code 1
AB46H			Filter 3 Acceptance Code 0	Filter 3 Acceptance Code 0
AB47H			Filter 3 Mask Code 0	Filter 3 Mask Code 0
AB48H | AC27H			Reserved	Reserved
AC28H	Dispatch Time 2 Register 1	——	Dispatch Time 2 Register 1	Dispatch Time 2 Register 1
AC29H	Dispatch Time 2 Register 0	——	Dispatch Time 2 Register 0	Dispatch Time 2 Register 0
AC2AH | AC2FH	Reserved	Reserved	Reserved	Reserved
AC30H	Dispatch 1 Frame Information Register	Dispatch 1 Frame Information Register	Dispatch 1 Frame Information Register	Dispatch 1 Frame Information Register

ADDR	Operating Mode		Reset Mode	
	read	write	read	write
AC31H	Dispatch 1 ID Register 3	Dispatch 1 ID Register 3	Dispatch 1 ID Register 3	Dispatch 1 ID Register 3
AC32H	Dispatch 1 ID Register 2	Dispatch 1 ID Register 2	Dispatch 1 ID Register 2	Dispatch 1 ID Register 2
AC33H	Dispatch 1 ID Register 1	Dispatch 1 ID Register 1	Dispatch 1 ID Register 1	Dispatch 1 ID Register 1
AC34H	Dispatch 1 ID Register 0	Dispatch 1 ID Register 0	Dispatch 1 ID Register 0	Dispatch 1 ID Register 0
AC35H	Dispatch 1 DATA Register 1	Dispatch 1 DATA Register 1	Dispatch 1 DATA Register 1	Dispatch 1 DATA Register 1
AC36H	Dispatch 1 DATA Register 2	Dispatch 1 DATA Register 2	Dispatch 1 DATA Register 2	Dispatch 1 DATA Register 2
AC37H	Dispatch 1 DATA Register 3	Dispatch 1 DATA Register 3	Dispatch 1 DATA Register 3	Dispatch 1 DATA Register 3
AC38H	Dispatch 1 DATA Register 4	Dispatch 1 DATA Register 4	Dispatch 1 DATA Register 4	Dispatch 1 DATA Register 4
AC39H	Dispatch 1 DATA Register 5	Dispatch 1 DATA Register 5	Dispatch 1 DATA Register 5	Dispatch 1 DATA Register 5
AC3AH	Dispatch 1 DATA Register 6	Dispatch 1 DATA Register 6	Dispatch 1 DATA Register 6	Dispatch 1 DATA Register 6
AC3BH	Dispatch 1 DATA Register 7	Dispatch 1 DATA Register 7	Dispatch 1 DATA Register 7	Dispatch 1 DATA Register 7
AC3CH	Dispatch 1 DATA Register 8	Dispatch 1 DATA Register 8	Dispatch 1 DATA Register 8	Dispatch 1 DATA Register 8
AC3DH	Dispatch 1 Status Register	Dispatch 1 Control Register	Dispatch 1 Status Register	Dispatch 1 Control Register
AC3EH | AC3FH	Reserved	Reserved	Reserved	Reserved
AC40H	Dispatch 2 Frame Information Register	Dispatch 2 Frame Information Register	Dispatch 2 Frame Information Register	Dispatch 2 Frame Information Register
AC41H	Dispatch 2 ID Register 3	Dispatch 2 ID Register 3	Dispatch 2 ID Register 3	Dispatch 2 ID Register 3
AC42H	Dispatch 2 ID Register 2	Dispatch 2 ID Register 2	Dispatch 2 ID Register 2	Dispatch 2 ID Register 2
AC43H	Dispatch 2 ID Register 1	Dispatch 2 ID Register 1	Dispatch 2 ID Register 1	Dispatch 2 ID Register 1
AC44H	Dispatch 2 ID Register 0	Dispatch 2 ID Register 0	Dispatch 2 ID Register 0	Dispatch 2 ID Register 0
AC45H	Dispatch 2 DATA Register 1	Dispatch 2 DATA Register 1	Dispatch 2 DATA Register 1	Dispatch 2 DATA Register 1
AC46H	Dispatch 2 DATA Register 2	Dispatch 2 DATA Register 2	Dispatch 2 DATA Register 2	Dispatch 2 DATA Register 2
AC47H	Dispatch 2 DATA Register 3	Dispatch 2 DATA Register 3	Dispatch 2 DATA Register 3	Dispatch 2 DATA Register 3
AC48H	Dispatch 2 DATA Register 4	Dispatch 2 DATA Register 4	Dispatch 2 DATA Register 4	Dispatch 2 DATA Register 4
AC49H	Dispatch 2 DATA Register 5	Dispatch 2 DATA Register 5	Dispatch 2 DATA Register 5	Dispatch 2 DATA Register 5

ADDR	Operating Mode		Reset Mode	
	read	write	read	write
AC4AH	Dispatch 2 DATA Register 6	Dispatch 2 DATA Register 6	Dispatch 2 DATA Register 6	Dispatch 2 DATA Register 6
AC4BH	Dispatch 2 DATA Register 7	Dispatch 2 DATA Register 7	Dispatch 2 DATA Register 7	Dispatch 2 DATA Register 7
AC4CH	Dispatch 2 DATA Register 8	Dispatch 2 DATA Register 8	Dispatch 2 DATA Register 8	Dispatch 2 DATA Register 8
AC4DH	Dispatch 2 Status Register	Dispatch 2 Control Register	Dispatch 2 Status Register	Dispatch 2 Control Register
AC4EH | AC4FH	Reserved	Reserved	Reserved	Reserved
AC50H	Dispatch 3 Frame Information Register	Dispatch 3 Frame Information Register	Dispatch 3 Frame Information Register	Dispatch 3 Frame Information Register
AC51H	Dispatch 3 ID Register 3	Dispatch 3 ID Register 3	Dispatch 3 ID Register 3	Dispatch 3 ID Register 3
AC52H	Dispatch 3 ID Register 2	Dispatch 3 ID Register 2	Dispatch 3 ID Register 2	Dispatch 3 ID Register 2
AC53H	Dispatch 3 ID Register 1	Dispatch 3 ID Register 1	Dispatch 3 ID Register 1	Dispatch 3 ID Register 1
AC54H	Dispatch 3 ID Register 0	Dispatch 3 ID Register 0	Dispatch 3 ID Register 0	Dispatch 3 ID Register 0
AC55H	Dispatch 3 DATA Register 1	Dispatch 3 DATA Register 1	Dispatch 3 DATA Register 1	Dispatch 3 DATA Register 1
AC56H	Dispatch 3 DATA Register 2	Dispatch 3 DATA Register 2	Dispatch 3 DATA Register 2	Dispatch 3 DATA Register 2
AC57H	Dispatch 3 DATA Register 3	Dispatch 3 DATA Register 3	Dispatch 3 DATA Register 3	Dispatch 3 DATA Register 3
AC58H	Dispatch 3 DATA Register 4	Dispatch 3 DATA Register 4	Dispatch 3 DATA Register 4	Dispatch 3 DATA Register 4
AC59H	Dispatch 3 DATA Register 5	Dispatch 3 DATA Register 5	Dispatch 3 DATA Register 5	Dispatch 3 DATA Register 5
AC5AH	Dispatch 3 DATA Register 6	Dispatch 3 DATA Register 6	Dispatch 3 DATA Register 6	Dispatch 3 DATA Register 6
AC5BH	Dispatch 3 DATA Register 7	Dispatch 3 DATA Register 7	Dispatch 3 DATA Register 7	Dispatch 3 DATA Register 7
AC5CH	Dispatch 3 DATA Register 8	Dispatch 3 DATA Register 8	Dispatch 3 DATA Register 8	Dispatch 3 DATA Register 8
AC5DH	Dispatch 3 Status Register	Dispatch 3 Control Register	Dispatch 3 Status Register	Dispatch 3 Control Register
AC5EH | AC5FH	Reserved	Reserved	Reserved	Reserved
AC60H | AC90H	Reserved			
AC91H	Loop Test Enable	Loop Test Enable	Loop Test Enable	Loop Test Enable
AC92H	Loop Test	Loop Test	Loop Test	Loop Test

ADDR	Operating Mode		Reset Mode	
	read	write	read	write
AC93H | ACFFH	Reserved			

10.4 Configuration, Control and Status Registers

CANMODE (0xAB00) CAN Mode Configuration Register R/W (0x01)

	7	6	5	4	3	2	1	0
RD	-	-	-	DPP	ERAR	STE	LOM	RSTM
WR	-	-	-	DPP	ERAR	STE	LOM	RSTM

DPP — Dispatch Priority

DPP=1 will set higher priority for dispatch packet over transmit packet

DPP=0 will set higher priority for transmit packet over dispatch packet

ERAR — Error Auto Recovery

ERAR=1 will allow auto recovery from bus off state

When this bit is set, auto recovery from Bus Off state to Error Active state is enabled. The auto recovery condition is met when 128 occurrences of bus-free time (11 consecutive 'recessive' bits) is received. When exiting the Bus Off state, TEC and REC are cleared to 0 by hardware if ERAR=1.

ERAR=0 for disable the auto recovery function.

If ERAR=0, software must clear REC and TEC to exit of Bus Off state to Error Active state. Please note in the entry of Bus Off state, if ERAR=0, RSTM will be set by hardware, so modification of REC and TEC is allowed. After clearing REC and TEC, the software must also clear RSTM bit to return to normal operation.

STE — Self Test Enable

STE=1 enables self-test mode.

When STE=1, the only difference from normal operation mode is that all ACK errors are ignored.

STE=0 for normal mode operation

LOM — Listen-Only Mode

LOM=1 will set the CAN controller as listen-only mode

When LOM=1, CAN controller will work in Listen-Only mode. In this mode, the CAN controller will only perform receiver functions and does not engage any transmission (including ACK signaling). All other functions can be used like in Operating mode. In this mode, the error counters are stopped at the current value and message transmission is not possible. But when detecting bus error, listen-only node will still transmit error frame.

LOM=0 for normal mode operation

RSTM — Reset Mode

RSTM=1 will force the CAN Controller into rest mode regardless of other setting, and will initialize all state machine. It does not affect the register contents. RSTM defaults to 1 after system reset. RSTM is set to 1 by hardware when entering into Bus-Off state if ERAR=0.

RSTM must be cleared to 0 to allow the CAN Controller into normal operation.

Please note RSTM must be set to 1 first in order to modify DPP, ERAR, STE, and LOM. The write cycle of RSTM must be preceding the modification cycles of DPP, ERAR, STE, and LOM.

CANCMD (0xAB01) CAN Command Register WO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	OFR	SRR	CDO	RRB	AT	TR
WR	-	-	OFR	SRR	CDO	RRB	AT	TR

OFR — Overload Frame Request

OFR=1 will request transmitting an overload frame. The overload frame is transmitted the first bit time of next expected INTERMISSION. This bit is self cleared by hardware after the overload frame is transmitted. Since the overload frame will be transmitted regardless of the bus condition, no interrupt is generated at the completion.

SRR — Self Receive Request

SRR=1 will result a message be transmitted and received simultaneously. Upon self reception request command, a message is transmitted and simultaneously received if the acceptance filter is set to the corresponding identifier. This command is used mainly for self test of the controller. If STE=1 which enables the self test mode, the transmit and receive can be successful without other nodes present. This bit is self cleared by hardware when transmit and receive operations are completed and both transmit and receive interrupts will be generated.

CDO — Clear Data Overrun

CDO=1 will clear data overrun status bit. This command bit is used to clear the data overrun condition indicated by the data overrun status bit. This command bit is self cleared by hardware.

RRB — Reset Receive Buffer Status

RRB=1 will clear the Receive Buffer Status Bit in CANSTAT register. RRB command must be used to clear RBS after reading out the received messages from XRAM. This bit is self cleared by hardware.

AT — Abort Transmitting

AT=1 will abort the any pending transmission request. AT does not stop an on-going message transmission. But the next transmission attempt will be aborted. The completion of the abortion will cause TCS=1 in CANSTAT register. AT should be used only for aborting a message transmission with errors (mostly in ACK error). As AT can not guarantee to stop an on-going message transmission, the software should check Bus Error Flag (BEI) bit and Arbitration Lost Flag (ALI) when TCS=1. TCS=1 can be resulted by either abortion or success of transmission. If BEI=0 and ALI=0, then the message is sent successfully even abort command is issued. Otherwise the transmission was actually aborted because of bus error or arbitration lost.

TR — Transmit Request

TR=1 will start the transmit process of the message in the transmit buffer. Before setting TR=1, the software should ensure if the previous message has been transmitted successfully by checking TCS bit. It is also necessary to check TCS bit after TR command.

This register is write-only and reading will return 0x00.

CANSTAT (0xAB02) CAN Status Register RO (0x0C)

	7	6	5	4	3	2	1	0
RD	BOS	ES	TS	RS	TCS	TBS	DOS	RBS
WR	-	-	-	-	-	-	-	-

BOS — Bus Off State Status

BOS=1 indicates the CAN controller is in the Bus Off State.

BOS=0 indicates the CAN controller is involved in bus activities.

Bus Off state is defined in the error handling state diagram. Possible exit of Bus Off state is by Bus Auto recovery, or by writing a value less than 255 into TEC under reset mode.

ES — Error Status

ES=1 indicates at lease one of the error counters has reached or exceeded the warning limit defined by the Error Warning Limit register.

ES=0 indicates warning limits are not exceeded.

Errors detected during reception or transmission will affect the error counters according to the CAN 2.0B protocol specification. ES is set when at least one of the error counters has reached or exceeded the warning limit setting (EWLR). An error warning interrupt is generated, if enabled. The default value of EWLR after hardware reset is 96.

TS — Transmit Status

TS=1 indicates CAN controller is transmitting a message.

RS — Receive Status

RS=1 indicates CAN controller is receiving a message.

If both TS and RS are 0, the CAN controller is idle.

TCS — Transmit Complete Status

TCS=1 indicates last requested message transmission has been successfully completed.

TCS is cleared to 0 when TR or SRR command is issued. And TCS is set to 1 by hardware when the message is sent successfully. TCS remains to be 0 during the message transmission, and TCS is set to 1 by hardware if the message is sent successfully. Please note TCS=1 can also be resulted by AT command.

TBS — Transmit Buffer Status

TBS=1 indicates the transmit buffer is free and a new message can be written.

TBS=0 indicates the transmit buffer is locked and the content in the transmit buffer is either waiting for transmission or in the process of being transmitted. And no new message should be written. If software writes into transmit buffer when TBS=0, the write will disrupt the transmission and the written byte will be discarded.

DOS — Data Overrun Status

DOS=1 indicates a received message was lost due to RXFIFO overrun. DOS=1 also indicates the receive buffer has reached the STOP address and will return to START address. When a message matches the acceptance filter, the received message is stored into RXFIFO defined by CANSTAR and CANSTOP. Each valid received message occupies 16-byte space of the memory buffer and the RXFIFO pointer will increase 16 automatically. If RXFIFO pointer reaches the CANSTOP, it will restart to CANSTAR and DOS is set to 1. And an overrun interrupt is generated with DOI flag set.

DOS must be cleared by setting CDO (Clear Data Overrun) command bit or by setting to reset mode.

RBS — Receive Buffer Status.

RBS=1 indicates one or more valid messages are available in RXFIFO.

RBS must be cleared by setting the RRB (Reset Receive Buffer Status) command bit or by setting to reset mode.

This register is read-only, corresponding bits are self-cleared by hardware when the condition is not met or the register is read except DOS and RBS.

CANINTF (0xAB03) CAN Interrupt Flag Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	BEI	ALI	EPI	BOI	DOI	EI	TI	RI
WR	-	-	-	-	-	-	-	-

This is a read-only register. The corresponding interrupt enable bit must be enabled for individual flag to be effective. All flags (except BOI) are event change triggering so only the change of the condition will cause the interrupt and corresponding flag to be set. All bits are cleared after read except BOI.

BEI — Bus Error Interrupt Flag

BEI is set when the CAN controller detects error(s) on the CAN-bus. The error types include format error, bit stuff error, bit error (when data or CRC transmitted does not match with received), CRC error, and ACK error. The BEIE bit must be set to enable the BEI function.

ALI — Arbitration Lost Interrupt

ALI is set when a message was not sent successfully due to loss of arbitration. The ALIE bit must be set to enable the ALI function.

EPI Error Passive Interrupt
EPI is set when the CAN controller enters the error passive state, or the CAN controller is already in the error passive state and enters the error active state. EPIE bit must be set to enable EPI function.

BOI Bus Off State Interrupt
BOI is set when the CAN controller in the bus off state or equivalently BOS=1. Since BOI and BOS is 1 whenever in Bus Off state, the software must take care to disable BOIE when entering BOI ISR and then correct the BOS status otherwise the interrupt will occur indefinitely. BOIE bit must be set to enable BOI function.

DOI Data Overrun Interrupt
DOI is set when writing pointer turn around from CANSTOP address to CANSTAR address or equivalently when DOS changes from 0 to 1. DOIE bit must be set to enable DOI function.

EI Error Warning Interrupt
EI is set on when REC or TEC is greater or equal to the CANEWLR setting or equivalently when ES changes from 0 to 1. EIE bit must be set to enable EI function.

TI Transmit Interrupt
TI is set whenever the transmit buffer becomes free or equivalently the transmission of the message has finished and TBS changes from 0 to 1. TIE bit must be set to enable TI function.

RI Receive Interrupt
RI is set when a new message has been put into the RXFIFO by the CAN controller. RI is cleared when read. RIE bit must be set to enable RI function.

CANIE (0xAB04) CAN Interrupt Enable Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	BEIE	ALIE	EPIE	BOIE	DOIE	EIE	TIE	RIE
WR	BEIE	ALIE	EPIE	BOIE	DOIE	EIE	TIE	RIE

The interrupt enable register allows indicating different types of interrupt source. The register appears to the CPU as a read/write memory.

BEIE Bus Error Interrupt Enable
BEI=1 enables bus error interrupt

ALIE Arbitration Lost Interrupt Enable
ALIE=1 enables arbitration loss interrupt

EPIE Error Passive Interrupt Enable

BOIE Bus Off Interrupt Enable

DOIE Data Overrun Interrupt Enable

EIE Error Warning Interrupt Enable

TIE Transmit Interrupt Enable

RIE Receive Interrupt Enable

The receive FIFO (RXFIFO) is shared with XRAM of the CPU. The CAN controller uses DMA to store the received message into the XRAM. The DMA is implemented in cycle stealing from CPU access to the XRAM, thus does not affect execution of CPU.

CANSTAR (0xAB05) CAN Receive Buffer Start Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CANSTAR[12-5]							
WR	CANSTAR[12-5]							

Indicate the starting address of the XRAM for receive FIFO. CANSTAR[12-5] maps to XRAM address [12-5]. Therefore the receive FIFO is allocated in 32-byte increment. While CANSTAR is changed, hardware always updates the Start Address into Current Write Address register to point to the next valid buffer address.

CANSTOP (0xAB06) CAN Receive Buffer Stop Address Register R/W (0x00)

	7	6	5	4	3	2	1	0
RD	CANSTOP[12-5]							
WR	CANSTOP[12-5]							

Indicate the stop address of the XRAM for receive FIFO. CANSTOP[12-5] maps to XRAM address [12-5]. The allocation is in 32-byte increment so the total number of frames can be stored is 2*(CANSTOP[12-5] – CANSTAR[12-5] + 1) since each frame occupies 16 bytes. The software must ensure CANSTOP is greater then CANSTAR for proper operations if the RXFIFO. When the internal write pointer turns around to the CANSTAR, the overrun status and the overrun interrupt will be issued if enabled.

CANCWARL (0xAB07) CAN Receive Buffer Current Write Pointer Low Address Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	CANCWAR[11-4]							
WR	-							

CANCWARH (0xAB08) CAN Receive Buffer Current Write Pointer High Address Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	-	CANCWAR[12]
WR	-	-	-	-	-	-	-	-

CANCWAR[12-4] points to the next frame address for received data storing in 16-byte boundary size. The default value of CANCWAR is same as the Receive Buffer Start Address (CANSTAR). After receive a valid frame, the CANCWAR[12-4] is incremented by 1 by hardware until CANCWAR equals CANSTOP. If CANCWAR[4]=1 and CANCWAR[12-5] equals to CANSTOP[12-5], the CANCWAR[12-5] will return to CANSTAR[12-5] and clear CANCWAR[4] by receiving another frame and data overrun flag and interrupt is generated.

CANBTR0 (0xAB09) CAN Bus Timing Register 0 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	SJW[1-0]		CANCS[5-0]					
WR	SJW[1-0]		CANCS[5-0]					

CANBR[5-0] CANBR[5-0] defines the CAN Controller Clock
The CAN controller clock CANCLK = SYSCLK/2/(CANCS[5-0]+1). This defines the CAN Time Quanta from the system clock.

SJW[1-0] Synchronization Jump Length
This defines the maximum adjustment in CANCLK period units for re-synchronization. The receiver uses the falling edges to determine the physical delay of the network. The synchronization is achieved by inserting a delay compensation segment in the bit time after SYNC SEG and thus also delays the sampling points of the local receiver. SJW[1-0]+1 is used to set the maximum allowed delay segment in CANCLK period.

