

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	I ² C, IRSCI, SPI
Peripherals	LCD, LVD, POR, PWM
Number of I/O	40
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908lk24cfue

Table of Contents

Section 1. General Description

1.1	Contents	37
1.2	Introduction	38
1.3	Features	38
1.4	MCU Block Diagram	40
1.5	Pin Assignments	42
1.6	Pin Functions	44
1.6.1	Power Supply Pins (V_{DD} and V_{SS})	44
1.6.2	Analog Power Supply Pin (V_{DDA})	44
1.6.3	LCD Bias Voltage (V_{LCD})	45
1.6.4	Oscillator Pins (OSC1 and OSC2)	45
1.6.5	External Reset Pin (\overline{RST})	45
1.6.6	External Interrupt Pin (\overline{IRQ})	45
1.6.7	External Filter Capacitor Pin (CGMXFC)	45
1.6.8	ADC Voltage High Reference Pin (V_{REFH})	45
1.6.9	ADC Voltage Low Reference Pin (V_{REFL})	46
1.6.10	Port A Input/Output (I/O) Pins (PTA7–PTA0)	46
1.6.11	Port B I/O Pins (PTB7–PTB0)	46
1.6.12	Port C I/O Pins (PTC7–PTC0)	46
1.6.13	Port D I/O Pins (PTD7–PTD0)	46
1.6.14	Port E I/O Pins (PTE7–PTE0)	47
1.6.15	Port F I/O Pins (PTF7–PTF0)	47
1.6.16	LCD Backplane and Frontplane (BP0–BP2, BP3/FP0, FP1–FP10, FP27–FP32)	47

Section 12. Real Time Clock (RTC)

12.1	Contents	217
12.2	Introduction	218
12.3	Features	218
12.4	I/O Pins	219
12.5	Functional Description	221
12.5.1	Time Functions	223
12.5.2	Calendar Functions	223
12.5.3	Alarm Functions	223
12.5.4	Chronograph Functions	223
12.5.5	Timebase Interrupts	224
12.6	RTC Interrupts	224
12.7	RTC Clock Calibration and Compensation	225
12.7.1	Calibration Error	226
12.8	RTC Register and Bit Write Protection	227
12.9	Low-Power Modes	229
12.9.1	Wait Mode	229
12.9.2	Stop Mode	230
12.10	RTC Registers	230
12.10.1	RTC Calibration Control Register (RTCCOMR)	231
12.10.2	RTC Calibration Data Register (RTCCDAT)	233
12.10.3	RTC Control Register 1 (RTCCR1)	234
12.10.4	RTC Control Register 2 (RTCCR2)	235
12.10.5	RTC Status Register (RTCSR)	237
12.10.6	Alarm Minute and Hour Registers (ALMR and ALHR) ...	240
12.10.7	Second Register (SECR)	241
12.10.8	Minute Register (MINR)	241
12.10.9	Hour Register (HRR)	242
12.10.10	Day Register (DAYR)	242
12.10.11	Month Register (MTHR)	243
12.10.12	Year Register (YRR)	243
12.10.13	Day-Of-Week Register (DOWR)	244
12.10.14	Chronograph Data Register (CHRR)	244

Section 4. FLASH Memory (FLASH)

4.1 Contents

4.2	Introduction	69
4.3	Functional Description	70
4.4	FLASH Control Register	71
4.5	FLASH Page Erase Operation	72
4.6	FLASH Mass Erase Operation	73
4.7	FLASH Program Operation.	74
4.8	FLASH Block Protection	76
4.8.1	FLASH Block Protect Register	77

4.2 Introduction

This section describes the operation of the embedded FLASH memory. This memory can be read, programmed, and erased from a single external supply. The program and erase operations are enabled through the use of an internal charge pump.

8.7	Interrupts	133
8.8	Special Modes	133
8.8.1	Wait Mode	133
8.8.2	Stop Mode	134
8.8.3	CGM During Break Interrupts	134
8.9	Acquisition/Lock Time Specifications	135
8.9.1	Acquisition/Lock Time Definitions	135
8.9.2	Parametric Influences on Reaction Time	135
8.9.3	Choosing a Filter	137

8.2 Introduction

This section describes the clock generator module (CGM). The CGM generates the base clock signal, CGMOUT, which is based on either the oscillator clock divided by two or the divided phase-locked loop (PLL) clock, CGMPCLK, divided by two. CGMOUT is the clock from which the SIM derives the system clocks, including the bus clock, which is at a frequency of $\text{CGMOUT} \div 2$.