CANBTR1 (0xAB0A) CAN Bus Timing Register 1 R/W (0x00)

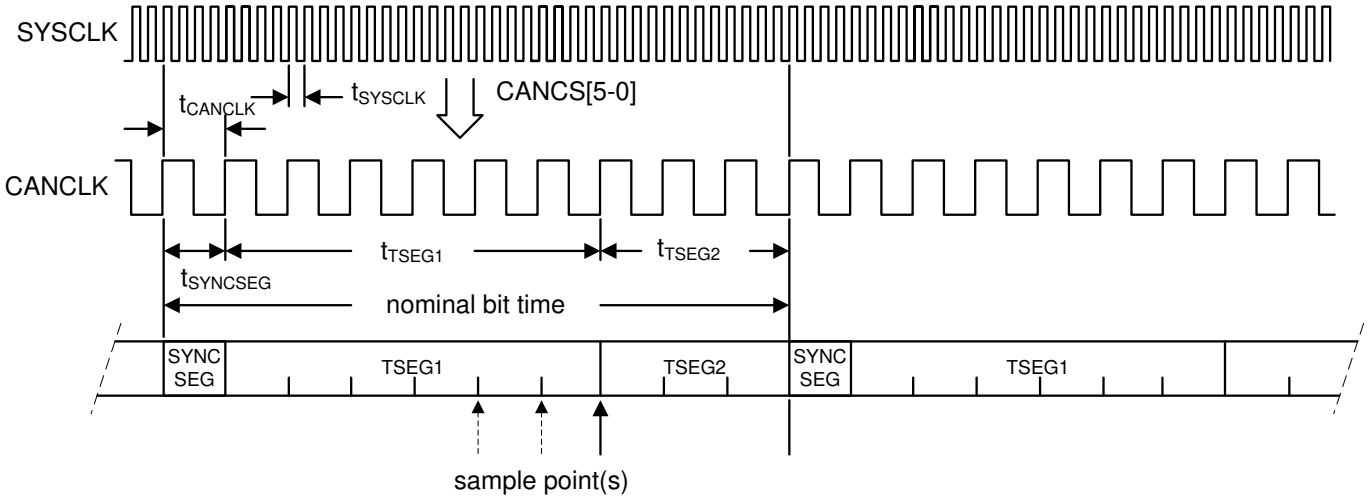
	7	6	5	4	3	2	1	0
RD	SAM	TSEG2[2-0]			TSEG1[3-0]			
WR	SAM	TSEG2[2-0]			TSEG1[3-0]			

SAM SAM=1 will use triple-sampling method. The triple-sampling method uses three instances between TSEG1 and TSEG2 as shown in the following timing diagram. The data is determined by the majority of the sampled results.
SAM=0 will use single-sampling method. The single sampling occurs at the time instance between TSEG1 and TSEG2

TSEG2[2-0] TSEG2[2-0] defines the bit time component 2. TSEG2 = TSEG2[2-0] + 1.

TSEG1[3-0] TSEG1[3-0] defines the bit time component 1. TSEG1 = TSEG1[3-0] + 1.

A CAN bit time is partitioned by CANCLK periods as the following diagram. The first CAN CLK period is used for edge synchronization, and TSEG1 is used for defining the sampling edge. For SAM=0, the sampling edge is the first edge of TSEG2. For SAM=1, the sampling edges are the last two edges of TSEG1, and the first edge of TSEG2. For typical CAN bus design, TSEG1 and TSEG2 is organized for sampling at the last two quanta (period of CANCLK).



CAN baud rate is thus can be calculated as CANCLK/((TSEG2[2-0]+1)+(TSEG1[3-0]+1)+1). Or in terms of SYSCLK, SYSCLK/2/(CANCS[5-0]+1)/{(TSEG2[2-0]+1)+(TSEG1[3-0]+1)+1}. Also note, with re-synchronization, a delay compensation segment is inserted between SYNC SEG and TSEG1. The maximum length of delay compensation segment is limited by SJW[1-0].

CANFTER (0xAB0F) CAN ID Filter Enable Register 1 R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	FTER3	FTER2	FTER1	FTER0
WR	-	-	-	-	FTER3	FTER2	FTER1	FTER0

There are three Acceptance Filters for message ID matching. This register controls the enabling of each filter, multiple filters can be enabled. This register can be modified only in reset mode and appears as read-only in normal operating mode.

FTER3	Acceptance Filter 3 Enable FTER3=1 enables Acceptance Filter 3
FTER2	Acceptance Filter 2 Enable FTER2=1 enables Acceptance Filter 2
FTER1	Acceptance Filter 1 Enable FTER1=1 enables Acceptance Filter 1
FTER0	Acceptance Filter 0 Enable FTER0=1 enables Acceptance Filter 0

CANALC (0xAB10) CAN Arbitration Lost Capture Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	ALC4	ALC3	ALC2	ALC1	ALC0
WR	-	-	-	-	-	-	-	-

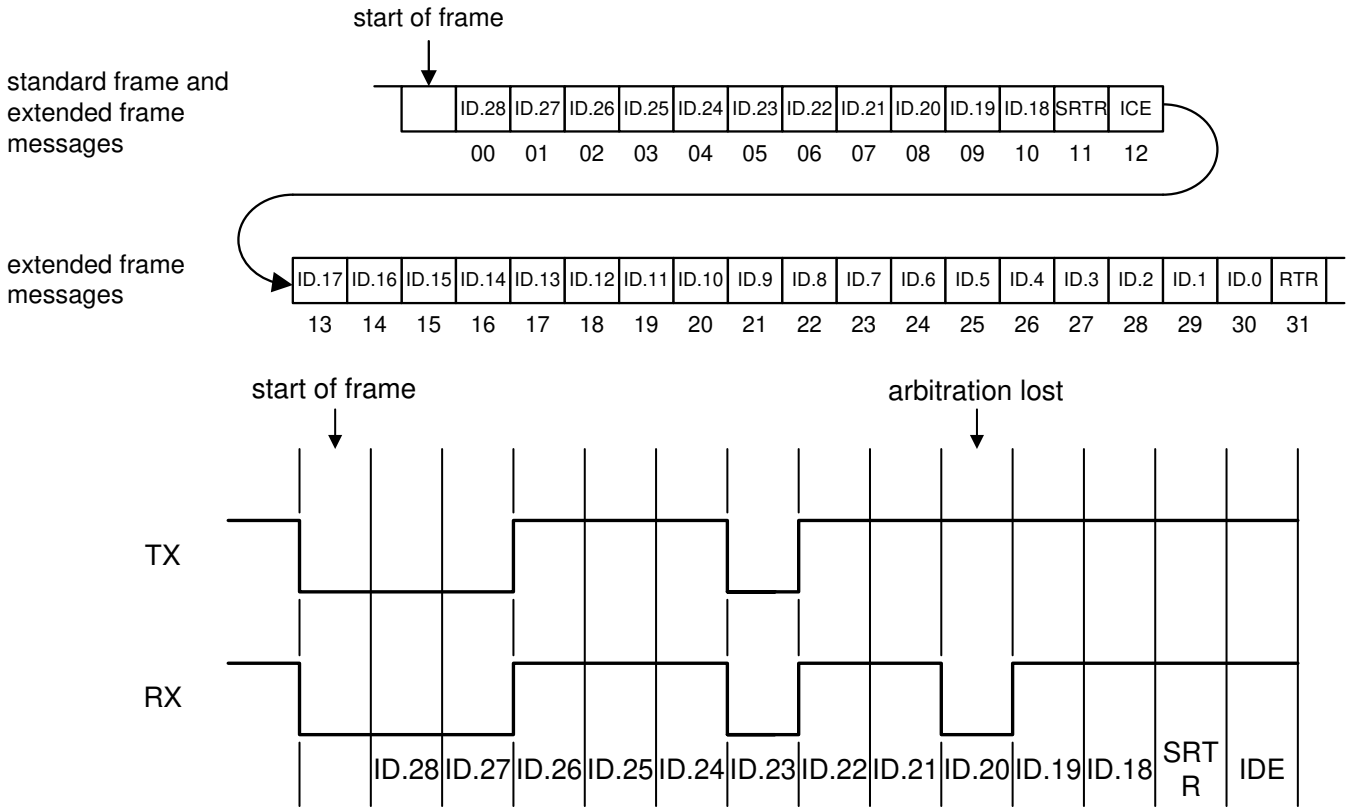
This register is read only and reports the bit locations of a lost arbitration when transmitting a message. The following table defines the corresponding relationship.

Bits					Decimal Value	Description
ALC4	ALC3	ALC2	ALC1	ALC0		
0	0	0	0	0	0	Arbitration lost in bit ID28
0	0	0	0	1	1	Arbitration lost in bit ID27
0	0	0	1	0	2	Arbitration lost in bit ID26
0	0	0	1	1	3	Arbitration lost in bit ID25
0	0	1	0	0	4	Arbitration lost in bit ID24
0	0	1	0	1	5	Arbitration lost in bit ID23

Bits					Decimal Value	Description
ALC4	ALC3	ALC2	ALC1	ALC0		
0	0	1	1	0	6	Arbitration lost in bit ID22
0	0	1	1	1	7	Arbitration lost in bit ID21
0	1	0	0	0	8	Arbitration lost in bit ID20
0	1	0	0	1	9	Arbitration lost in bit ID19
0	1	0	1	0	10	Arbitration lost in bit ID18
0	1	0	1	1	11	Arbitration lost in bit SRTR (Note 1)
0	1	1	0	0	12	Arbitration lost in bit IDE
0	1	1	0	1	13	Arbitration lost in bit ID17 (Note 2)
0	1	1	1	0	14	Arbitration lost in bit ID16 (Note 2)
0	1	1	1	1	15	Arbitration lost in bit ID15 (Note 2)
1	0	0	0	0	16	Arbitration lost in bit ID14 (Note 2)
1	0	0	0	1	17	Arbitration lost in bit ID13 (Note 2)
1	0	0	1	0	18	Arbitration lost in bit ID12 (Note 2)
1	0	0	1	1	19	Arbitration lost in bit ID11 (Note 2)
1	0	1	0	0	20	Arbitration lost in bit ID10 (Note 2)
1	0	1	0	1	21	Arbitration lost in bit ID9 (Note 2)
1	0	1	1	0	22	Arbitration lost in bit ID8 (Note 2)
1	0	1	1	1	23	Arbitration lost in bit ID7 (Note 2)
1	1	0	0	0	24	Arbitration lost in bit ID6 (Note 2)
1	1	0	0	1	25	Arbitration lost in bit ID5 (Note 2)
1	1	0	1	0	26	Arbitration lost in bit ID4 (Note 2)
1	1	0	1	1	27	Arbitration lost in bit ID3 (Note 2)
1	1	1	0	0	28	Arbitration lost in bit ID2 (Note 2)
1	1	1	0	1	29	Arbitration lost in bit ID1 (Note 2)
1	1	1	1	0	30	Arbitration lost in bit ID0 (Note 2)
1	1	1	1	1	31	Arbitration lost in bit RTR (Note 2)

Note 1: Bit RTR for standard frame messages.
Note 2: Extended frame messages only.

On arbitration lost, the corresponding arbitration lost interrupt is generated. At the same time, the current bit position of the bit stream processor is captured into the Arbitration Lost Capture register. The content in this register is locked until the software read out its contents. The capture mechanism is then activated again. The corresponding interrupt flag located in the interrupt register is cleared during the read access to the interrupt flag register. A new arbitration lost interrupt is not possible until the arbitration lost capture register is read out. The following diagram shows the bit locations of the arbitration lost. And an ALC=0x08 is shown as an example, where arbitration lost is detected when outputting a recessive bit but receiving a dominant bit at ID20.



CANECC (0xAB11) CAN Error Code Capture Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	ERRC[1]	ERRC[0]	DIR	SEG[4]	SEG[3]	SEG[2]	SEG[1]	SEG[0]
WR	-	-	-	-	-	-	-	-

This register is read only. This register always keeps the last error status on CAN bus.

ERRC[1-0] Error Code

This reflects the error type as defined in the following table

ERCC[1]	ERRC[0]	Descriptions
0	0	Bit error
0	1	Form error
1	0	Stuff error
1	1	Other type of error

DIR Direction of Transfer Error

DIR=1 means errors occurred during reception

DIR=0 means errors occurred during transmission

SEG[4-0] Error Locations

SEG[4-0]					Function
0	0	0	1	1	Start of frame
0	0	0	1	0	ID28 to ID21
0	0	1	1	0	ID20 toID18
0	0	1	0	0	Bit SRTR
0	0	1	0	1	Bit IDE
0	0	1	1	1	ID17 to ID13
0	1	1	1	1	ID12 to ID5
0	1	1	1	0	ID4 to ID0
0	1	1	0	0	Bit RTR
0	1	1	0	1	Reserved bit 1
0	1	0	0	1	Reserved bit 0
0	1	0	1	1	Data length code
0	1	0	1	0	Data field
0	1	0	0	0	CRC sequence
1	1	0	0	0	CRC delimiter
1	1	0	0	1	Acknowledge slot
1	1	0	1	1	Acknowledge delimiter
1	1	0	1	0	End of frame
1	0	0	1	0	Intermission
1	0	0	0	1	Active error flag
1	0	1	1	0	Passive error flag
1	0	0	1	1	Tolerate dominant bits (Note)
1	0	1	1	1	Error delimiter
1	1	1	0	0	Overload flag

Note: Any node tolerates up to 7 consecutive 'dominant' bits after send Active Error Flag, Passive Error Flag or Overload Flag.

If a bus error occurs, the corresponding bus error interrupt is generated. At the same time, the current position of the bit stream processor is captured into the Error Code Capture register. The content in this register is locked until the software read out its content. A new bus error capture is prohibited until the capture register is read. The capture mechanism is then activated again. The corresponding interrupt flag located in the interrupt flag register is cleared by reading of the interrupt flag register.

CANEWLR (0xAB12) CAN Error Warning Limit Register RW (0x60)

	7	6	5	4	3	2	1	0
RD	CANEWLR[7-0]							
WR	CANEWLR[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. This register set the warning limit for the error counter. When CANREC or CANTEC value exceed CANEWLR value, an error warning interrupt is generated, and the EI flag is set in the interrupt flag register. CANEWLF defaults to 0x60 after hardware reset.

CANREC (0xAB13) CAN Receive Error Count Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANREC[7-0]							
WR	CANREC[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. CANREC contains the current error count. When the controller enters bus-off state, CANREC is re-initialized as 0, and during bus-off, the writing to CANREC has no effect.

CANTEC (0xAB13) CAN Transmit Error Count Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTEC[7-0]							
WR	CANTEC[7-0]							

This register is modifiable on in reset mode and appears as read-only in normal operating mode. CANTEC contains the current error count, and is initialized as 0 after hardware rest. If a bus-off event occurs, the transmit error counter is initialized to 128 if auto-recovery is not enabled. The software can just use the reset mode to allow recovery to the normal error-active state.

10.5 Transmit Buffer

The global layout of the transmit buffer in XFR is shown in figure below. The transmit buffer contains 13-bytes of XFR space can be in the form of Standard Frame Format (SFF) or Extended Frame Format (EFF) configuration.

Standard Format Frame

AB17h	TX Frame Information
AB18h	TX Identifier 4
AB19h	TX Identifier 3
AB1Ah	TX Identifier 2
AB1Bh	TX Identifier 1
AB1Ch	TX Data 1
AB1Dh	TX Data 2
AB1Eh	TX Data 3
AB1Fh	TX Data 4
AB20h	TX Data 5
AB21h	TX Data 6
AB22h	TX Data 7
AB23h	TX Data 8

Extended Format Frame

AB17h	TX Frame Information
AB18h	TX Identifier 4
AB19h	TX Identifier 3
AB1Ah	TX Identifier 2
AB1Bh	TX Identifier 1
AB1Ch	TX Data 1
AB1Dh	TX Data 2
AB1Eh	TX Data 3
AB1Fh	TX Data 4
AB20h	TX Data 5
AB21h	TX Data 6
AB22h	TX Data 7
AB23h	TX Data 8

The transmit buffer layout is subdivided into descriptor and data fields where the first byte of the descriptor field is the frame information byte (frame information). It describes the frame format (SFF or EFF), remote or data frame and the data length. Four identifier bytes for both SFF and EFF messages follow. But only the lower 11 bits in the four identifier bytes are valid for SFF and the upper 21 bits are invalid. For EFF, the lower 29 bits in the four identifier bytes are valid and the upper 3 bits are invalid. The data field contains up to eight data bytes. Note that a direct access to the transmit buffer RAM is possible using the XFR address from AB17H to AB23H.

CANTFIR (0xAB17) CAN Transmit Frame Information Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	FF	RTR	-	-	DLC[3]	DLC[2]	DLC[1]	DLC[0]
WR	FF	RTR	-	-	DLC[3]	DLC[2]	DLC[1]	DLC[0]

FF — Frame Format
FF=1 indicates Extended format frame
FF=0 indicates Standard format frame

RTR — Remote Frame Transmit Request
RTR=1 indicates remote frame
RTR=0 indicates regular data frame

DLC[3-0] — Data Byte Count = DLC[3-0]
The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being logic 1 (remote). This forces the number of transmitted/received data bytes to be 0. Nevertheless, the data length code must be specified correctly to avoid bus errors, if two CAN controllers start a remote frame transmission with the same identifier simultaneously. For reasons of compatibility, no data length code >8 should be used. If a value greater than 8 is used, only 8 bytes of data are transmitted in the data frame with the Data Length Code specified in DLC. For remote frame, DLC[3-0] is used for indicating the requested data length.

In standard Frame Format (SFF) the identifier consists of 11 bits (ID.28 to ID.18) and in Extended Frame Format (EFF) messages the identifier consists of 29 bits (ID.28 to ID.0). ID.28 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message's name and is used for bus arbitration and used in a receiver for acceptance filtering. The smaller the binary value of the identifier is, the higher the priority is. This is due to the larger number of leading dominant bits during arbitration. Also please note the R0, R1 and SRR bits in the SFF and EFF are transmitted as R0=R1=0 and SRR=1.

CANTXID3 (0xAB18) CAN Transmit Identifier Register 3 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID3[7-0]							
WR	CANTXID3[7-0]							

CANTXID3[7-0] — For SFF ID, CANTXID3[7-0] is not used.
For EFF ID, CANTXID3[4-0] = ID[28-24]. CANTXID3[7-5] is don't care.

CANTXID2 (0xAB19) CAN Transmit Identifier Register2 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID2[7-0]							
WR	CANTXID2[7-0]							

CANTXID2[7-0] — For SFF ID, CANTXID2[7-0] is not used.
For EFF ID, CANTXID2[7-0] = ID[23-16].

CANTXID1 (0xAB1A) CAN Transmit Identifier Register 1 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID1[7-0]							
WR	CANTXID1[7-0]							

CANTXID1[7-0] — For SFF ID, CANTXID1[2-0] = ID[28-26]. CANTXID1[7-3] is not used.
For EFF ID, CANTXID1[7-0] = ID[15-8].

CANTXID0 (0xAB1B) CAN Transmit Identifier Register 0 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXID0[7-0]							
WR	CANTXID0[7-0]							

CANTXID0[7-0] For SFF ID, CANTXID0[7-0] = ID[25-18]
For EFF ID, CANTXID0[7-0] = ID[7-0].

Data field registers is mapping to XFR address AB1CH to AB23H. The number of transferred data bytes is defined by the data length code. The first bit transmitted is the most significant bit of data byte 1 at XFR address AB1CH.

CANTXDATA0 (0xAB1C) CAN Transmit DATA Register 1 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA1[7-0]							
WR	CANTXDATA1[7-0]							

CANTXDATA2 (0xAB1D) CAN Transmit DATA Register 2 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA2[7-0]							
WR	CANTXDATA2[7-0]							

CANTXDATA3 (0xAB1E) CAN Transmit DATA Register 3 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA3[7-0]							
WR	CANTXDATA3[7-0]							

CANTXDATA4 (0xAB1F) CAN Transmit DATA Register 4 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA4[7-0]							
WR	CANTXDATA4[7-0]							

CANTXDATA5 (0xAB20) CAN Transmit DATA Register 5 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA5[7-0]							
WR	CANTXDATA5[7-0]							

CANTXDATA6 (0xAB21) CAN Transmit DATA Register 6 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA6[7-0]							
WR	CANTXDATA6[7-0]							

CANTXDATA7 (0xAB22) CAN Transmit DATA Register 7 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA7[7-0]							
WR	CANTXDATA7[7-0]							

CANTXDATA8 (0xAB23) CAN Transmit DATA Register 8 RW (0x00)

	7	6	5	4	3	2	1	0
RD	CANTXDATA8[7-0]							
WR	CANTXDATA8[7-0]							

10.6 Acceptance Filter

There are four Acceptance Filters for matching with the ID of a message on the bus. Each filter can be enabled or disabled independently. Each acceptance filter is consisted of 4 ID registers and corresponding ID Mask register to mask the match comparing the specific bit in the ID. The layout of the ID registers corresponding to forming ID28 to ID1 for SFF and EFF is the same as Transmit ID registers. Please note the ID registers only defines the ID and do not define the frame type, therefore it is possible a value in the ID registers corresponding to two possible matches for SFF and EFF separately. This is especially true since the ID bit ordering for SFF and EFF is opposite, the user should check if unintended accepted ID is possible for another frame type. For the Filter's Mask register, when mask bit is 0 the corresponding bit is compared for matching, and when the mask bit is 1 the corresponding bit is ignored. All Acceptance Filter ID registers and Mask register can only be modified in the reset mode, and appear as read-only in normal operating mode.

CANDSR (0xAB27) CAN Acceptance Filter Match Status Register RO (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	FMS[3]	FMS[2]	FMS[1]	FMS[0]
WR	-	-	-	-	-	-	-	-

This register is read-only and the corresponding match bit is set to 1 when a match condition occurs for the acceptance filer.

FMS[3] Acceptance Filter 3 Match Status
FSM[3]=1 Acceptance Filter 3 match
FSM[3]=0 Acceptance Filter 3 not match

FMS[2] Acceptance Filter 2 Match Status
FSM[2]=1 Acceptance Filter 2 match
FSM[2]=0 Acceptance Filter 2 not match

FMS[1] Acceptance Filter 1 Match Status
FSM[1]=1 Acceptance Filter 1 match
FSM[1]=0 Acceptance Filter 1 not match

FMS[0] Acceptance Filter 0 Match Status
FSM[0]=1 Acceptance Filter 0 match
FSM[0]=0 Acceptance Filter 0 not match

CANAFID00 (0xAB28) Acceptance Filter 0 ID Byte 0

	7	6	5	4	3	2	1	0
RD	CANFID00[7-0]							
WR	CANFID00[7-0]							

CANAFMK00 (0xAB29) Acceptance Filter 0 Mask Byte 0

	7	6	5	4	3	2	1	0
RD	CANFMK00[7-0]							
WR	CANFMK00[7-0]							

CANAFID01 (0xAB2A) Acceptance Filter 0 ID Byte 1

	7	6	5	4	3	2	1	0
RD	CANFID01[7-0]							
WR	CANFID01[7-0]							

CANAFMK01 (0xAB2B) Acceptance Filter 0 Mask Byte 1

	7	6	5	4	3	2	1	0
RD	CANFMK01[7-0]							
WR	CANFMK01[7-0]							

CANAFID02 (0xAB2C) Acceptance Filter 0 ID Byte 2

	7	6	5	4	3	2	1	0
RD	CANFID02[7-0]							
WR	CANFID02[7-0]							

CANAFMK02 (0xAB2D) Acceptance Filter 0 Mask Byte 2

	7	6	5	4	3	2	1	0
RD	CANFMK02[7-0]							
WR	CANFMK02[7-0]							

CANAFID03 (0xAB2E) Acceptance Filter 0 ID Byte 3

	7	6	5	4	3	2	1	0
RD	CANFID03[7-0]							
WR	CANFID03[7-0]							

CANAFMK03 (0xAB2F) Acceptance Filter 0 Mask Byte 3

	7	6	5	4	3	2	1	0
RD	CANFMK03[7-0]							
WR	CANFMK03[7-0]							

CANAFID10 (0xAB30) Acceptance Filter 1 ID Byte 0

	7	6	5	4	3	2	1	0
RD	CANFID10[7-0]							
WR	CANFID10[7-0]							

CANAFMK10 (0xAB31) Acceptance Filter 1 Mask Byte 0

	7	6	5	4	3	2	1	0
RD	CANFMK10[7-0]							
WR	CANFMK10[7-0]							

CANAFID11 (0xAB32) Acceptance Filter 1 ID Byte 1

	7	6	5	4	3	2	1	0
RD	CANFID11[7-0]							
WR	CANFID11[7-0]							

CANAFMK11 (0xAB33) Acceptance Filter 1 Mask Byte 1

	7	6	5	4	3	2	1	0
RD	CANFMK11[7-0]							
WR	CANFMK11[7-0]							

CANAFID12 (0xAB34) Acceptance Filter 1 ID Byte 2

	7	6	5	4	3	2	1	0
RD	CANFID12[7-0]							
WR	CANFID12[7-0]							

CANAFMK12 (0xAB35) Acceptance Filter 1 Mask Byte 2

	7	6	5	4	3	2	1	0
RD	CANFMK12[7-0]							
WR	CANFMK12[7-0]							

CANAFID13 (0xAB36) Acceptance Filter 1 ID Byte 3

	7	6	5	4	3	2	1	0
RD	CANFID13[7-0]							
WR	CANFID13[7-0]							

CANAFMK13 (0xAB37) Acceptance Filter 1 Mask Byte 3

	7	6	5	4	3	2	1	0
RD	CANFMK13[7-0]							
WR	CANFMK13[7-0]							

CANAFID20 (0xAB38) Acceptance Filter 2 ID Byte 0

	7	6	5	4	3	2	1	0
RD	CANFID20[7-0]							
WR	CANFID20[7-0]							

CANAFMK20 (0xAB39) Acceptance Filter 2 Mask Byte 0

	7	6	5	4	3	2	1	0
RD	CANFMK20[7-0]							
WR	CANFMK20[7-0]							

CANAFID21 (0xAB3A) Acceptance Filter 2 ID Byte 1

	7	6	5	4	3	2	1	0
RD	CANFID21[7-0]							
WR	CANFID21[7-0]							

CANAFMK21 (0xAB3B) Acceptance Filter 2 Mask Byte 1

	7	6	5	4	3	2	1	0
RD	CANFMK21[7-0]							
WR	CANFMK21[7-0]							

CANAFID22 (0xAB3C) Acceptance Filter 2 ID Byte 2

	7	6	5	4	3	2	1	0
RD	CANFID22[7-0]							
WR	CANFID22[7-0]							

CANAFMK22 (0xAB3D) Acceptance Filter 2 Mask Byte 2

	7	6	5	4	3	2	1	0
RD	CANFMK22[7-0]							
WR	CANFMK22[7-0]							

CANAFID23 (0xAB3E) Acceptance Filter 2 ID Byte 3

	7	6	5	4	3	2	1	0
RD	CANFID23[7-0]							
WR	CANFID23[7-0]							

CANAFMK23 (0xAB3F) Acceptance Filter 2 Mask Byte 3

	7	6	5	4	3	2	1	0
RD	CANFMK23[7-0]							
WR	CANFMK23[7-0]							

CANAFID30 (0xAB40) Acceptance Filter 3 ID Byte 0

	7	6	5	4	3	2	1	0
RD	CANFID30[7-0]							
WR	CANFID30[7-0]							

CANAFMK30 (0xAB41) Acceptance Filter 3 Mask Byte 0

	7	6	5	4	3	2	1	0
RD	CANFMK30[7-0]							
WR	CANFMK30[7-0]							

CANAFID31 (0xAB42) Acceptance Filter 3 ID Byte 1

	7	6	5	4	3	2	1	0
RD	CANFID31[7-0]							
WR	CANFID31[7-0]							

CANAFMK31 (0xAB43) Acceptance Filter 3 Mask Byte 1

	7	6	5	4	3	2	1	0
RD	CANFMK31[7-0]							
WR	CANFMK31[7-0]							

CANAFID32 (0xAB44) Acceptance Filter 3 ID Byte 2

	7	6	5	4	3	2	1	0
RD	CANFID32[7-0]							
WR	CANFID32[7-0]							

CANAFMK32 (0xAB45) Acceptance Filter 3 Mask Byte 2

	7	6	5	4	3	2	1	0
RD	CANFMK32[7-0]							
WR	CANFMK32[7-0]							

CANAFID33 (0xAB46) Acceptance Filter 3 ID Byte 3

	7	6	5	4	3	2	1	0
RD	CANFID33[7-0]							
WR	CANFID33[7-0]							

CANAFMK33 (0xAB3F) Acceptance Filter 3 Mask Byte 3

	7	6	5	4	3	2	1	0
RD	CANFMK33[7-0]							
WR	CANFMK33[7-0]							

10.7 Receive FIFO

The receive FIFO is located in the XRAM and the start and stop address is defined by CANSTAR and CANSTOP. The RXFIFO is allocated in 32-bytes increment. Each received frame occupies a 16-byte XRAM space but and can contain up to 14 bytes when data length is 8 bytes. The following diagram shows the frame arrangement.

FRAME START @ 16-BYTE INCREMENT →

LOCATION	DESCRIPTION	BIT [7-0]						
BYTE 0	ACP FILTER	-	-	-	-	-	-	ACPFIL[1-0]
BYTE 1	FORMAT	IDE	RTR	0	0	DLC[3-0]		
BYTE 2	ID3	0	0	0	ID[28-24]			
BYTE 3	ID2	ID[23-16]						
BYTE 4	ID1	ID[15-8]						
BYTE 5	ID0	ID[7-0]						
BYTE 6	DATA1	DATA1[7-0]						
BYTE 7	DATA2	DATA2[7-0]						
BYTE 8	DATA3	DATA3[7-0]						
BYTE 9	DATA4	DATA4[7-0]						
BYTE A	DATA5	DATA5[7-0]						
BYTE B	DATA6	DATA6[7-0]						
BYTE C	DATA7	DATA7[7-0]						
BYTE D	DATA8	DATA8[7-0]						
BYTE E	RESERVED	-						
BYTE F	RESERVED	-						

ACPFIL[1-0] contains the Acceptance Filter number that matches with the received frame. IDE bit determines whether the frame is SFF (IDE=0) or EFF (IDE=1). For SFF, ID[28-11] will be all 0 and only ID[10-0] is meaningful. The received data bytes depend on DLC code. For unspecified data bytes and reserved bytes, the contents are not guaranteed. Please note the R0, R1 and SRR bits along with CRC field are not stored.

10.8 Dispatch

The dispatch function allows the CAN controller to send out periodic dispatch frame without software interactions. There are total three built-in dispatchers. Each dispatcher contains 13-byte dispatch transmit buffer consisting of a transmit frame information register, four transmit ID registers, and 8 transmit data registers. The layout of this dispatch transmit buffer is the same as the transmit buffer. The interval of the dispatch can be defined by either dispatch timers, D1TR (a 28-bit timer), or by D2TR (a 16-bit timer). All these register can be written only in reset mode. The transmit priority is DISPATH1 > DISPATCH2 > DISPATCH3.

CAND1TR3 (0xAB24) CAN Dispatch Interval Timer 1 Byte 3 Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	WP[3-0]				D1TR[27-24]			
WR	WP[3-0]				D1TR[27-24]			

This register can be modified only in reset mode.

WP[3-0] Write Protection Control
WP[3-0] must be written 0110 to write into D1TR[27-0].

D1TR[27-24] Dispatch Timer 1 bit 27 to 24

CAND1TR2 (0xAB25) CAN Dispatch Interval Timer 1 Byte 2 Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CAND1TR[23-16]							
WR	CAND1TR[23-16]							

This register can be modified only in reset mode and WP[3-0] of CAND1TR3 must be written 0110.

D1TR[23-16] Dispatch Timer 1 bit 23 to 16.

CAND1TR1 (0xAB26) CAN Dispatch Interval Timer 1 Byte 1 Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CAND1TR[15-8]							
WR	CAND1TR[15-8]							

This register can be modified only in reset mode and WP[3-0] of CAND1TR3 must be written 0110.

D1TR[15-8] Dispatch Timer 1 bit 15 to 8.

CAND1TR0 (0xAB27) CAN Dispatch Interval Timer 1 Byte 0 Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CAND1TR[7-0]							
WR	CAND1TR[7-0]							

This register can be modified only in reset mode and WP[3-0] of CAND1TR3 must be written 0110.

D1TR[7-0] Dispatch Timer 1 bit 7 to 0.

The interval defined by D1TR is D1TR[27-0] * 100 * BT, where BT is the bit time of CAN bus. The maximum interval is 2^{28} * 100 * 1usec = 26843 sec for CAN baud rate of 1Mbps.

CAND2TR1 (0xAB28) CAN Dispatch Interval Timer 2 Byte 1 Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CAND2TR[15-8]							
WR	CAND2TR[15-8]							

This register can be modified only in reset mode. CAND2TR can be updated only first by writing 0x0000.

D2TR[15-8] Dispatch Timer 2 bit 15 to 8.

CAND2TR0 (0xAB27) CAN Dispatch Interval Timer 2 Byte 0 Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	CAND2TR[7-0]							
WR	CAND2TR[7-0]							

This register can be modified only in Silent mode. CAND2TR can be updated only first by writing 0x0000.

D2TR[7-0] Dispatch Timer 2 bit 7 to 0.

The interval defined by D2TR is D2TR[15-0] * 100 * BT, where BT is the bit time of CAN bus. The maximum interval is 2^{16} * 100 * 1usec = 6.55 sec for CAN baud rate of 1Mbps.

There are total of three dispatchers, Dispatcher A, Dispatcher B and Dispatcher C. Dispatcher A uses CAND1TR for frame interval. Dispatcher B and C can be configured to use either CAND1TR or CAND2TR.

CANDACTL (0xAC3D) CAN Dispatcher A Control/Status Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	DAEN	-	-	-	-	-	-	-
WR	DAEN	-	-	-	-	-	-	DABUSY

DAEN Dispatcher A Enable
DAEN=1 enables the Dispatcher A
DAEN=0 disables the Dispatcher A
DAEN can be modified only if DABUSY is 0. And the dispatch data can be modified only if DAEN is 0.

DABUSY Dispatcher A Busy Status
DABUSY=1 indicates Dispatcher A is on going transmitting.
DABUSY=0 indicates Dispatcher A is idle.

CANDBCTL (0xAC4D) CAN Dispatcher B Control/Status Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	DBEN	DTSEL[1]	DTSEL[0]	-	-	-	-	-
WR	DBEN	DTSEL[1]	DTSEL[0]	-	-	-	-	DBBUSY

DBEN — Dispatcher B Enable
DBEN=1 enables the Dispatcher B
DBEN=0 disables the Dispatcher B
DBEN can be modified only if DBBUSY is 0. And the dispatch data can be modified only if DBEN is 0.

DTSEL[1-0] — Dispatch Timer Select
DTSEL[1-0] = 01 selects Dispatch Timer 1 CAND1TR as interval timer
DTSEL[1-0] = 10 selects Dispatch Timer 2 CAND2TR as interval timer
Other setting of DTSEL is no allowed.

DBBUSY — Dispatcher B Busy Status
DBBUSY=1 indicates Dispatcher B is on going transmitting.
DBBUSY=0 indicates Dispatcher B is idle.

CANDCCTL (0xAC5D) CAN Dispatcher C Control/Status Register RW (0x00)

	7	6	5	4	3	2	1	0
RD	DCEN	DTSEL[1]	DTSEL[0]	-	-	-	-	-
WR	DCEN	DTSEL[1]	DTSEL[0]	-	-	-	-	DCBUSY

DCEN — Dispatcher C Enable
DCEN=1 enables the Dispatcher C
DCEN=0 disables the Dispatcher C
DCEN can be modified only if DCBUSY is 0. And the dispatch data can be modified only if DCEN is 0.

DTSEL[1-0] — Dispatch Timer Select
DTSEL[1-0] = 01 selects Dispatch Timer 1 CAND1TR as interval timer
DTSEL[1-0] = 10 selects Dispatch Timer 2 CAND2TR as interval timer
Other setting of DTSEL is no allowed.

DCBUSY — Dispatcher C Busy Status
DCBUSY=1 indicates Dispatcher C is on going transmitting.
DCBUSY=0 indicates Dispatcher C is idle.

Each Dispatcher contains a 13-byte transmit buffer. The transmit buffer is consisted of a Transmit Frame Information register, 4 bytes of Transmit ID registers, and 8 bytes of Transmit Data registers. These buffers can be modified only under Silent mode and the dispatcher being disabled. The transmit buffer for Dispatcher A, B, and C are located at 0xAC30 to 0xAC3C, 0xAC40 to 0xAC4C, and 0xAC50 to 0xAC5C respectively. The layout is the same as regular transmit buffer as Frame ID, TX ID, and TX DATA.

ACx0h	TX Frame Information
ACx1h	TX Identifier 4
ACx2h	TX Identifier 3
ACx3h	TX Identifier 2
ACx4h	TX Identifier 1
ACx5h	TX Data 1
ACx6h	TX Data 2
ACx7h	TX Data 3
ACx8h	TX Data 4
ACx9h	TX Data 5
ACxAh	TX Data 6
ACxBh	TX Data 7
ACxCh	TX Data 8

10.9 Basic Operations

The basic function of CAN controller will be discussed in this section. The following chapter will describe the advanced functions of CAN controller. To communicate with other nodes on the CAN-bus, CAN controller must be configured properly as described.

10.9.1 Setting of Operation Mode

Some registers are writable only in reset mode such as Bus Timing registers, so we must write these registers in the reset mode and the CAN controller must quit reset mode when these registers are configured properly to make the controller work.

10.9.2 Baud Rate Setting

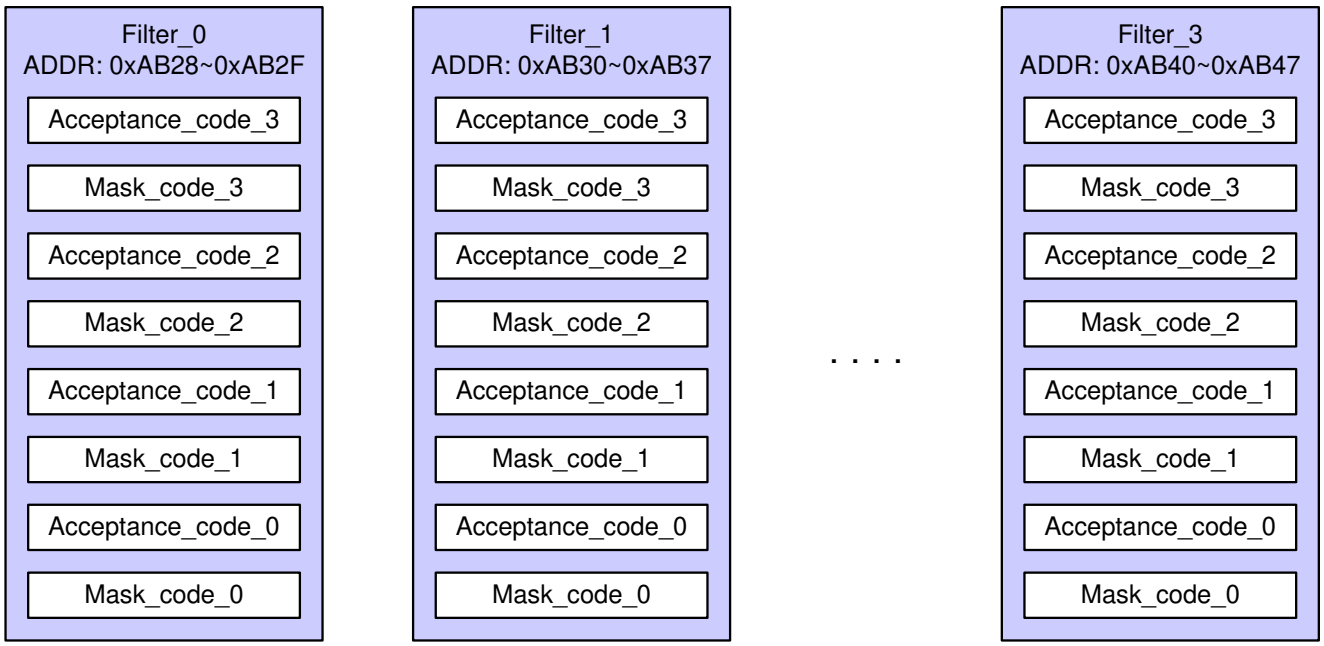
The baud rate in the CAN-bus is determined by the two registers CANBTR0 and CANBTR1. The following example explains how to calculate the baud rate of 500K assuming system clock is 16MHz, and CANBTR0=01h CANBTR1=14h. Then the CANCLK is SYSCLK/2/(CANCS[5-0]+1) = SYSCLK/4 = 4MHz. Thus each time quanta is 250nsec. Then the one CAN bit time BT is (1 + (TSEG1+1) + (TSEG2+1)) x 250 nsec = (1 + 2 + 5) x 250 nsec = 2usec. As the result the CAN bus baud rate is 500Kbps.