The PLL is a frequency generator designed for use with a low frequency crystal (typically 32.768kHz) to generate a base frequency and dividing to a maximum bus frequency of 8MHz.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$FE00	SIM Break Status Register (SBSR)	Read:						SBSW	R	
		Write:	R	R	R	R	R	Note		
		Reset:	0							
Note: Writing a logic 0 clears SBSW.										
\$FE01	SIM Reset Status Register (SRSR)	Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
\$FE03	SIM Break Flag Control Register (SBFCR)	Read:								
		Write:	BCFE	R	R	R	R	R	R	R
		Reset:	0							
\$FE04	Interrupt Status Register 1 (INT1)	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
			= Unimplemented			R = Reserved				

Figure 9-2. SIM I/O Register Summary

9.3 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in [Figure 9-3](#). This clock can come from either the oscillator module or from the on-chip PLL. (See [Section 8. Clock Generator Module \(CGM\)](#).)

9.6.1.5 Interrupt Status Register 2

Address: \$FE05

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 9-13. Interrupt Status Register 2 (INT2)

IF14–IF7 — Interrupt Flags 14–7

These flags indicate the presence of interrupt requests from the sources shown in [Table 9-3](#).

- 1 = Interrupt request present
- 0 = No interrupt request present

9.6.1.6 Interrupt Status Register 3

Address: \$FE06

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IF18	IF17	IF16	IF15
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 9-14. Interrupt Status Register 3 (INT3)

IF18–IF15 — Interrupt Flags 18–15

These flags indicate the presence of an interrupt request from the source shown in [Table 9-3](#).

- 1 = Interrupt request present
- 0 = No interrupt request present

Timer Interface Module (TIM)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0029	Timer 1 Channel 1 Register High (T1CH1H)	Read:								
		Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
\$002A	Timer 1 Channel 1 Register Low (T1CH1L)	Read:								
		Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
\$002B	Timer 2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$002C	Timer 2 Counter Register High (T2CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$002D	Timer 2 Counter Register Low (T2CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$002E	Timer 2 Counter Modulo Register High (T2MODH)	Read:								
		Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
\$002F	Timer 2 Counter Modulo Register Low (T2MODL)	Read:								
		Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
\$0030	Timer 2 Channel 0 Status and Control Register (T2SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
\$0031	Timer 2 Channel 0 Register High (T2CH0H)	Read:								
		Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
				= Unimplemented						


 = Unimplemented

Figure 11-2. TIM I/O Register Summary (Sheet 2 of 3)

NOTE: *In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.*

11.5.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use the following initialization procedure:

1. In the TIM status and control register (TSC):
 - a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
 - b. Reset the TIM counter and prescaler by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH:TMODL), write the value for the required PWM period.
3. In the TIM channel x registers (TCHxH:TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
 - a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. (See [Table 11-3](#).)
 - b. Write 1 to the toggle-on-overflow bit, TOVx.
 - c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. (See [Table 11-3](#).)

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

The mechanism uses the RTCWE[1:0] bits in the RTC calibration control register (RTCCOMR) in a state machine, which requires a bit-write sequence to disable the write-protection. A block diagram of the state machine is shown in [Figure 12-5](#).