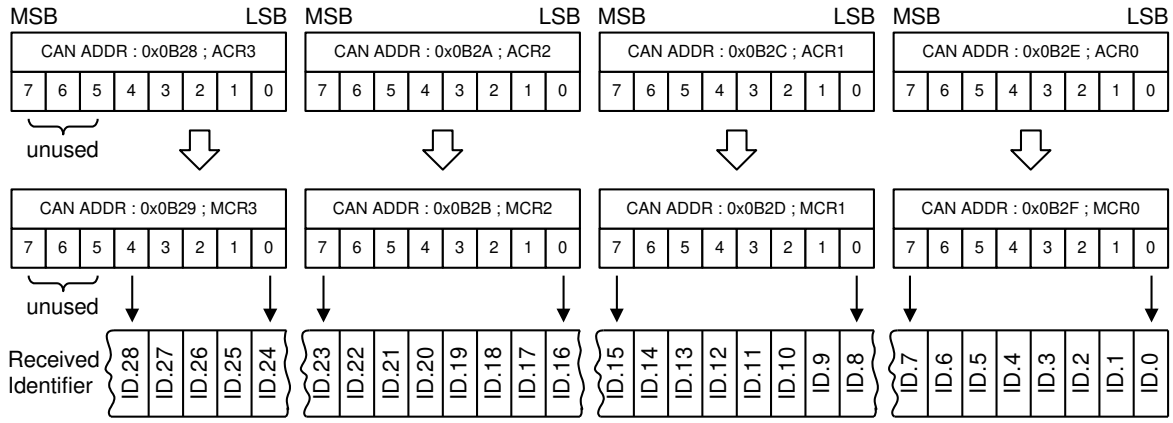
10.9.3 Setting of Acceptance Filter

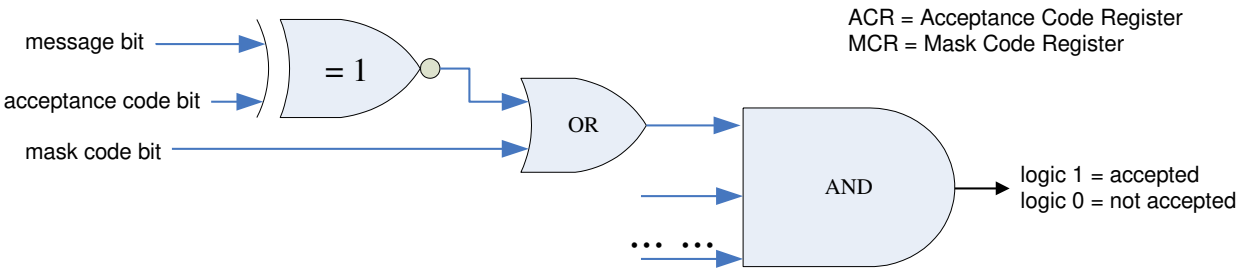
With the help of the acceptance filter, the CAN controller can receive proper frames from CAN-bus only when the identifier bits of the received frames are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the Acceptance Code Registers (ACR) and the Mask Code Registers (MCR). The bit patterns of frames to be received are defined within the Acceptance Code registers. The corresponding mask code registers allow define certain bit positions to be ‘don’t care’. After hardware reset, all the Acceptance Code registers and Mask Code registers are cleared.

In IS31CS8968A, there are 4 groups of acceptance filter. We must enable at least one group of filter previously. Each group of filter has 4 Acceptance Code bits and 4 Mask Code bits and occupies 8 bytes address space. They are mapping to the XFR address space as the Figure below. Acceptance code 3 register is the highest byte in the Acceptance Code; the Acceptance code 0 register is the lowest byte in the Acceptance Code. Mask code 3 register is the highest byte in the Mask Code and Mask Code 0 register is the lowest byte in the Mask Code. The Acceptance Code Register and Mask Code Register distribute crossly in the XFR address space. We take Filter 0 in CAN0 as an example: the address of the Acceptance Code 3 is 0xAB28 and the Mask Code 3 is 0xAB29. The address of the Acceptance Code 2 is 0xAB2A and the Mask Code 2 is 0xB2B and so on. When received message is

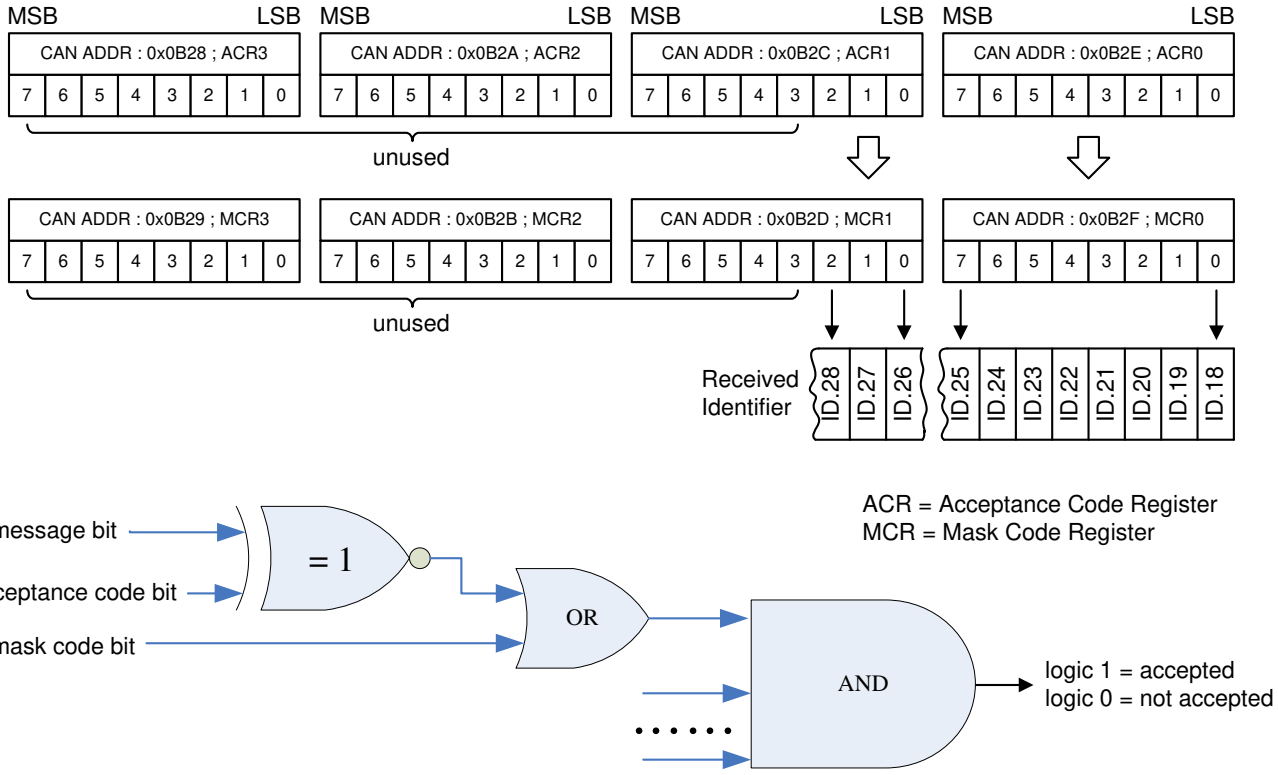
standard format frame, the lower 11 bits of the Acceptance Code and the Mask Code are valid. When received message is extended format frame, the lower 29 bits of the Acceptance Code and the Mask Code are valid.







Extended frame: if an extended format frame is received, the complete identifier including the RTR bit is used for acceptance filtering. For a successful reception of a message, all single bit comparisons have to signal acceptance. It should be noted that the 3 most significant bits of ACR3 and MCR3 are not used. In order to be compatible with future products these bit should be 'don't care' by setting MCR3.7, MCR3.6 and MCR3.5 to logic 1.



Standard frame: if a standard format frame message is received, the completely identifier are used for acceptance filtering. For a successful reception of a frame, all single bit comparisons have to signal acceptance. Note that ACR3, MCR3, ACR2, MCR2 and the 5 most significant bits of ACR1 and MCR1 are not used. In order to be compatible with future products, these unused bits in Mask Code registers should be programmed to be 'don't care' by setting these bits to logic 1.

10.9.4 Transmitting a Frame

You must write transmit buffer to send a frame to CAN-bus, then set the transmit request bit to logic 1. The controller will wait for the bus idle and sends the frame to the bus. For example, CAN0 will send a standard format frame to CAN-bus. Frame identifier is 0x001, the data have a length of 8 and the data are 11h, 22h, 33h, 44h, 55h, 66h, 77h and 88h.The program written in Keil-C is as follows:

```
Write_reg(0xAB17,0x08);  //write the frame information register, FF=0，RTR=0，DLC=8
Write_reg(0xAB18,0x00);  //write the TX ID3 Register
Write_reg(0xAB19,0x00);  // write the TX ID2 Register
Write_reg0xAB1A,0x00);  // write the TX ID1 Register
Write_reg(0xAB1B,0x01);  // write the TX ID0 Register
Write_reg(0xAB1C,0x11);  //write the TX data 1 Register
Write_reg(0xAB1C,0x22);  // write the TX data 2 Register
Write_reg0xAB1C,0x33);   // write the TX data 3 Register
Write_reg(0xAB1C,0x44);  // write the TX data 4 Register
Write_reg(0xAB1C,0x55);  // write the TX data 5 Register
Write_reg(0xAB1C,0x66);  // write the TX data 6 Register
Write_reg(0xAB1C,0x77);  // write the TX data 7 Register
Write_reg(0xAB1C,0x88);  // write the TX data 8 Register
Write_reg(0xAB01,0x01);   //write the transmit request bit
```

Note that the prototype of the function Write_reg is :

Void Write_reg(unsigned int address, unsigned char value);

10.9.5 Receiving a Frame

When CAN controller receives a frame successfully and the receive interrupt is enabled, the receive interrupt bit will be set logic 1. MCU is informed by the receive interrupt and reads the data in the receive buffer out, then releases the receive buffer. Please pay special attention here that the data must be read before releasing receive buffer, because the value in the receive buffer has changed after releasing receive buffer.

10.9.6 Dispatching

The CAN controller can transmit data periodically to CAN-bus to report its status. To realize this function, we must configure the dispatch time registers properly. Dispatch-1 has 28-bit dispatch time registers. Dispatch-2, and Dispatch-3 have 24-bit dispatch time registers. The 28 or 24 dispatch time bits control the period that the controller transmits data to CAN-bus and the dispatch period is calculate as follows: Provided the bit period is $t_b = \frac{1}{BaudRate}$ and the value of the 28 or 24 dispatch time bits is “A”, then the dispatch time is $100 \times A \times t_b$ second. In other words, the CAN controller transmits a message in the TX buffer to the CAN-bus every $100 \times A \times t_b$ second. When dispatch time bits are all set to logic 0, the dispatch function is disabled. When bus error in dispatching a frame, it can transmit an error frame and can’t transmit the failed frame again.

10.10 Self Test Mode

The self-test mode is of the CAN control is the same normal operation mode except all ACK errors are ignored. This allows the software to initiate a “self reception request” command. Upon this command, a message is transmitted to the bus and will be treated as successfully transmitted without check the ACK flag. The transmitted message is also received by the sending CAN controller if the ID field matches with the acceptance filter. The software can then check the correctness of the received message and perform loop-back test. The loop-back can be done by shorting CANTX and CANRX, or can be done through the external transceiver on the actual CAN bus. When loop back from the bus, the CAN bus needs to be properly terminated to ensure correct electrical level for normal bus operation.

11. Essential Analog Blocks

11.1 On-Chip 1.8V Regulator

The core logic and flash uses 1.8V as supply voltage. The 1.8V supply is provided by an on-chip 1.8V regulator that obtains supply from VDDH. The output of the regulator is VDD18 pin and for good transient suppression an external decoupling capacitor should be placed close to the chip and the ground plane. To achieve both reliable operations and low power consumption, the regulator consists of two parts. A main regulator that consumes about 200uA with high driving capability and accurate output level is used for normal, PMM, and STOP mode. A back-up regulator that consumes less than 20uA with limited drive (< 2.5mA) and wider variations (1.3V – 1.6V) is used in SLEEP mode. The switching between these two regulators is automatic when the operation mode switches. The back-up regulator is also used for generating Power-On-Reset conditions. When switching off the main regulator, the software must ensure that the chip 1.8V consumes less than 1mA for proper operation of the back-up regulator. It is also important that when exiting SLEP mode, the main regulator needs approximately 10 msec to become stable.

There is variation of this 1.8V supply from the main regulator due to chip to chip difference. Because this 1.8V is also used to generate the reference for IOSC and other analog peripherals such as ADC (1.8V is used as one of the ADC reference), the relative accuracy of this supply voltage is important. The on-chip regulator can be trimmed by

REGTRM (A000h) Regulator Trim Register RW 11111111 TB Protected

	7	6	5	4	3	2	1	0
RD	REGTRM[7-0]							
WR	REGTRM[7-0]							

REGTRM[7-0] Trim Register for main 1.8V regulator.
REGTRM[7-0]=FF corresponds to maximum output level. REGTRM[7-0]=00 corresponds to minimum output level. The in-between value in general is linear to the output level. Typically the maximum is around 1.95V while the minimum is around 1.65V

11.2 Precision Internal 16MHz Oscillator (IOSC)

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 also exhibits chip-to-chip variations. A calibrated value that set IOSC at 16MHz +/- 2% is stored in IFB. The user program can set this value to IOSC trimming register, IOSCITRM (A001h) and IOSCVTRM (A002h). The IOSC frequency has very little variations over the operation range (-40^{o} – 85^{o}C and VDD = 2.5V – 5.5V). The variation is typically less than +/-2% over the operation conditions. It is possible that user program to set a different frequency other than 16MHz 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. Please note that the trimming register will be set to its default value after resets, the user program must reinitialize to its calibrated value. The total trim range of the IOSC is roughly from 7MHz up to 24MHz.

The IOSC is also equipped with Spread Spectrum capability for EMI sensitive applications. The SS deviation can be controlled to fit various requirements. However, once SS is enabled, the accuracy of IOSC cannot be maintained.

IOSCITRM (A001h) IOSC Coarse Trim Register R/W 00000001 TB Protected

	7	6	5	4	3	2	1	0
RD	SSC[3-0]				SSA[1-0]		ITRM[1-0]	
WR	SSC[3-0]				SSA[1-0]		ITRM[1-0]	

SSC[3-0] SSC[3-0] defines the spread spectrum sweep rate. If SSC[3-0] = 0000, then the spread spectrum is disabled.

SSA[1-0] SSA[1-0] defines the amplitude of spread spectrum frequency change. The frequency is changed by adding SSA[1-0] range to actual IOSCVTRM[7-0].
SSA[1-0] = 11, +/- 32
SSA[1-0] = 10, +/- 16
SSA[1-0] = 01, +/- 8
SSA[1-0] = 00, +/- 4

ITRM[1-0] ITRM[1-0] is the coarse trimming of the IOSC.

IOSCVTRM (A002h) IOSC Fine Trim Register WO 10001110 TB Protected

	7	6	5	4	3	2	1	0
RD	IOSCVTRM[7-0]							
WR	IOSCVTRM[7-0]							

This register provides fine trimming of the IOSC frequency. Please note the higher the value of IOSCVTRM, the lower the frequency will be.

The manufacturer supplied trim value is stored in IFB and is trimmed to 16MHz. 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, F_IOSC= 14.0MHz - 9.5MHz - 7.0MHz (VTRM[7-0]= 00 - 80 - FF)

ITRM[1-0]=01, F_IOSC= 18.0MHz - 12.5MHz - 9.3MHz (VTRM[7-0]= 00 - 80 - FF)

ITRM[1-0]=10, F_IOSC= 22.0MHz - 15.5MHz - 11.5MHz (VTRM[7-0]= 00 - 80 - FF)

ITRM[1-0]=11, F_IOSC= 25.5MHz - 18.5MHz - 13.5MHz (VTRM[7-0]= 00 - 80 - FF)

The trimming of the IOSC should use the following procedure to obtain the default setting for 16MHz.

Set ITRM = 01, and

Set VTRM = 00, measure frequency

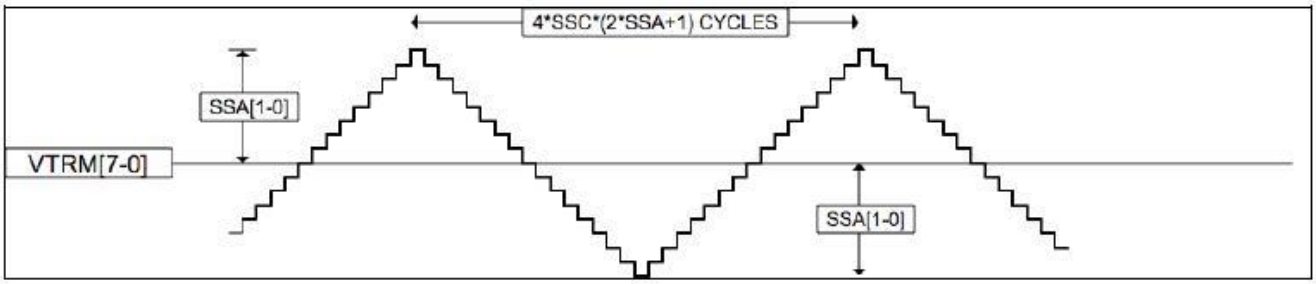
Set VTRM = FF, measure frequency

Set VTRM = 7F, measure frequency

Use binary search to obtain the closest setting for 16MHz

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

A hardware Spread Spectrum can be enabled for the IOSC. This is controlled by SSC[3-0]. When SSC[3-0] = 0, the spread spectrum is disabled, and IOSC behave normally as a fixed frequency oscillator. If SSC[3-0] is not 0, then Spread Spectrum is enabled and IOSC frequency is swept according to the setting of SSC[3-0] and SSA[1-0]. The spread is achieved by varying the actual VTRM output to the oscillator circuit thus effectively changes the oscillation frequency. The effect of SSC[3-0] and SSA[1-0] is shown in the following graph.

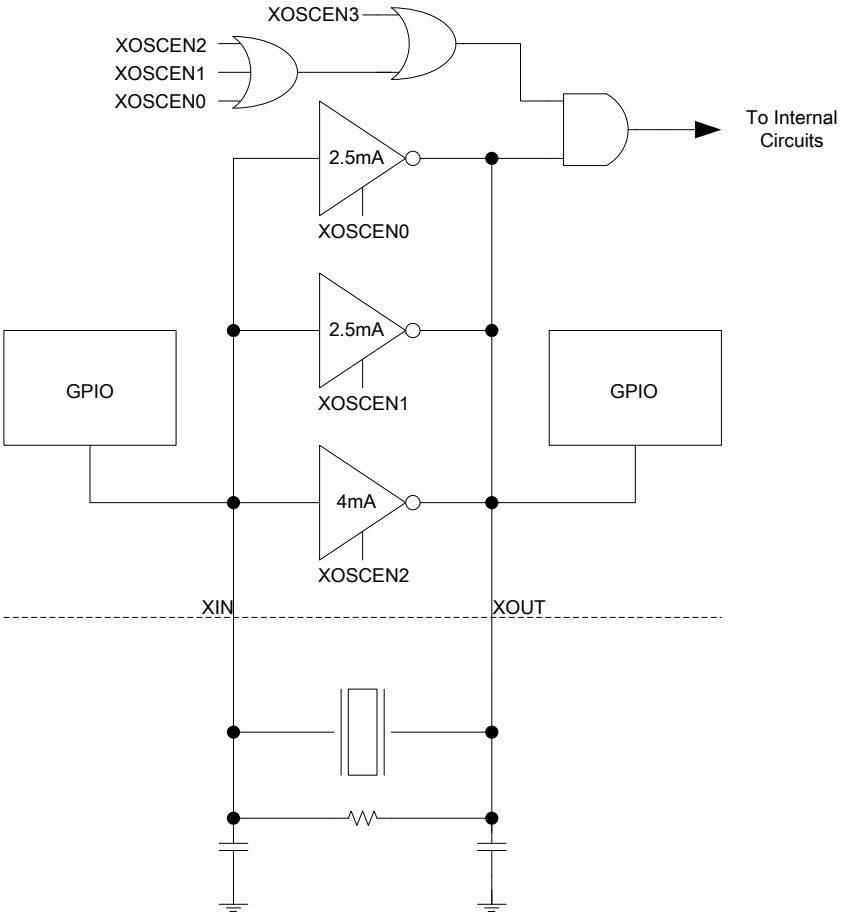


When Spread Spectrum is enabled, the actual controlling output to IOSC is VTRM[7-0] +/- SSA. This is shown in the above graph as the bold curve. The above example shows SSA[1:0] = 01, and the deviation is +/- 8. SSC[3-0] defines the update time in IOSC cycles. Then we can calculate the period of a complete sweep is 4*SSC*(2SSA+1) IOSC cycles, and we can obtain the sweep frequency from this period. When SS is enabled, the frequency of IOSC varies according to time and setting, therefore, the accuracy of IOSC frequency cannot be guaranteed. Please also note that VTRMOUT will be VTRM[7-0] +/- SSA but is bounded by 0 and 255. Therefore for a linear non-clipped sweep, VTRM[7-0] need to be within the range of SSA – 256-SSA, for example, SSA[10] = 01, then SSA is 8. VTRM[7-0] should be in the range from 8 to 248 to prevent the sweep been clipped. As Spread Spectrum suggests, the total EMI energy is not reduced, but the energy is spread over wider frequency. It is recommended that SS usage should be carefully evaluated and the setting of spread amplitude and the sweep frequency should be chosen carefully for reducing EMI effect.

11.3 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. The XOSC uses two pins to connect to an external crystal to form the oscillator. XIN is shared with GPIO P2.0 and XOUT is shared with GPIO P2.1. Sharing is in the form of double bonding. For proper operation of XOSC, IOCFG2.0 and IOCFG2.1 must be configured to high-impedance state. In addition, an external feedback resistor across the crystal is required for XOSC to oscillate.

Please note when enabling XOSC from disabled state to enabled state, it usually takes 10s of milliseconds for XOSC to stabilize. The software programmer needs to take this fact into consideration when programming switching to XOSC clock. 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 for optimization of oscillator power consumption. 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 RW 00000000 TB Protected

	7	6	5	4	3	2	1	0
RD	RTCEN	-	-	-	XOSCEN[3-0]			
WR	RTCEN	-	-	-	XOSCEN[3-0]			

This register configures the operation of the crystal oscillator and the RTC oscillator. This register is affected only by power-on reset and not by other reset conditions. In other words, the content of this register is not changed by reset conditions except the initial power on. As results, the oscillator IO configurations will not be changed by these reset conditions. RTCEN will be used to directly control the ANIO of P5.6 and P5.7.

RTCTEN RTCEN=0 disables the 32K RTC oscillator. RTCEN=1 enables the RTC oscillator.

XOSCEN[3] XOSCEN[3]=1 will allows external clocks to be applied through XOUT pin. To allow internal crystal oscillator to operate, XOSECN[3] must be 0.

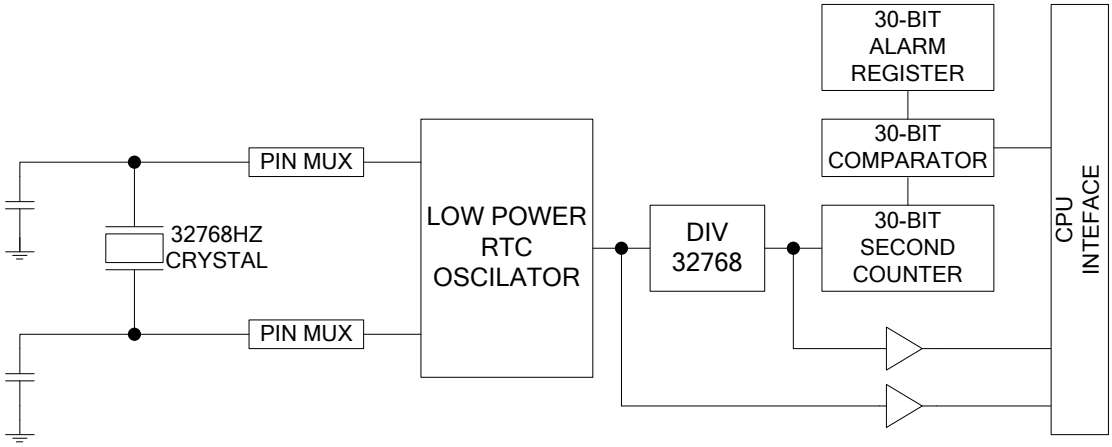
XOSCEN[2-0] XOSC oscillator power control.
The following setting is recommended but user can determine the optimal setting if the application environment is set.

XOSCEN[3-0]	OSC Operations
0000	Powered down.
0001	Low power for 3V up to 8MHz, 5V up to 12MHz.
0011	Medium power for 3V up to 16MHz, 5V up to 24MHz.
0111	High power for 3V up to 25MHz, 5V up to 30MHz.
1XXX	External clock input through P2.1

If an external clock source is applied, it should be applied at XOUT (P2.1) pin. In this case, XOSCEN3 needs to be set to 1, and XOSCEN0, XOSCEN1, and XOSCEN2 needs to be set to 0 to allow the external clock passing into the internal circuits. Under this case, P2.0 can be used for GPIO if necessary.

11.4 Real Time Clock and 32KHz Oscillator (RTC)

The on-chip RTC 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 interrupts when the counter matches the alarm. In addition, it accommodates a 32K clock (RTCCLK) as an alternative system clock source. Also available from RTC is 4HZ, and 1Hz interrupt. The block diagram of the RTC is shown in the following.



The RTC oscillator is enabled by RTCEN bit in XOSCCFG register. When enabled, RTC oscillator consumes about 2.5uA. Please note RTCEN bit is only cleared by power-on reset and not by other types of reset. The software must set RTCEN to 1 to use the RTC. The RTCXIN and RTCXOUT crystal pins are multiplexed with GPIO port P5.6 and P5.7 respectively. To use RTC, the configuration of the IO driver of these two bits must be set correctly to high impedance state. When RTCEN is set, the ANIO (ANEN is forced to 1) of P5.6 and P5.7 is connected to RXIN and RXOUT directly.

The oscillator output of 32768Hz goes into a 16-bit dividing counter (RTCCNT[15-0]). This counter provides 4Hz and 1Hz clocks for interrupt purpose. The 1Hz clock also goes into a 32-bit SECOND counter (RTCSCND[31-0]). Both RTCCND and RTCSCND are read only and can be cleared to 0 by issuing a clear command through RTCCMD register. There is also a 32-bit alarm register (RTCALRM[31-0]). The alarm register provides a compare value with RTCSCND. When a match condition occurs, an alarm interrupt is triggered. The alarm register, RTCALRM, is accessed at the same address location as RTCSCND.

RTCCMD (0xA00E) RTC Configuration and Command Register RW 0000000

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

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 Second Interrupt bit. This bit must be cleared by software.
ALMINT Alarm Interrupt bit. This bit must be cleared by software.

4HZINTEN	4Hz Interrupt Enable bit, Set to enable 4Hz interrupt.
4HZINT	4Hz Interrupt bit. Set by hardware every 250 millisecond if enabled. This bit must be cleared by software.
CLRSCND	Clear (RTC) Second Counter bit Set this bit to 1 to force the SECOND counter to 0. CLRSCND must be cleared to 0 to allow the SECOND counter to continue to count.
CLRCNT	Clear (RTC) 32K Counter bit Set this bit to 1 to force the 32768 divider counter to 0. CLRCNT must be cleared to 0 to allow RTC counter to continue to count.

RTCCNT0 (0xA00C) RTC Counter Low Byte Register 0 RO 00000000

	7	6	5	4	3	2	1	0
RD	RTCCNT[15-8]							
WR	-							

This register holds the value for the lower 8 bits of the RTC 32KHz divider counter. The counter value can be cleared by issuing CLRCNT.

RTCCNT1 (0xA00D) RTC Counter High Byte Register 1 RO 00000000

	7	6	5	4	3	2	1	0
RD	RTCCNT[15-8]							
WR	-							

This register holds the value for the upper 8 bits of the RTC 32KHz divider counter. The counter value can be cleared by issuing CLRCNT.

RTCSCND0 (0xA008) RTC SECOND Counter Register 0 RW 00000000

	7	6	5	4	3	2	1	0
RD	RTCSCND[7-0]							
WR	RTCALRM[7-0]							

This register holds the value for the SECOND counter bit 7 to 0 locations. This is read only and can be cleared by CLRSCND. When written, it writes into the ALARM register.

RTCSCND1 (0xA009) RTC SECOND Counter Register 1 RW 00000000

	7	6	5	4	3	2	1	0
RD	RTCSCND[15-8]							
WR	RTCALRM[15-8]							

This register holds the value for the SECOND counter bit 15 to 8 locations. This is read only and can be cleared by CLRSCND. When written, it writes into the ALARM register.

RTCSCND2 (0xA00A) RTC SECOND Counter Register 2 RW 00000000

	7	6	5	4	3	2	1	0
RD	RTCSCND[23-16]							
WR	RTCALRM[23-16]							

This register holds the value for the SECOND counter bit 23 to 16 locations. This is read-only and can be cleared by CLRSCND. When written, it writes into the ALARM register.

RTCSCND3 (0xA00B) RTC SECOND Counter Register 3 RW 00000000

	7	6	5	4	3	2	1	0
RD	RTCSCND[31-24]							
WR	RTCALRM[31-24]							

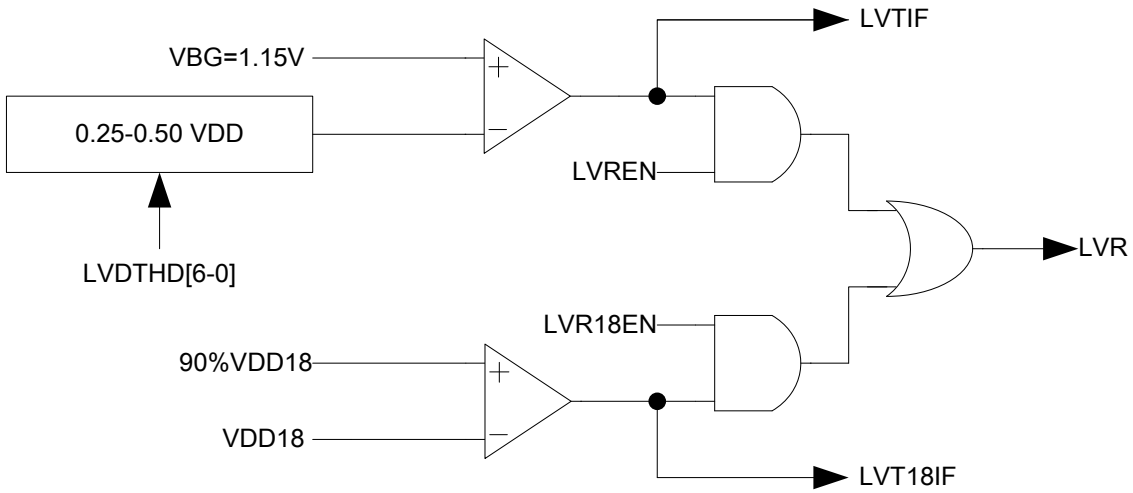
This register holds the value for the SECOND counter bit 31 to 24 locations. This is read only and can be cleared by CLRSCND. When written, it writes into the ALARM register.

11.5 Slow Internal Oscillator (SIOSC)

The SIOSC is a 100 KHz low power internal R/C oscillator. The oscillator consumes about 5uA and is always enabled. SIOSC can be used as system clock or as T5 clocking source to provide extended long period timing for wake up purpose. SIOSC connects to the VDD power supply, and its frequency will vary as VDD supply varies. The accuracy of SIOSC is not guaranteed and typically lies within 50KHz to 150KHz, and variations to VDD and temperature is about +/- 50%.

11.6 Supply Low Voltage Detection (LVD)

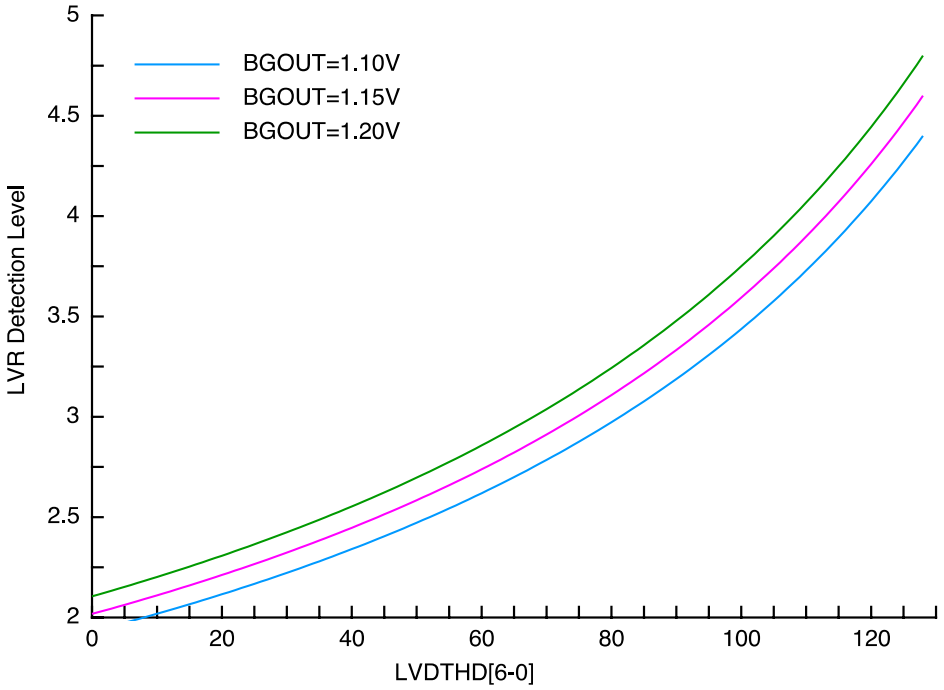
There are two Supply Low Detection circuits are combined together for reliable MCU operations. The first is low supply detection for VDD18. The detection threshold is fixed at 90% of its stable value. This guarantees internal logic can be shut down at falling VDD18 supply. The second is low supply detection circuit on VDD (LVD). When enabled, it detect VDD < VTH condition. This provides an earlier detection point on unstable VDD supply. Both detection results can be configured to generate an interrupt or a system reset. When used as system reset, it also forces the RSTN pin to low that extends the reset period. The block diagram of Low Voltage Detection Circuit is shown in the following figure.



An enabled LVD circuit consumes about 100uA to 400uA depending on its VDD level. The LVDTHD[6-0] sets the compare threshold from 0.25VDD to 0.50VDD against the internal band-gap reference voltage of 1.15V. The detection threshold can be represented in the following equation when LVDTHV is the detection voltage.

LVDTHV = BGOUT / [0.57 – 0.0025 * LVDTHD[6-0]].

Band-gap voltage suffers some chip-to-chip variations. The following graph shows the supply detection threshold of BGOUT of 1.10V, 1.15V and 1.20V. The vertical axis is the detection voltage and the horizontal axis is the LVDTHD[6-0] value in decimal. For rough detection, the user program can use BGOUT=1.15V as the nominal value and obtain the corresponding LVDTHD value from this graph for specific detection level. This may result approximately +/- 5% to +7% variation.



For very precise detection level, the manufacturing process stores the LVDTHD value for detection of 4.0V and 3.0V in IFB. The user program can then use these two values and obtain a real average BGOUT value using the above formula. Then use the real BGOUT value to obtain the specific LVDTHD value for an arbitrary detection level.

The user program enables the preferred configuration and sets 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. Because in the SLEEP mode, the main regulator is turned off and VDD18 is supplied by backup regulator with lower than 1.8V level. LVR18 or LVT18 should be turned off too. This prevents false triggering of the LVR18 or LVT18. These should be turned off before entering SLEEP mode, and enabled after VDD18 is stable after exiting of the SLEEP mode. Typical stable time of VDD18 after the exiting of SLEEP mode is 20usec.

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

	7	6	5	4	3	2	1	0
RD	LVDEN	LVREN	LVTEN	LVR18EN	LVT18EN	-	LVT18IF	LVTIF
WR	LVDEN	LVREN	LVTEN	LVR18EN	LVT18EN	-	LVT18IF	LVTIF

LVDEN	LVD Enable bit. Set to turn on supply voltage detection circuits.
LVREN	LVR Enable bit. LVREN = 1 allows low voltage detect condition to cause a system reset.
LVTEN	LVT Enable bit. LVTEN = 1 allows low voltage detect condition to generate an interrupt.
LVR18EN	LVR18 Enable bit. LVR18EN = 1 allows low voltage detect condition to cause a system reset.
LVT18EN	LVT18 Enable bit. LVT18EN = 1 allows low voltage detect condition to generate an interrupt.
LVT18IF	Core VDD18 Low Voltage Detect Interrupt Flag LVT18IF is set by hardware when LVD detection occurs and must be cleared by software.
LVTIF	Low Voltage Detect Interrupt Flag LVTIF is set by hardware when LVD detection occurs and must be cleared by software.

LVDTHD (A011h) Supply Low Voltage Detection Threshold Register WO X1111111 TB Protected

	7	6	5	4	3	2	1	0
RD	-	LVDTHD6	LVDTHD5	LVDTHD4	LVDTHD3	LVDTHD2	LVDTHD1	LVDTHD0
WR	-	LVDTHD6	LVDTHD5	LVDTHD4	LVDTHD3	LVDTHD2	LVDTHD1	LVDTHD0

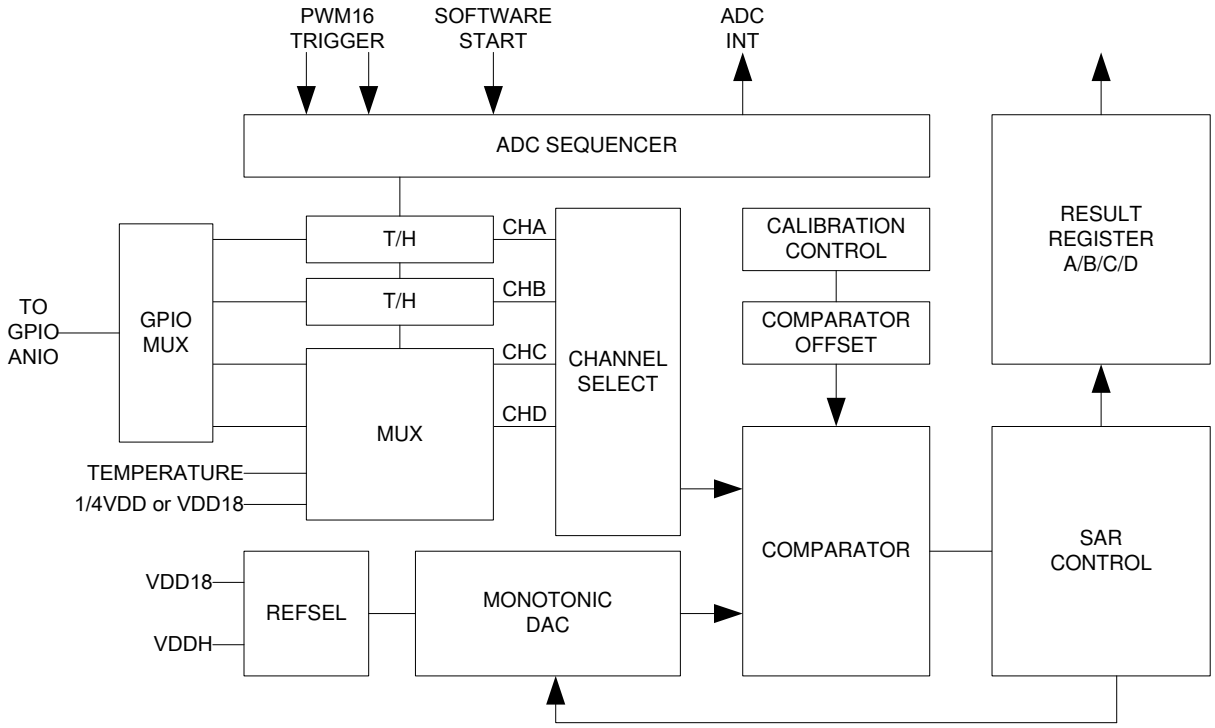
LVDTHD = 00 will set the detection threshold at its minimum (approximately 2.2V), and LVDTHD = 7F will set the detection threshold at its maximum (approximately 4.5V).

12. 12-Bit SAR ADC (ADC)

The on-chip ADC is a 12-bit SAR based ADC with maximum ADC clock rate of 8MHz. The ADC has inherent monotonic characteristics with built-in offset cancellation and each conversion takes 32 cycles to complete. The ADC clock is programmable and set by the ADC clock scaler. The ADC uses VDDH as reference full range. When enabled, the ADC consumes about 3mA of current.

The ADC has 4 intrinsic channels each can be separately enabled and each has independent result registers. Two of the intrinsic channels have T/H stages before the ADC. The ADC under triggering will first put the T/H in hold phase then converts each enabled channel sequentially. At completion of the conversions, an ADC interrupt is generated.

In CS8958A, each inherent channel is further multiplexed to various external pins which are shared with GPIO port and the connection is through the IOCONFIG ANEN control. The block diagram of ADC is shown in the following



ADCCFG (0xA9h) ADC Configuration Register RW 0000000

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

ADCEN — ADC Enable bit
ADCEN=1 enables ADC.
ADCEN=0 puts ADC into power down mode.
When ADCEN is set from 0 to 1, the program needs to wait at least 20us to allow analog bias to stabilize to ensure ADC's proper functionality.
Before enter STOP/SLEEP mode, the ADCEN must be cleared to reduce the power consumption.

ADCINTE — ADC Interrupt Enable bit
ADCINTE=1 enables the ADC interrupt when conversion completes.
ADCINTE=0 disables the ADC interrupt

CSTART — Software Start Conversion bit
Set this CSTART=1 to trigger an ADC conversion on selected channels. This bit is self-cleared when the conversion is done. If a hardware triggered conversion is on-going, this will be ignored.

ADCFM ADC Result Format Control bit

ADCFM = 1 sets ADC result as MSB justified. ADCAH contains the MSB bits of the result. ADCAL[7-4] contains LSB results and ADCAL[3-0] is filled with 0000.

ADCFM = 0 sets ADC result as LSB justified. ADCAH[7-4] is filled with 0000. ADCAH[3-0] contains MSB result. ADCAL contains the LSB results.

PRE1, PRE0 ADC Clock Divider

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

ADCAVG (0xCEh) ADC Control Register RW 00XXX000

	7	6	5	4	3	2	1	0
RD	CHDSL[1]	CHDSL[0]	CHATHEN	CHBTHEN	REFSEL	AVG2	AVG1	AVG0
WR	CHDSL[1]	CHDSL[0]	CHATHEN	CHBTHEN	REFSEL	AVG2	AVG1	AVG0

CHDSL[1-0] CH D Auxiliary Channel Enable and Select

Channel D can be used to connect to internal temperature sensor or VDD18. Since VDD18 is calibrated, this allows precision measurement using VDD as reference through mathematics conversion. If internal channel is selected, the external connection is automatically disconnected.

CHDSL[1-0] = 00, default for external connection.