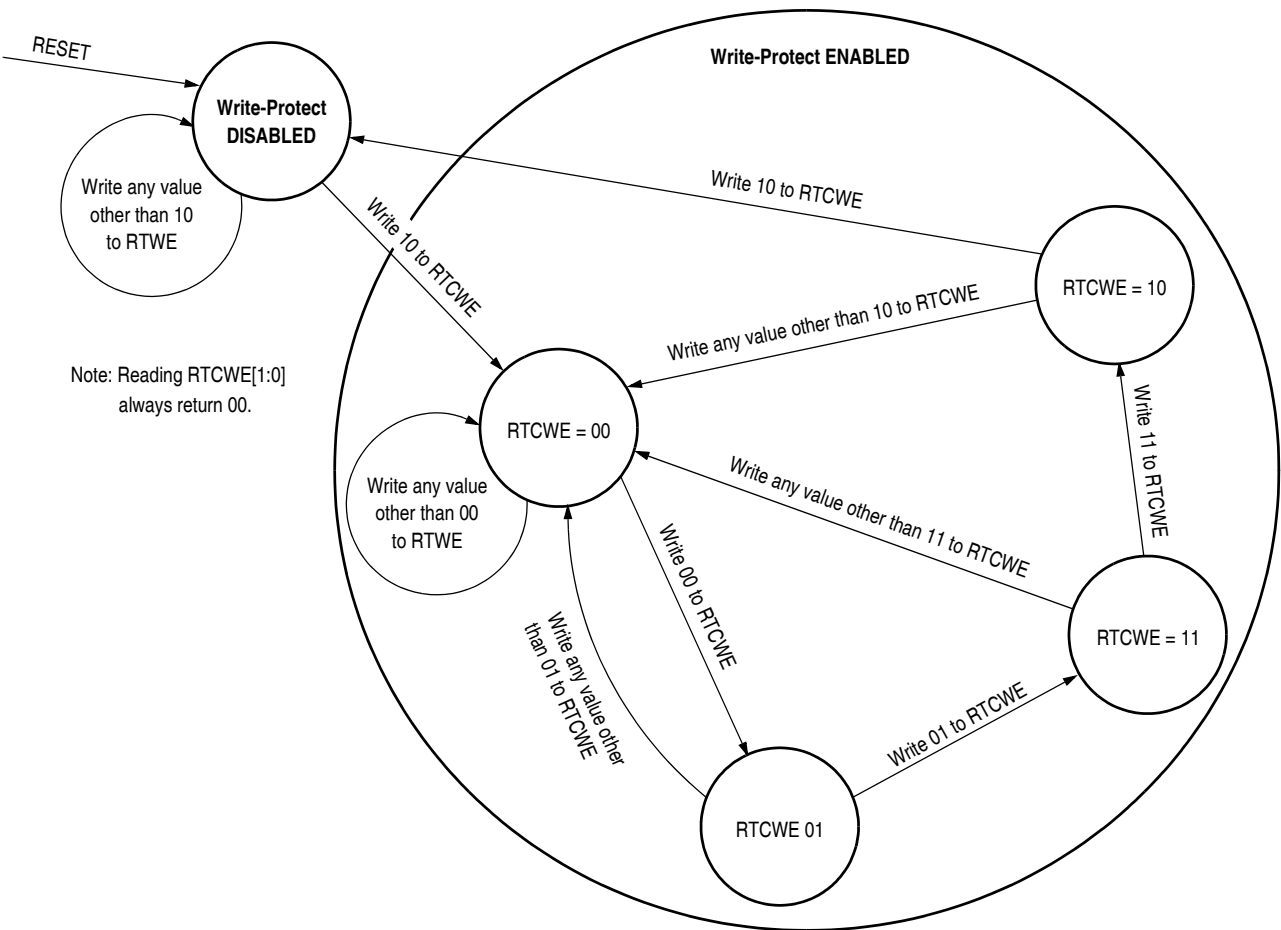


Figure 12-5. RTC Write Protect State Diagram

After a reset, the write-protect mechanism is disabled, allowing the user code to calibrate the RTC clock, set the time in the clock registers, and set the date in the calendar registers.

To enable write-protect after reset or write-protect is disabled execute the following code:

```
RTCWE1 EQU 1 ;RTCWE1 bit
RTCWE0 EQU 0 ;RTCWE0 bit
```

Section 13. Infrared Serial Communications Interface Module (IRSCI)