CHDSL[1-0] = 01, external connection

CHDSL[1-0] = 10, connects to internal temperature sensor

CHDSL[1-0] = 11, connects to VDD18.

CHATHEN Channel A Track/Hold Enable

CHATHEN=0 disable the function of T/H

CHATHEN=1 enables the function of T/H. In this mode, the T/H circuit follows the input signal and is put into holding mode when the conversion is started.

CHBTHEN Channel B Track/Hold Enable

CHATHEN=0 disable the function of T/H

CHATHEN=1 enables the function of T/H. In this mode, the T/H circuit follows the input signal and is put into holding mode when the conversion is started.

REFSEL Full Scale Reference Selection

REFSEL = 0 use VDD as full scale reference

REFSEL = 1 use VDD18 as full scale reference

Before enter STOP/SLEEP mode, the REFSEL must be set to “1” to reduce the power consumption.

AVG[2-0] AVG[2-0] controls the hardware averaging logic of ADC readout. It is recommended the setting is changed only when ADC is stopped. The default is 00.

AVG2	AVG1	AVG0	ADC Result
0	0	0	1 Times Average
0	0	1	2 Times Average
0	1	0	4 Times Average
0	1	1	8 Times Average
1	0	0	16 Times Average
1	0	1	32 Times Average
1	1	0	64 Times Average
1	1	1	TEST MODE

ADCCHSL (0xB9h) 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 ADCCHA=1 enables ADC Channel A for conversion cycle
ADCCHB ADCCHB=1 enables ADC Channel B for conversion cycle
ADCCHC ADCCHC=1 enables ADC Channel C for conversion cycle
ADCCHD ADCCHD=1 enables ADC Channel D for conversion cycle
CHAIF Channel A Conversion Completion Interrupt Flag bit
CHAIF is set by hardware when the conversion is completed and new result is written to ADCAL and ADCAH result registers. If ADC interrupt is enabled, this also generates an interrupt. This bit is cleared when ADCAL is read. When this flag is set, no new conversion result is updated.
CHBIF Channel B Conversion Completion Interrupt Flag bit
CHBIF is set by hardware when the conversion is completed and new result is written to ADCBL and ADCBH result registers. If ADC interrupt is enabled, this also generates an interrupt. This bit is cleared when ADCBL is read. When this flag is set, no new conversion result is updated.
CHCIF Channel C Conversion Completion Interrupt Flag bit
CHCIF is set by hardware when the conversion is completed and new result is written to ADCCL and ADCCH result registers. If ADC interrupt is enabled, this also generates an interrupt. This bit is cleared when ADCCL is read. When this flag is set, no new conversion result is updated.
CHDIF Channel D Conversion Completion Interrupt Flag bit
CHDIF is set by hardware when the conversion is completed and new result is written to ADCDL and ADCDH result registers. If ADC interrupt is enabled, this also generates an interrupt. This bit is cleared when ADCDL is read. When this flag is set, no new conversion result is updated.

ADCAH to ADCDH and ADCAL to ADCDL are the low and high byte result registers respectively, and are read-only. Reading low byte result also clears its corresponding interrupt flag. If the flag is not cleared, no new result is updated. The software should always read the low byte last. The format of the high byte and low byte depends on ADCFM setting.

ADCAL (0xBAh) Channel A Result Register Low Byte RO XXXXXXXX

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

ADCAH (0xBBh) Channel A Result Register High Byte RO XXXXXXXX

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

ADCBL (0xBCh) Channel B Result Register Low Byte RO XXXXXXXX

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

ADCBH (0xBDh) Channel B Result Register High Byte RO XXXXXXXX

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

ADCCL (0xBEh) Channel C Result Register Low Byte RO XXXXXXXX

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

ADCCH (0xBFh) Channel C Result Register High Byte RO XXXXXXXX

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

ADCDL (0xAAh) Channel D Result Register Low Byte RO XXXXXXXX

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

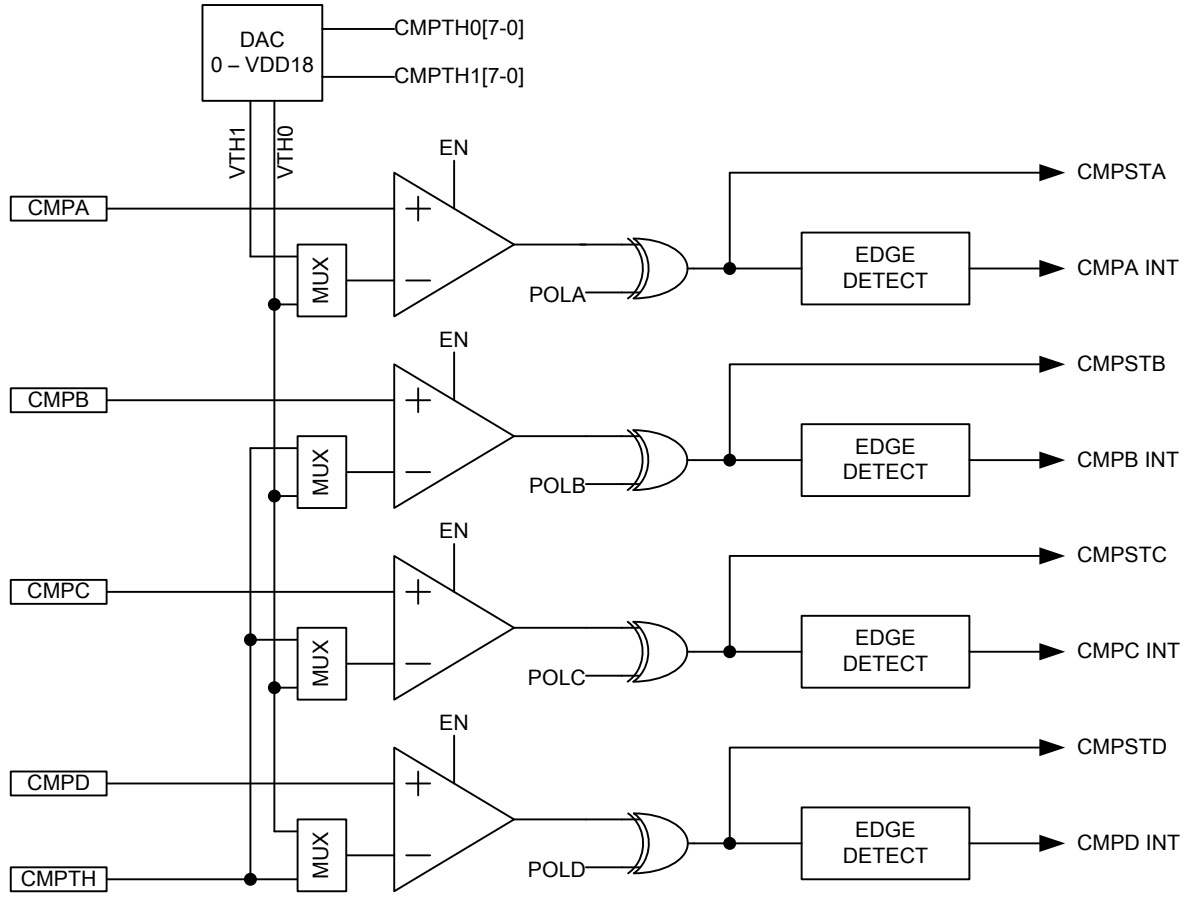
ADCDH (0xABh) Channel D Result Register High Byte RO XXXXXXXX

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

13. Analog Comparators (ACMP)

IS31CS8968A has four analog comparators as its on-chip external peripherals. When enabled, each comparator consumes about 250uA. The input signal range is from 0 to VDD. There are two 8-bit R-2R DAC associated with the comparators to generate the compare threshold. The R-2R DAC uses the internal 1.8V supply as the full scale range thus limiting the comparator threshold from 0V to 1.8V in 256 steps. Channel B/C/D can select a common external threshold. The inputs of the comparators are multiplexed with multi-function GPIO pins, P2.7/P3.0/P3.1, P2.6, P2.5, P2.4, and the external threshold is through P4.6. To use these ports as comparator inputs, the ANEN must be enabled and other drivers to be in high-impedance state.

The real-time outputs of the comparator can be read by the CPU directly through register access. The output is also sent to an edge-detector and any edge transition can be used to trigger an interrupt. The stabilization time from off state to enabled state of the comparator block is about 20usec. The block diagram of the analog comparator is shown in the following diagram.



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

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

CMPENA	Comparator A Enable bit. Set to enable the comparator. When CMPENA is set from 0 to 1, the program needs to wait at least 20us allowing analog bias to stabilize to ensure comparator A's proper functionality.
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.
POLA	Channel A Output polarity control bit POLA=0 set default polarity POLA=1 reverse the output polarity of the comparator
CMPENB	Comparator B Enable bit. Set to enable the comparator. When CMPENB is set from 0 to 1, the program needs to wait at least 20us allowing analog bias to stabilize to ensure comparator B's proper functionality.
THSELB	Comparator A Threshold Select Bit. THSELB = 0, the comparator B uses VTH0 as the threshold. THSELB = 1, the comparator B uses external threshold.
INTENB	Set to enable the comparator B interrupt.
POLB	Channel B Output polarity control bit POLB=0 set default polarity POLB=1 reverse the output polarity of the comparator

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

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

CMPENC	Comparator C Enable Bit. Set to enable the comparator. When CMPENC is set from 0 to 1, the program needs to wait at least 20us to allow analog bias to stabilize to ensure comparator C's proper functionality.
THSELC	Comparator C Threshold Select Bit. THSELC = 0, the comparator C uses VTH0 as the threshold. THSELC = 1, the comparator C uses external threshold.
INTENC	Set to enable the comparator C interrupt.
POLC	Channel C Output polarity control bit POLC=0 set default polarity POLC=1 reverse the output polarity of the comparator
CMPEND	Comparator D Enable Bit. Set to enable the comparator. When CMPEND is set from 0 to 1, the program need to wait at least 20us to allow analog bias to stabilize to ensure comparator D's proper functionality.
THSELD	Comparator A Threshold Select Bit. THSELD = 0, the comparator D uses VTH0 as the threshold. THSELD = 1, the comparator D uses external threshold.
INTEND	Set to enable the comparator D interrupt.
POLD	Channel D Output polarity control bit POLD=0 set default polarity POLD=1 reverse the output polarity of the comparator

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 1.8V. When not used, it should be set to 0x00 to save power consumption.

CMPVTH1 (0xA033h) Analog Comparator Threshold Control Register RW 0000000

	7	6	5	4	3	2	1	0
RD	VTH1 Register							
WR	VTH1 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 1.8V. 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	CMPHYSB[1-0]		CMPHYSA[1-0]	

CMPIFD Comparator D Interrupt Flag bit. This bit is set when CMPSTD is toggled and the comparator D setting is enabled. This bit must be cleared by software.

CMPIFC Comparator C Interrupt Flag bit. This bit is set when CMPSTC is toggled and the comparator C setting is enabled. This bit must be cleared by software.

CMPIFB Comparator B Interrupt Flag bit. This bit is set when CMPSTB is toggled and the comparator B setting is enabled. This bit must be cleared by software.

CMPIFA Comparator A Interrupt Flag bit. This bit is set when CMPSTA is toggled and the comparator A setting is enabled. This bit must be cleared by software.

CMPSTD Comparator D Real-time Output. If the comparator is disabled, this bit is forced low.

CMPSTC Comparator C Real-time Output. If the comparator is disabled, this bit is forced low.

CMPSTB Comparator B Real-time Output. If the comparator is disabled, this bit is forced low.

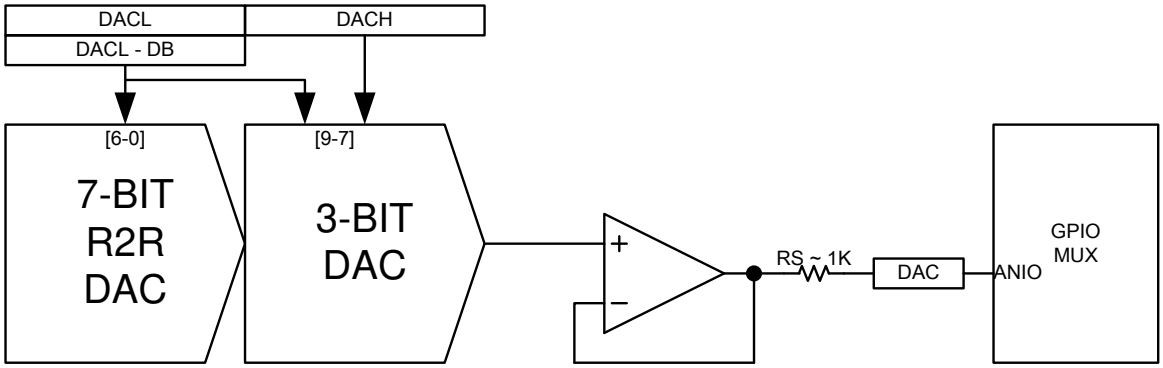
CMPSTA Comparator A Real-time Output. If the comparator is disabled, this bit is forced low.

CMPHYSB[1-0] Comparator B/C/D Hysteresis Enable bit.
CMPHYSB[1-0] = 00 disables comparator hysteresis
CMPHYSB[1-0] = 01 comparator hysteresis = 10mV
CMPHYSB[1-0] = 10 comparator hysteresis = 20mV
CMPHYSB[1-0] = 10 comparator hysteresis = 30mV

CMPHYSA[1-0] Comparator A Hysteresis Enable bit.
CMPHYSA[1-0] = 00 disables comparator hysteresis
CMPHYSA[1-0] = 01 comparator hysteresis = 10mV
CMPHYSA[1-0] = 10 comparator hysteresis = 20mV
CMPHYSA[1-0] = 10 comparator hysteresis = 30mV

14. 10-Bit Voltage Output DAC (VDAC)

A 10-bit voltage output DAC is included. The DAC is composed of LSB 7-bit R2R and MSB 3-bit linear DAC. The output is buffered by unity configured OPAMP. The output range of the DAC is from 0V to VDD. Due to the circuit structure of the OPAMP, the output accuracy will suffer some loss from ¾ VDD to VDD. The output impedance of the buffer is less than 1K Ohm and should not drive high capacitance load. Please also note that the linearity and accuracy of the DAC will suffer when the output is close to rail or 0V because of the OPAMP. The update of DAC must start with low-byte first and then high-byte because the low-byte is double-buffered.



DACH (0xA037h) DAC High Register R/W 0x00

	7	6	5	4	3	2	1	0
RD	DACEN	-	-	-	-	-	DAC[9-8]	
WR	DACEN	-	-	-	-	-	DAC[9-8]	

DACEN DAC Enable Control bit
DACEN=1 enables DAC
DACEN=0 disable DAC

DAC[9-8] DAC[9-8] Data bits
These two bits are MSB of 10-bit DAC data. Writing to this register updates the DAC output.

DACL (0xA036h) DAC Low Register R/W 0x00

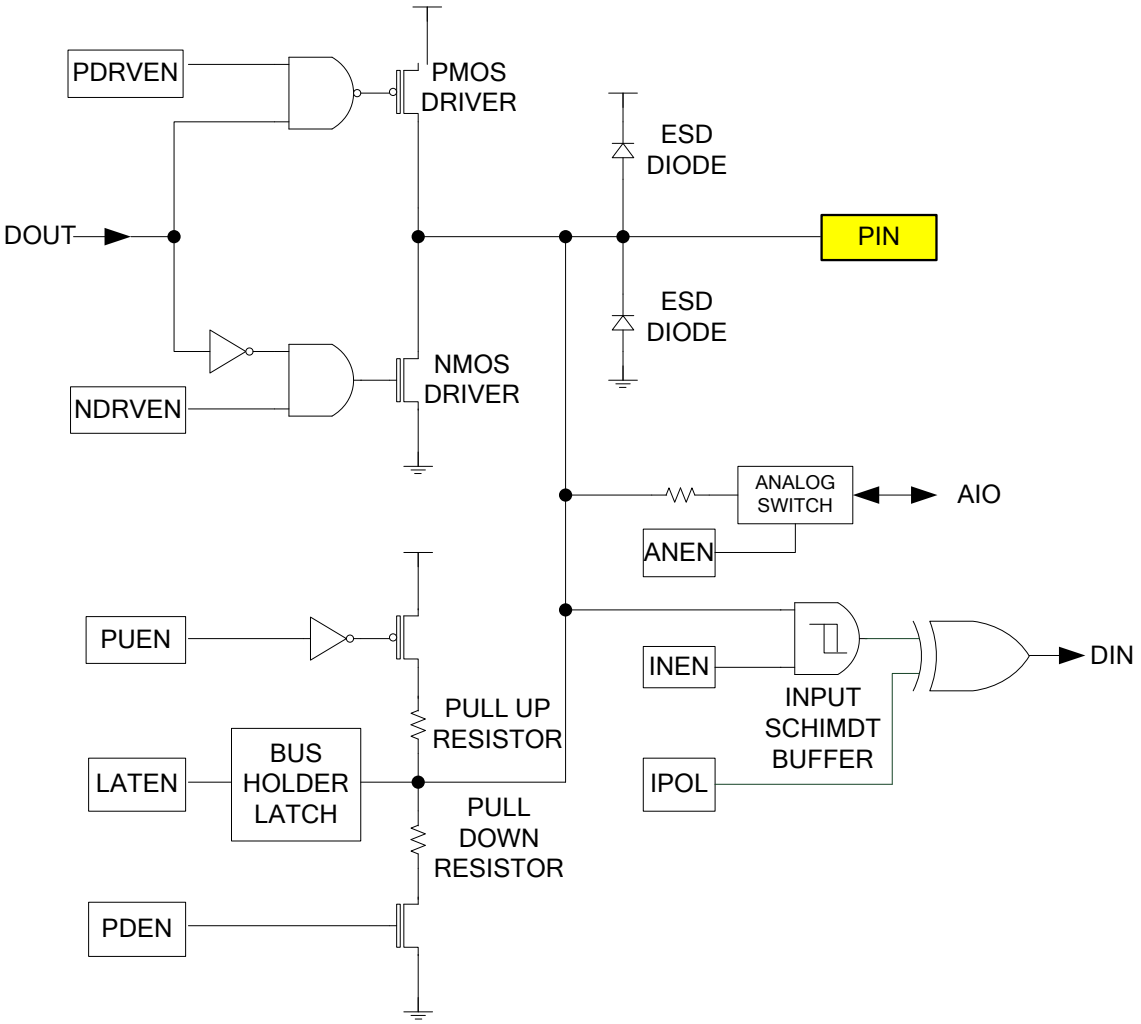
	7	6	5	4	3	2	1	0
RD	DAC[7-0]							
WR	DAC[7-0]							

This register is double-buffered and the output is not updated until DACH is written.

15. GPIO Port Function and Pin Configurations

This section describes the pin functions and configurations. Almost all signal pins are multi-functional with default setting 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 registers and the register names and pin names are referenced by their default GPIO port name. For example, in IS31CS8968A, pin 35 of LQFP-48 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 are referred as P0.0 name. Because every signal pin is a general purpose I/O as well as multi-functional pin, IS31CS8968A 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 provides analog I/O capability that can be selected when the GPIO pin is shared with analog peripheral purposes such as analog OPAMP, ADC input or DAC output.

The supply voltage of the I/O buffer uses VDD (2.5V 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.



Please note for those pins denoted as OD (Open Drain), P1.3 (SCL), P1.2 (SDA), P4.0 (CEC), P4.1 (SDA2), P4.3 (SCL2), the upper side ESD diode, PMOS Driver, and Analog Switch are removed. This allows the external pull-up of the OD pins to be connected up to 12V.

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, 0xA068 – 0xA06F for P3.0 to P3.7, 0xA0C0 – 0xA0C8 for P4.0 to P4.7, and 0xA0AE – 0xA0AF for P5.6 to P5.7. The 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	IPOL

INEN Input buffer control. Set this bit to enable the GPIO 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.

IPOL DIN polarity switch. Set this bit will invert the DIN input value from the PIN

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

Please note the following exceptions exist for IOCFG registers.

IOCFGP2.0 and IOCFGP2.1 are used for XIN and XOUT. When XOSC is enabled, P2.0 and P2.1 are forced to high impedance.

IOCFGP5.6 and IOCFGP5.7 are used for RXIN and RXOUT. When RTC is enabled, P5.6 and P5.7 have ANEN forced to 1.

Because each signal pin is a multi-functional and 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 pins. These selection and definitions are pin specific and product specific. The following description describes the selection and control for IS31CS8968A signal pins.

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

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

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

PINTEDGE Pin Interrupt Edge bit. 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 ADA1 Enable bit. 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. DISABLE is the default value.

PINTEN Pin Interrupt Enable Control Bit. Set this bit to enable the P0.0 pin as the pin interrupt PINT1.0. DISABLE is the default value.

GPIOEN P0.0 GPIO Function Enable Bit. Set this bit to enable P0.0's GPIO function. DISABLE is the default value.

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

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

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

PINTEDGE Pin Interrupt Edge bit. 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.

ADA2EN ADA2 Enable bit. Set this bit to enable the P0.1 pin as the ADC channel A2 input. Please note to enable this function, the IOCFGP0.1's ANEN bit must also be set. DISABLE is the default value.

PINTEN Pin Interrupt Enable Bit. Set this bit to enable the P0.1 pin as the pin interrupt PINT1.1. DISABLE is the default value.

GPIOEN P0.1 GPIO Function Enable Bit. Set this bit to enable P0.1's GPIO function. DISABLE is the default value.

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

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

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

PINTEDGE Pin Interrupt Edge bit. 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 ADB1 Enable bit. 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. DISABLE is the default value.

PINTEN Pin Interrupt Enable Bit. Set this bit to enable the P0.2 pin as the pin interrupt PINT1.2. DISABLE is the default value.

GPIOEN P0.2 GPIO Function Enable Bit. Set this bit to enable P0.2's GPIO function. DISABLE is the default value.

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

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

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

PINTEDGE Pin Interrupt Edge bit. 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 ADB2 Enable bit. 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. DISABLE is the default value.

PINTEN Pin Interrupt Enable bit. Set this bit to enable the P0.3 pin as the pin interrupt PINT1.3. DISABLE is the default value.

GPIOEN P0.3 GPIO Function Enable Bit. Set this bit to enable P0.3's GPIO function. DISABLE is the default value.

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

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

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

PINTEDGE Pin Interrupt Edge bit. 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 when PINTEN is set.

ADC1EN ADC1 Enable bit. 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. DISABLE is the default value.

PINTEN Pin Interrupt Enable Control Bit. Set this bit to enable the P0.4 pin as the pin interrupt PINT0.4. DISABLE is the default value.

GPIOEN P0.4 GPIO Function Enable Bit. Set this bit to enable P0.4's GPIO function. DISABLE is the default value.

MFCFGP0.5 (0xA055) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	PINTEDG	ADC2EN	PINTEN	GPIOEN
WR	-	-	-	-	PINTEDG	ADC2EN	PINTEN	GPIOEN

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

PINTEDGE Pin Interrupt Edge bit. 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 when PINTEN is set.

ADC2EN ADC2 Enable bit. 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. DISABLE is the default value.

PINTEN Pin Interrupt Enable Control Bit. Set this bit to enable the P0.5 pin as the pin interrupt. PINT0.5. DISABLE is the default value.

GPIOEN P0.5 GPIO Function Enable Bit. Set this bit to enable P0.5's GPIO function. DISABLE is the default value.

MFCFGP0.6 (0xA056) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	ADD1EN	TXD0EN	GPIOEN
WR	-	-	-	-	-	ADD1EN	TXD0EN	GPIOEN

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

ADD1EN ADD1 Enable bit. Set this bit to enable the P0.6 pin as the ADC channel D1 input. Please note to enable this function, the IOCFGP0.6's ANEN bit must also be set. DISABLE is the default value.

TXD0EN TXD Enable bit. Set this bit to enable P0.6 pin as TXD signal for UART0. DISABLE is the default value.

GPIOEN P0.6 GPIO Function Enable Bit. Set this bit to enable P0.6's GPIO function. DISABLE is the default value.

MFCFGP0.7 (0xA057) R/W (0x00)

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

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

ADD2EN ADD2 Enable bit. 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. DISABLE is the default value.

RXD0EN RXD Enable bit. Set this bit to enable P0.7 pin as RXD signal for UART0. DISABLE is the default value.

GPIOEN P0.7 GPIO function Enable Bit. Set this bit to enable P0.7's GPIO function. DISABLE is the default value.

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

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

P1.0 is shared with PCA CEX0, and ADC Channel D3 input. Only one bit can be set at one time..

CEX0EN CEX0 Enable bit. Set this bit to enable the P1.0 pin as CEX0 for PCA peripheral. DISABLE is the default value.

ADD3EN ADD3 Enable bit. 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. DISABLE is the default value.

GPIOEN P1.0 GPIO Function Enable Bit. Set this bit to enable P1.0's GPIO function. DISABLE is the default value.

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

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

P1.1 is shared with PCA CEX1, and ADC Channel D4 input. Only one bit can be set at one time..

CEX1EN CEX1 Enable bit. Set this bit to enable the P1.1 pin as CEX1 for PCA peripheral. DISABLE is the default value.

ADD4EN ADD4 Enable bit. 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. DISABLE is the default value.

GPIOEN P1.1 GPIO Function Enable Bit. Set this bit to enable P1.1's GPIO function. DISABLE is the default value.

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

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

P1.2 is shared with I^2C Master and I^2C Slave's SDA. Only one bit can be set at one time.

MSDAEN MSDA Enable bit. Set this bit to enable the P1.2 pin as the Master/Slave I^2C master SDA signal. The corresponding IOCFGP1.2 should also be set correctly as SDA signal should be an open-drain configuration. DISABLE is the default value.

SSDAEN SSDA Enable bit. Set this bit to enable the P1.2 pin as the Master/Slave I^2C slave SDA signal. The corresponding IOCFGP1.2 should also be set correctly as SDA signal should be an open-drain configuration. DISABLE is the default value.

GPIOEN P1.2 GPIO Function Enable Bit. Set this bit to enable P1.2's GPIO function. DISABLE is the default value.

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

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

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

MSCLEN MSCL Enable bit. Set this bit to enable the P1.3 pin as the Master/Slave I²C master SCL signal. The corresponding IOCFGP1.3 should also be set correctly as SCL signal should be an open-drain configuration. DISABLE is the default value.

SSCLEN SSCL Enable bit. Set this bit to enable the P1.3 pin as the Master/Slave I²C slave SCL signal. The corresponding IOCFGP1.3 should also be set correctly as SCL signal should be an open-drain configuration. DISABLE is the default value.

GPIOEN P1.3 GPIO Function Enable Bit. Set this bit to enable P1.3's GPIO function. DISABLE is the default value.

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

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

GPIOEN P1.4 GPIO Function Enable Bit. Set this bit to enable P1.4's GPIO function. DISABLE is the default value.

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

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

GPIOEN P1.5 GPIO Function Enable Bit. Set this bit to enable P1.5's GPIO function. DISABLE is the default value.

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

	7	6	5	4	3	2	1	0
RD	-	-	-	-	T1EN	PINTEDG	PINTEN	GPIOEN
WR	-	-	-	-	T1EN	PINTEDG	PINTEN	GPIOEN

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

T1EN T1 Enable bit. Set this bit to enable the P1.6 pin as T1. DISABLE is the default value.

PINTEDGE Pin Interrupt Edge bit. 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. DISABLE is the default value.

GPIOEN P1.6 GPIO Function Enable bit. Set this bit to enable P1.6's GPIO function. DISABLE is the default value.

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

	7	6	5	4	3	2	1	0
RD	-	-	-	-	T0EN	PINTEDG	PINTEN	GPIOEN
WR	-	-	-	-	T0EN	PINTEDG	PINTEN	GPIOEN

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

T0EN T0 Enable bit. Set this bit to enable the P1.7 pin as T0. DISABLE is the default value.

PINTEDGE Pin Interrupt Edge bit. 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. DISABLE is the default value.

GPIOEN P1.7 GPIO Function Enable bit. Set this bit to enable P1.6's GPIO function. DISABLE is the default value.

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

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

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

XINEN XIN Enable bit. Set this bit to enable the P2.0 pin as crystal oscillator XIN. DISABLE is the default value.

GPIOEN P2.0 GPIO Function Enable bit. Set this bit to enable P2.0's GPIO function. DISABLE is the default value.

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

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

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

XOUTEN XOUT Enable bit. Set this bit to enable the P2.1 pin as crystal oscillator XOUT. DISABLE is the default value.

GPIOEN P2.1 GPIO Function Enable bit. Set this bit to enable P2.1's GPIO function. DISABLE is the default value.

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

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	TXD2EN	GPIOEN
WR	-	-	-	-	-	-	TXD2EN	GPIOEN

P2.2 is shared with UART2 TXD2. Only one bit can be set at one time.

TXD2EN TXD2 Enable. Set this bit to enable the P2.2 pin as TXD2. DISABLE is the default value.

GPIOEN P2.2 GPIO Function Enable bit. Set this bit to enable P2.2's GPIO function. DISABLE is the default value.

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

	7	6	5	4	3	2	1	0
RD	VDACEN	-	-	-	-	-	RXD2EN	GPIOEN
WR	VDACEN	-	-	-	-	-	RXD2EN	GPIOEN

P2.3 is shared with UART2 RXD2. Only one bit can be set at one time.

VDACEN Voltage DAC Output Enable bit. VDAC is an analog circuit output therefore ANEN in IOCFGP2.3 must also be enabled.

RXD2EN RXD2 Enable bit. Set this bit to enable the P2.3 pin as RXD2 and PINT0.3. DISABLE is the default value.

GPIOEN P2.3 GPIO Function Enable bit. Set this bit to enable P2.3's GPIO function. DISABLE is the default value.

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

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

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

T2EXEN T2EX Enable bit. Set this bit to enable the P2.4 pin as T2EX. DISABLE is the default value.

GPIOEN P2.4 GPIO Function Enable bit. Set this bit to enable P2.4's GPIO function. DISABLE is the default value.

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

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

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

T2EN T2 Enable bit. Set this bit to enable the P2.5 pin as T2. DISABLE is the default value.

GPIOEN P2.5 GPIO Function Enable bit. Set this bit to enable P2.5's GPIO function. DISABLE is the default value.

Set the MFCFG2.5 to 00h and IOCFG2.5 as 08h (ANEN=1) for this pin used as input of comparator C.

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

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

GPIOEN P2.6 GPIO Function Enable bit. Set this bit to enable P2.6's GPIO function. DISABLE is the default value.

Set the MFCFG2.6 to 00h and IOCFG2.6 as 08h (ANEN=1) for this pin used as input of comparator B.

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

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

GPIOEN P2.7 GPIO Function Enable bit. Set this bit to enable P2.7's GPIO function. DISABLE is the default value.

Set the MFCFG2.7 to 00h and IOCFG2.7 as 08h (ANEN=1) for this pin used as input of comparator A.

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

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

P3.0 is shared with PCA CEX2. Only one bit can be set at one time.

CEX2EN CEX2 Enable bit. Set this bit to enable the P3.0 pin as CEX2 for PCA peripheral. DISABLE is the default value.

GPIOEN P3.0 GPIO Function Enable bit. Set this bit to enable P3.0's GPIO function. DISABLE is the default value.

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

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

P3.1 is shared with PCA CEX3. Only one bit can be set at one time.

CEX3EN CEX3 Enable bit. Set this bit to enable the P3.1 pin as CEX3 for PCA peripheral. DISABLE is the default value.

GPIOEN P3.1 GPIO Function Enable bit. Set this bit to enable P3.1's GPIO function. DISABLE is the default value.

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

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

P3.2 is shared with PCA CEX4. Only one bit can be set at one time.

VDACEN Voltage DAC Output Enable bit. VDAC is an analog circuit output therefore ANEN in IOCFGP3.2 must also be enabled.

CEX4EN CEX4 Enable bit. Set this bit to enable the P3.2 pin as CEX4 for PCA peripheral. DISABLE is the default value.

GPIOEN P3.2 GPIO Function Enable bit. Set this bit to enable P3.2's GPIO function. DISABLE is the default value.

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

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

P3.3 is shared with PCA CEX5. Only one bit can be set at one time.

CEX5EN CEX5 Enable bit. Set this bit to enable the P3.3 pin as CEX5 for PCA peripheral. DISABLE is the default value.

GPIOEN P3.3 GPIO Function Enable bit. Set this bit to enable P3.3's GPIO function. DISABLE is the default value.

MFCFGP3.4 (0xA07C) R/W (0x00)

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

P3.4 is shared with SPI SS. Only one bit can be set at one time.

SSEN Set this bit to enable the P3.4 pin as CEB (Chip Enable asserted low) for SPI peripheral when SPI is configured as Slave. DISABLE is the default value.

GPIOEN P3.4 GPIO Function Enable bit. Set this bit to enable P3.4's GPIO function. DISABLE is the default value.

MFCFGP3.5 (0xA07D) R/W (0x00)

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

P3.5 is shared with SPI MOSI. Only one bit can be set at one time.

MOSIEN MOSI Enable bit. Set this bit to enable the P3.5 pin as MOSI for SPI peripheral. DISABLE is the default value.

GPIOEN P3.5 GPIO Function Enable bit. Set this bit to enable P3.5's GPIO function. DISABLE is the default value.

MFCFGP3.6 (0xA07E) R/W (0x00)

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

P3.6 is shared with SPI MISO. Only one bit can be set at one time.

MISOEN MISO Enable bit. Set this bit to enable the P3.6 pin as MISO for SPI peripheral. DISABLE is the default value.

GPIOEN P3.6 GPIO Function Enable bit. Set this bit to enable P3.6's GPIO function. DISABLE is the default value.

MFCFGP3.7 (0xA07F) R/W (0x00)

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

P3.7 is shared with SPI SPICLK. Only one bit can be set at one time.

SPICLKEN SPICLK Enable bit. Set this bit to enable the P3.7 pin as SPICLK for SPI peripheral. DISABLE is the default value.

GPIOEN P3.7 GPIO Function Enable bit. Set this bit to enable P3.7's GPIO function. DISABLE is the default value.

MFCFGP4.0 (0xA0D0) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	CANRXEN	CECSBEN	CECEN	GPIOEN
WR	-	-	-	-	CANRXEN	CECSBEN	CECEN	GPIOEN

P4.0 is shared with CEC. Only one bit can be set at one time.

CANRXEN Set it to enable CAN serial receiver input function through P4.0.Setting this bit will also enable the PINT0.2 for remote wakeup by CAN bus.

CECSBEN CEC Switch-back Enable bit. Set it to enable system switch back from STOP mode by CEC pin receive a "low".

CECEN CEC Enable bit. Set it to configure this pin as input/output pin of the on-chip CEC controller, and attach capability to wake up the IS31CS8968A from STOP mode.

GPIOEN P4.0 GPIO Function Enable bit. Set this bit to enable P4.0's GPIO function. DISABLE is the default value.

MFCFGP4.1 (0xA0D1) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	SSDA2EN	GPIOEN
WR	-	-	-	-	-	-	SSDA2EN	GPIOEN

P4.1 is shared with SSDA2EN. Only one bit can be set at one time.

SSDA2EN SSDA2 Enable bit. Set it to configure this pin as SDA pin of the second I²C slave controller.

GPIOEN P4.1 GPIO Function Enable bit. Set this bit to enable P4.1's GPIO function. DISABLE is the default value.

MFCFGP4.2 (0xA0D2) R/W (0x00)

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

GPIOEN P4.2 GPIO Function Enable bit. Set this bit to enable P4.2's GPIO function. DISABLE is the default value.

MFCFGP4.3 (0xA0D3) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	-	SSCL2EN	GPIOEN
WR	-	-	-	-	-	-	SSCL2EN	GPIOEN

P4.3 is shared with SSCL2EN. Only one bit can be set at one time.

SSCL2EN SSCL2 Enable bit. Set it to configure this pin as SCL pin of the second I²C slave controller.

GPIOEN P4.3 GPIO Function Enable bit. Set this bit to enable P4.3's GPIO function. DISABLE is the default value.

MFCFGP4.4 (0xA0D4) R/W (0x00)

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

GPIOEN P4.4 GPIO Function Enable bit. Set this bit to enable P4.4's GPIO function. DISABLE is the default value.

MFCFGP4.5 (0xA0D5) R/W (0x00)

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

GPIOEN P4.5 GPIO Function Enable bit. Set this bit to enable P4.5's GPIO function. DISABLE is the default value.

MFCFGP4.6 (0xA0D6) R/W (0x00)

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

GPIOEN P4.6 GPIO Function Enable bit. Set this bit to enable P4.6's GPIO function. DISABLE is the default value.

Set the MFCFG4.6 to 00h and IOCFG4.6 as 08h (ANEN=1) for this pin used as input of external threshold for comparator B/C/D.

MFCFGP4.7 (0xA0D7) R/W (0x00)

	7	6	5	4	3	2	1	0
RD	-	-	-	-	-	CANTXEN	PWM16EN	GPIOEN
WR	-	-	-	-	-	CANTXEN	PWM16EN	GPIOEN

P4.7 is shared with PWM16. Only one bit can be set at one time.

CANTXEN Set it to configure P4.7as CAN serial transmitter output pin

PWM16EN PWM16 Enable bit. Set it to configure this pin as PWM16 output pin of the dedicated 16-bit PWM controller.

GPIOEN P4.7 GPIO Function Enable bit. Set this bit to enable P4.7's GPIO function. DISABLE is the default value.

MFCFGP5.6 (0xA0BE) R/W (0x00)

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

GPIOEN P5.6 GPIO Function Enable bit. Set this bit to enable P5.6's GPIO function. DISABLE is the default value.

Set the MFCFG5.6 to 00h and IOCFG5.6 as 08h (ANEN=1) for this pin used as RXIN for RTC.

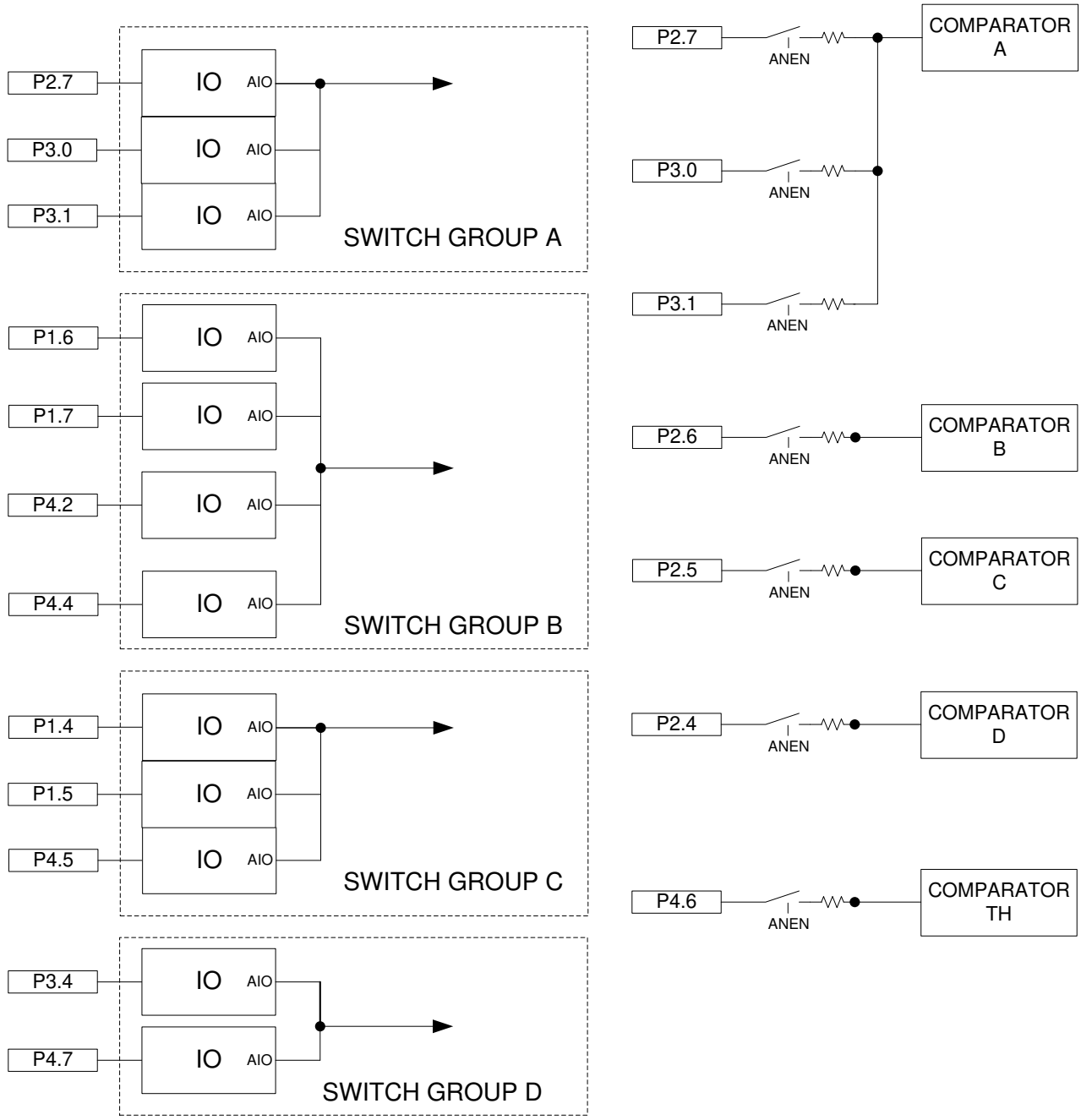
MFCFGP5.7 (0xA0BF) R/W (0x00)

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

GPIOEN P5.7 GPIO Function Enable bit. Set this bit to enable P5.7's GPIO function. DISABLE is the default value.

Set the MFCFG5.7 to 00h and IOCFG5.7 as 08h (ANEN=1) for this pin used as RXOUT for RTC.

Using the analog switch within the PIN IO circuits, it is possible to form analog multiplexer function. In IS31CS8968A, 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.

Set MFCFGx.y as 00h for those pins used for RTC analog switches or analog comparators.

16. IFB Block and Writer Mode and Boot Code/ISP

16.1 IFB Block

The main flash memory is 128Kx8 and also contains a separate 256B Information Block (IFB). The IFB is partitioned into two parts. 00 to 3F range contains critical manufacture and calibration information. And 40 to FF range is user data which is one time programmable. The IFB can not be erased but can be programmable through Flash Controller Command. IFB can be erased and written through Writer mode under privileged operations only, and should not be done by the user. The user data portion can only serves as One-Time-Programmable storage by the user. The following table shows the IFB contents.

ADDRESS	TYPE	DESCRIPTION
00 – 01	M	IFB Version. 0x00 is MSB and 0x01 is LSB.
02 – 07	M	Product Name. 0x02 and 0x03 is “CS”. These fields use ASCII coding.
08 – 09	M	Package and Product Code
0A – 0B	M	Product Version and Revision. These fields use ASCII coding.
0C – 0F	M	Customer Specific Code. These fields use ASCII coding.
10	M	CP1 Information. This byte is written as 0x00 if CP1 is performed, otherwise 0xFF is present.
11	M	CP2 Information. This byte is written as 0x00 if CP2 is performed, otherwise 0xFF is present.
12	M	CP3 Version. This byte is written a value not equal to 0xFF to indicate CP3 is performed.
13	M	CP3 BIN. The value is binary coding. If CP3 is not performed, this value is 0xFF.
14	M	FT Version. This byte is written a value not equal to 0xFF to indicate FT is performed.
15	M	FT BIN. The value is binary coding. If FT is not performed, this value is 0xFF.
16 – 1B	M	Last Test Date. These fields use ASCII coding.
1C – 1D	M	Boot Code Version. These two bytes contain the boot code version with two digits in ASCII code.
1E	M	Boot Code Segment. This byte contains the MSB byte of the address of the last instruction of the boot code. If no boot code is written, this value is 0xFF.
1F	M	Checksum for 0x00 – 0x1E. Checksum is the XOR results of all bytes.
20	M	REGTRM value for 1.8V
21	M	IOSC ITRM value for 16MHz
22	M	IOSC VTRM value for 16MHz
23	M	LVDTHD value for detection of 4.0V
24	M	LVDTHD value for detection of 3.0V. If not calibrated, this byte should be written 0x00.
25	M	IOSC ITRM value for 12MHz. If not calibrated, this byte should be written 0x00.
26	M	IOSC VTRM value for 12MHz. If not calibrated, this byte should be written 0x00.
27	M	IOSC ITRM value for 8MHz. If not calibrated, this byte should be written 0x00.
28	M	IOSC VTRM value for 8MHz. If not calibrated, this byte should be written 0x00.
29	M	IOSC ITRM value for 4MHz. If not calibrated, this byte should be written 0x00.
2A	M	IOSC VTRM value for 4MHz. If not calibrated, this byte should be written 0x00.
2B – 2C	M	Temperature Offset LSB/MSB. The ADC should use VDD18 as the full-scale reference. 0x2B is LSB and 0x2C is MSB. The upper 4-bit of the MSB is the offset of the calibration temperature from 20C. If not calibrated, these bytes should be written as 0x00.
2D	M	Temperature Coefficient. This byte contains the calibrated temperature coefficient for 10C in LSB of ADC. This is a binary number in 4.4 formats, i.e. the radix point is between bit 4 and bit 3. For example, 0b1011.1010 refers to 11.625 LSB. If calibration of temperature sensor is not done, then these two bytes should be written as 0x00.
2E – 2F	M	Internal Reference LSB/MSB
30 – 38	M	Reserved. These bytes are reserved for future extensions and should be written as 0x00.

ADDRESS	TYPE	DESCRIPTION
39	M	Checksum for 0x20 – 0x39. Checksum is the XOR results of all bytes.
3A – 3F	M	Retention Value. These bytes are used to check general flash retention conditions.
40	M/U	Boot Code Wait Time. Boot code uses this byte to determine the ISP wait-time. This wait-time is necessary for stable ISP. After user program download, the wait time can be reduced to minimize power-on time. Each "1" in bit[7-4] constitute 2 second, each "1" in bit[3-0] constitute 1 second. For example, 0b01001110 is 5 second wait time. The wait time can only be reduced from the current value by writing "0" to the bit locations. Default wait time is at its maximum of 12 second.
41 – FF	U	User One-Time Programmable Space

Note 1: M data cannot be modified and can only be written in writer mode when the entire Flash is erased.

Note 2: U data reads out as FF after the Flash is erased. It can only be programmed once after the Flash is mass erased.

****** The erasure of IFB or modifications of manufacture information in IFB void any manufacture warranty.

**** This table is for reference only. Please refer to most updated AP note and boot code documents.

16.2 Writer Mode

Writer Mode (WM) is used by the manufacturer or by users to program the flash (including IFB) through a dedicated hardware (Writer or Gang Writer). Under this set up, only WM related pins should be connected and all other unused pins left floating. Writer mode follows a proprietary protocol and is not released to general users. Users must obtain it through a formal written request to the manufacturer and must sign a strict Non-Disclosure-Agreement.

The Writer Mode provides the following commands.

ERASE Main Memory

ERASE Main Memory and IFB

READ AND VERIFY Main Memory (8-Byte)

WRITE BYTE Main Memory

READ BYTE IFB

WRITE BYTE IFB

Fast Continuous WRITE

Fast Continuous READ

The writer mode is protected against code piracy. The power-on state of the device deactivates the writer mode. Only ERASEMM and ERASEMMIFB, and READVERIFYMM commands can be executed. It is activated by READVERIFY the range of 0x0FFF8 to 0x0FFFF where a security key can be placed by the user program. The probability of guessing the key is 1 in 2^64 = 1.8E19. Since each trial of READVERIFY takes 10usec, it takes about 6E6 years to exhaust the combinations. If the key is unknown, a user can choose to issue the ERASEMM command then fully erase the entire contents (including the key). Once fully erased, all data in the flash is 0xFF, and it can be successfully unlocked by READVERIFYMM with 8-bytes of 0xFF.

The users must not erase the information in IFB. And the user should not modify the manufacturer data. Any violation of this results in the void of manufacturer warranty.

16.2.1 Writer Mode Pins

PIN	PIN#	TYPE	WRITER MODE PIN DESCRIPTION
VDD	38,14	P	VDD should be connected to a solid 5.0V supply with good decoupling to VSS
VSS	36,13	G	Tie to 0V and have good decoupling to VDD
VDD18	37	PO	Have a 4.7uF and a 0.1uF good decoupling to VSS
RSTN	39	IN	Pull to 5.0V through 1KOhm to enter to WRITER Mode.
P2.3	4	O	BUSY status
P2.4	41	O	TDO Data Output
P2.5	42	I	TDI Data Input
P2.6	43	I	TCLK Clock Input
P2.7	44	I	TENB Test Enable Input. Low assertion.

16.3 Boot Code and In-System-Programming

After production testing of the packaged devices, the manufacture writes the manufacturer information and calibration data in the IFB. At the last stage, it writes a fixed boot-code in the main memory residing from 0x0F000 to 0xF7FF. The boot code is executed after resets. Firstly, the program reads the boot code value from 0x0F7F0 to 0xF7FF. If any value of these bytes is not 0xFF, it skips the remaining of the boot code and jumps to 0x0000 as a normal 8051 reset. If all bytes are in 0x0F7F0 to 0xF7FF are 0xFF, the boot code scans the I^2C slave 0 and 1, as well as UART0 for any In-System-Programming request. This scanning takes about 10msec. If any valid request is valid during the scan, the boot-code proceeds to follow the request and performs the programming from the host. The default ISP commands available are

UNLOCK
DEVICE NAME
BOOTC VERSION
READ AND VERIFY Main Memory (8-Byte)
ERASE Main Memory exclude Boot Code
ERASE SECTOR Main Memory
WRITE BYTE Main Memory
SET ADDRESS
CONTINUOUSE WRITE
CONTINUOUS READ
READ BYTE IFB
WRITE BYTE IFB

Similar to writer mode, ISP is in default locked state. No command is accepted under locked state. To unlock the ISP, an 8-byte READVERIFY of 0x0FFF8 to 0x0FFFF must be successfully executed. Thus default ISP boot program provides similar code security as the Writer mode.

17. Electrical Characteristics

17.1 Absolute Maximum Ratings

SYMBOL	PARAMETER	RATING	UNIT	NOTE
VDD	Supply Voltage	5.5	V	
TA	Ambient Operating Temperature	-40 – 85	℃	
TSTG	Storage Temperature	-65 – 150	℃	

17.2 Recommended Operating Condition

SYMBOL	PARAMETER	RATING	UNIT	NOTE
VDD	Supply Voltage for IO and 1.8V regulator	2.5 - 5.5	V	
TA	Ambient Operating Temperature	-40 – 85	℃	

17.3 DC Electrical Characteristics (VDDHIO=VDDHA=4.5V to 5.5V TA=-40℃ to 85℃)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT	NOTE
Power Supply Current						
IDD, Normal	Total IDD through VDD at 16MHz, normal mode	-	8	-	mA	
IDDVF	Total IDD Core Current versus frequency, normal mode	-	0.5	-	mA/MHz	
IDD, Stop	IDD, stop mode	-	250	-	uA	Main regulator on
IDD, Sleep	IDD, sleep mode	-	20	-	uA	Main regulator off
RSTN Reset						
VIHRS	Input High Voltage	+1.2	-	-	V	
VILRS	Input Low Voltage	-	-	0.5	V	
VRSHYS	RSTN Hysteresis	-	0.7V	-	V	VDD18=1.8V
GPIO DC Characteristics						
VOH,4.5V	Output High Voltage 1 mA	-	-0.2	-0.5	V	Reference to VDD
VOH,4.5V	Output High Voltage 2 mA	-	-0.3	-0.7	V	Reference to VDD
VOL,4.5V	Output Low Voltage 4 mA	-	0.2	0.4	V	Reference to VSS
VOL,4.5V	Output Low Voltage 8 mA	-	0.3	0.5	V	Reference to VSS
VOH,3.0V	Output High Voltage 1 mA	-	-0.3	-0.6	V	Reference to VDD
VOH,3.0V	Output High Voltage 2 mA	-	-0.4	-0.8	V	Reference to VDD
VOL,3.0V	Output Low Voltage 4 mA	-	0.2	0.4	V	Reference to VSS
VOL,3.0V	Output Low Voltage 8 mA	-	0.3	0.6	V	Reference to VSS
VIH	Input High Voltage	2.2	-	-	V	Reference to 0.5VDD
VIL	Input Low Voltage	-	-	1.1	V	Reference to 0.5VDD
VIHYS	Input Hysteresis	100	300	600	mV	
RPU	Equivalent Pull-Up resistance, 3.3V	-	350K	-	Ohm	
	Equivalent Pull-Up resistance, 5.0V	-	200K	-	Ohm	
RPD	Equivalent Pull-Down Resistance, 3.3V	-	200K	-	Ohm	
	Equivalent Pull-Down Resistance, 5.0V	-	125K	-	Ohm	
RPULAT	Equivalent Pull-Up Resistance for Latch	-	10K	-	Ohm	Measured at VDDHIO
RPDLAT	Equivalent Pull-Down Resistance for Latch	-	5K	-	Ohm	Measured at 0V
RAN	Equivalent Resistance for Analog Switch 3.3V	-	330	-	Ohm	
	Equivalent Resistance for Analog Switch 5V	-	240	-	Ohm	
SAR ADC						
VINSAR	Input DC Range	0	-	VDD-1.2V	V	REF=VDD, FS=VDD
		0	-	VDD 18	V	REF=VDD18
LINSARM	SAR ADC Accuracy	-	+/- 2	-	LSB	After calibration

	0.5V to REF-0.5V					
FADC	ADC maximum frequency	-	-	4	MHz	VDD>= 3.0V
		-	-	1	MHz	VDD < 3.0V
TCONV	ADC conversion time	-	8	-	usec	VDD >= 3.0V
		-	32	-	usec	VDD < 3.0V
VOLTAGE DAC						
VOUT	Output Range	0	-	¾ VDD	V	For normal accuracy
LINDAC	ADAC Accuracy	-	+/- 2	-	LSB	Normal output range
		-	+/- 10	-	LSB	0 – 0.5V
		-	+/- 12	-	LSB	¾ VDD to VDD
Low Supply (VDDHR) Voltage Detection						
VDET	Detection Range	2.2	-	5.0	V	Setting by LVDTHD
VDETHYS	Detection Hysteresis		100	-	mV	

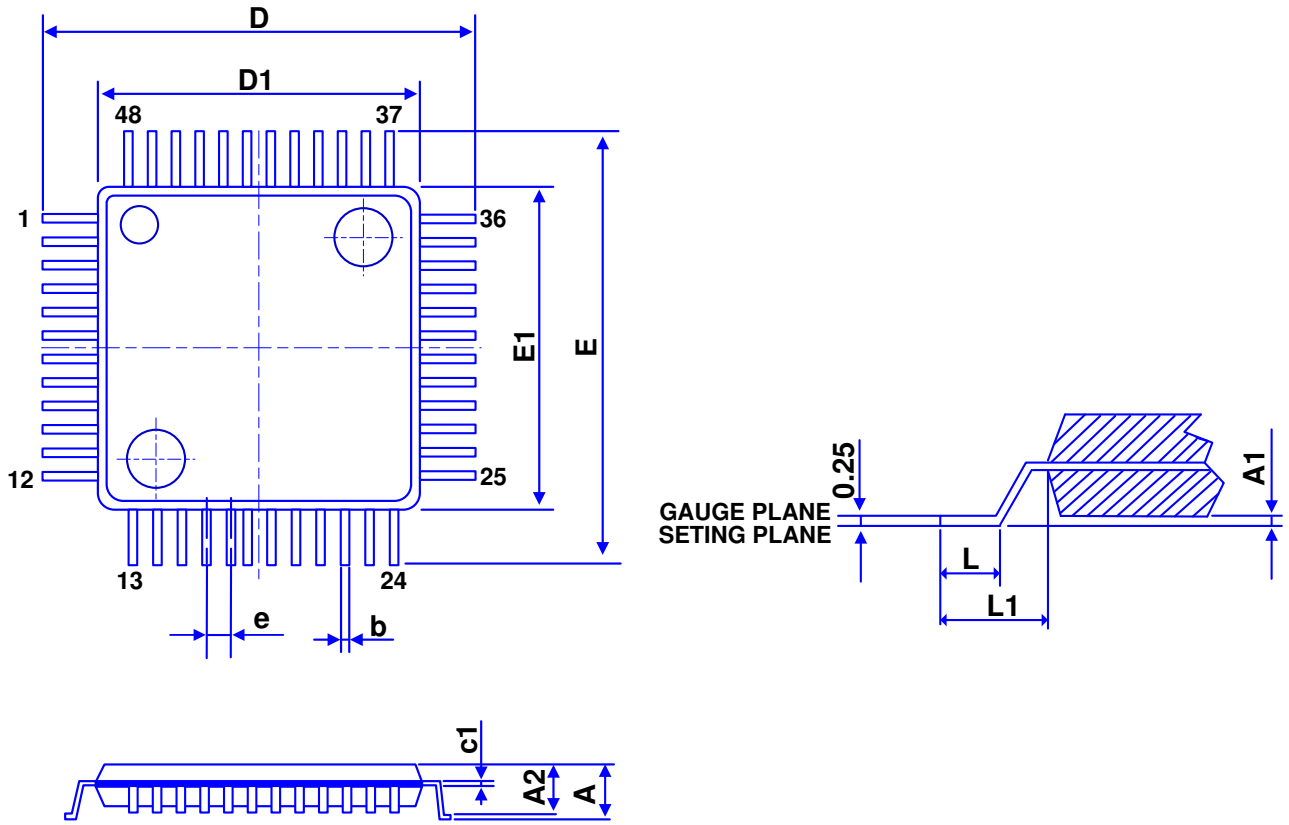
17.4 AC Electrical Characteristics (VDD =3.0V to 5.5V TA=-40℃ to 85℃)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT	NOTE
System Clock and Reset						
FSYS	System Clock Frequency	-	16	25	MHz	
FXOSC	Crystal Oscillator Frequency	1	16	25	MHz	
TSXOSC	Stable Time for XOSC after power up	50	-	-	msec	VDD > 3.0V
Supply Timing						
TSUPRU	Maximum VDD Ramp Up time	-	-	50	msec	
TSUPRD	Maximum VDD Ramp Down Time	-	-	50	msec	
TPOR	Power On Reset Delay	-	10	-	msec	IOSC=16MHz
IOSC						
FIOSC	IOSC calibrated 16MHz	-1	0	+1	%	
	Temperature and VDD variation	-2	0	+2	%	
SIOSC						
TPOR	Power On Reset Delay	-	10	-	msec	IOSC=16MHz
IO Timing						
TPD3 ++	Propagation Delay 3.3V No load	-	6	-	nsec	
TPD3 ++	Propagation Delay 3.3V 25pF load	-	15	-	nsec	
TPD3 ++	Propagation Delay 3.3V 50pF load	-	20	-	nsec	
TPD3 --	Propagation Delay 3.3V No load	-	5	-	nsec	
TPD3 --	Propagation Delay 3.3V 25pF load	-	12	-	nsec	
TPD3 --	Propagation Delay 3.3V 50pF load	-	15	-	nsec	
TPD5 ++	Propagation Delay 3.3V No load	-	5	-	nsec	
TPD5 ++	Propagation Delay 3.3V 25pF load	-	12	-	nsec	
TPD5 ++	Propagation Delay 3.3V 50pF load	-	16	-	nsec	
TPD5 --	Propagation Delay 3.3V No load	-	4	-	nsec	
TPD5 --	Propagation Delay 3.3V 25pF load	-	9	-	nsec	
TPD5 --	Propagation Delay 3.3V 50pF load	-	12	-	nsec	
Flash Memory Timing						
TEMAC	Embedded Flash Access Time	-	35	45	nsec	TWAIT must > TEMAC
TEMWR	Embedded Flash Write Time	-	20	25	usec	
TEMSER	Embedded Flash Sector Erase Time	-	2	2.5	msec	
TEMMER	Embedded Flash Mass Erase Time	-	10	12	msec	
SAR ADC						
FSARADC	Maximum SAR ADC Frequency	-	-	4	MHz	
TSARADC	Conversion time of SAR ADC	-	16	-	Cycle	ADC clock cycles

TSADCSE	Set up time for SAR ADC channel select	250	-	-	nsec	
Analog Comparator						
TDACMP	Analog comparator delay	-	-	250	nsec	

18. PACKAGE OUTLINE

18.1 48-pin LQFP



SYMBOL	DIMENSIONS IN MILLIMETERS	
	MIN.	MAX.
A	-	1.6
A1	0.05	0.15
A2	1.35	1.45
c1	0.09	0.16
D	9.00 BSC	
D1	7.00 BSC	
E	9.00 BSC	
E1	7.00 BSC	
e	0.5 BSC	
b	0.17	0.27
L	0.45	0.75
L1	1 REF	

Notes:

1. *JEDEC outline: MS-026 BBC.*
2. *Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. D1 and E1 are maximum plastic body size dimensions including mold mismatch.*
3. *Dimension b does not include dambar protrusion. Allowable dambar protrusion shall not cause the lead width to exceed the maximum b dimension by more than 0.08mm.*

19. ORDERING INFORMATION

Operating temperature -40°C to 85°C

Order Part No.	Package	QTY
IS31CS8968AG-LQLS2	LQFP-48, Lead-free	250/Plate

Revision History

Revision A (For H2)

1. Formal release version

V0.98 (For H2)

1. Change to ISSI naming rule and ordering information

V0.97 (For H2)

1. Revise WD[2-0] default to 011 in CKCON SFR
2. CLSR/CLIR (0xF2) PCA Snapshot Register is RW not RO

V0.96 (For H2)

1. Add DBPCIDL, DBPCIDH, DBPCIDT, and DBPCNXL, DBPCNXH, DBPCNXT
2. Revise I2CMCR and INTPCT2
3. Update the descriptions in the General Description, Features, Pin Descriptions, WTST, Dual Data Pointers, SPI, PCA, and LVD.
4. Update Flash Controller.
5. Add REFSEL comment in power management and ADC.
6. Revise ACMP and the analog switch figure.
7. Revise IFB table, Writer Mode, and Boot Code with ISP
8. Update the DC/AC Electrical Characteristics

V0.95 (For H0 since V0.95)

1. Spelling correction
2. LVDTH 7-bit
3. Change IDAC to VDAC
4. Add and modify WKDLY description
5. Revise Register Map for DACH and DACL and add VDACEN into MFCFGP2.3[7] and MFCFGP3.2[7]
6. Revise maximum ADC clock rate of 4MHz
7. Revise Stop IDD from 150uA to 250uA
8. Add MI2C maximum SCL frequency description.
9. Modify CNTPCTL and CNTPCTH description in Flash Controller
10. Add CEC into INT7 and update CEC description
11. Add REFSEL into ADCAVG[3]
12. Revise SI2C1, SI2C2, and SI2CDBGID
13. Revise EUART2/LIN Controller
14. Enable WDT with the max. timeout value in default
15. Add CAN interrupt into INT3
16. Change LVDTH max. threshold voltage from 5V to 4.8V