13.1 Contents

13.2	Introduction	246
13.3	Features	247
13.4	Pin Name Conventions	249
13.5	IRSCI Module Overview	249
13.6	Infrared Functional Description	250
13.6.1	Infrared Transmit Encoder	251
13.6.2	Infrared Receive Decoder	251
13.7	SCI Functional Description	252
13.7.1	Data Format	253
13.7.2	Transmitter	254
13.7.2.1	Character Length	255
13.7.2.2	Character Transmission	255
13.7.2.3	Break Characters	256
13.7.2.4	Idle Characters	256
13.7.2.5	Transmitter Interrupts	257
13.7.3	Receiver	257
13.7.3.1	Character Length	257
13.7.3.2	Character Reception	259
13.7.3.3	Data Sampling	259
13.7.3.4	Framing Errors	261
13.7.3.5	Baud Rate Tolerance	261
13.7.3.6	Receiver Wakeup	264
13.7.3.7	Receiver Interrupts	265
13.7.3.8	Error Interrupts	265
13.8	Low-Power Modes	266
13.8.1	Wait Mode	266
13.8.2	Stop Mode	266

ERRIE — Error Interrupt Enable Bit

This read/write bit enables the MODF and OVRF bits to generate CPU interrupt requests. Reset clears the ERRIE bit.

- 1 = MODF and OVRF can generate CPU interrupt requests
- 0 = MODF and OVRF cannot generate CPU interrupt requests

OVRF — Overflow Bit

This clearable, read-only flag is set if software does not read the byte in the receive data register before the next full byte enters the shift register. In an overflow condition, the byte already in the receive data register is unaffected, and the byte that shifted in last is lost. Clear the OVRF bit by reading the SPI status and control register with OVRF set and then reading the receive data register. Reset clears the OVRF bit.

- 1 = Overflow
- 0 = No overflow

MODF — Mode Fault Bit

This clearable, read-only flag is set in a slave SPI if the \overline{SS} pin goes high during a transmission with the MODFEN bit set. In a master SPI, the MODF flag is set if the \overline{SS} pin goes low at any time with the MODFEN bit set. Clear the MODF bit by reading the SPI status and control register (SPSCR) with MODF set and then writing to the SPI control register (SPCR). Reset clears the MODF bit.

- 1 = \overline{SS} pin at inappropriate logic level
- 0 = \overline{SS} pin at appropriate logic level

SPTE — SPI Transmitter Empty Bit

This clearable, read-only flag is set each time the transmit data register transfers a byte into the shift register. SPTE generates an SPTE CPU interrupt request if the SPTIE bit in the SPI control register is set also.

NOTE: *Do not write to the SPI data register unless the SPTE bit is high.*

During an SPTE CPU interrupt, the CPU clears the SPTE bit by writing to the transmit data register.

Reset sets the SPTE bit.

- 1 = Transmit data register empty
- 0 = Transmit data register not empty

MMAD[7:1] — Multi-Master Address

These seven bits represent the MMIIC interface’s own specific slave address when in slave mode, and the calling address when in master mode. Software must update MMAD[7:1] as the calling address while entering master mode and restore its own slave address after master mode is relinquished. This register is cleared as \$A0 upon reset.

MMEXTAD — Multi-Master Expanded Address

This bit is set to expand the address of the MMIIC in slave mode. When set, the MMIIC will acknowledge the following addresses from a calling master: \$MMAD[7:1], 0000000, and 0001 100. Reset clears this bit.

- 1 = MMIIC responds to the following calling addresses:
\$MMAD[7:1], 0000000, and 0001 100.
- 0 = MMIIC responds to address \$MMAD[7:1]

For example, when MMADR is configured as:

MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
1	1	0	1	0	1	0	1

The MMIIC module will respond to the calling address:

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

or the general calling address:

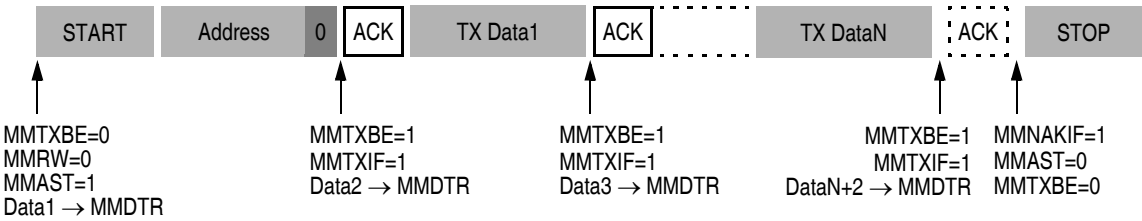
0	0	0	0	0	0	0
---	---	---	---	---	---	---

or the calling address:

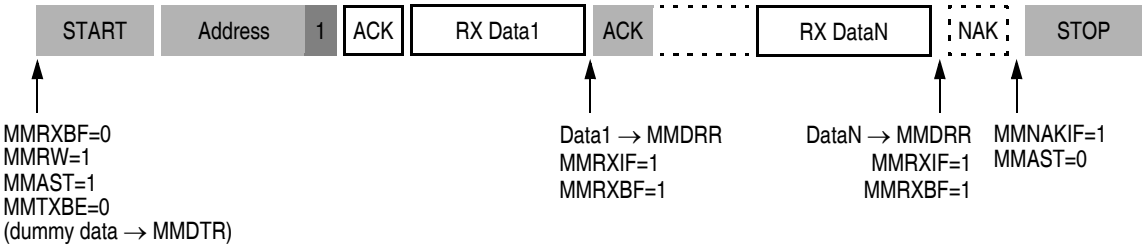
0	0	0	1	1	0	0
---	---	---	---	---	---	---

NOTE: bit-0 of the 8-bit calling address is the MMRW bit from the calling master.

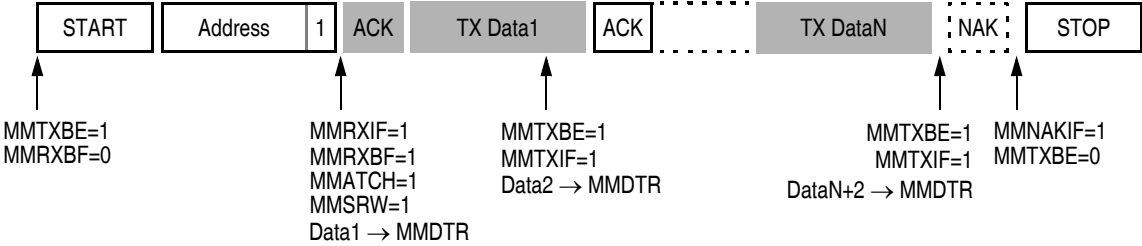
(a) Master Transmit Mode



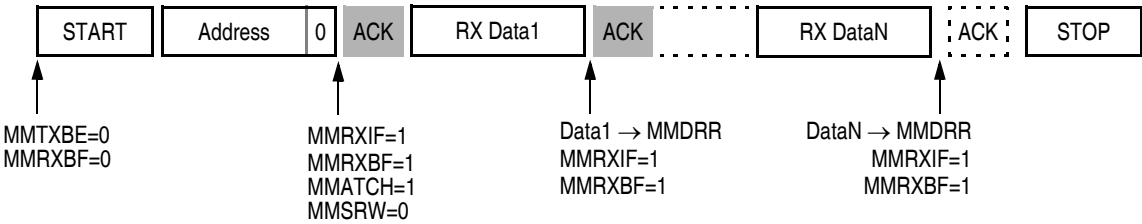
(b) Master Receive Mode



(c) Slave Transmit Mode



(d) Slave Receive Mode



■ Shaded data packets indicate transmissions by the MCU

Figure 15-8. Data Transfer Sequences for Master/Slave Transmit/Receive Modes

16.2 Introduction

This section describes the analog-to-digital convert (ADC). The ADC is a 6-channel 10-bit linear successive approximation ADC.

16.3 Features

Features of the ADC module include:

- Six channels with multiplexed input
- High impedance buffered input
- Linear successive approximation with monotonicity
- 10-Bit resolution
- Single or continuous conversion
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock
- Conversion result justification
 - 8-bit truncated mode
 - Right justified mode
 - Left justified mode
 - Left justified sign mode

17.5.1 LCD Duty

The setting of the LCD output waveform duty is dependent on the number of backplane drivers required. Three LCD duties are available:

- Static duty — BP0 is used only
- 1/3 duty — BP0, BP1, and BP3 are used
- 1/4 duty — BP0, BP1, BP2, and BP3 are used

When the LCD driver module is enabled the backplane waveforms for the selected duty are driven out of the backplane pins. The backplane waveforms are periodic and are shown in [Figure 17-5](#), [Figure 17-6](#), and [Figure 17-7](#).

17.5.2 LCD Voltages (V_{LCD} , V_{LCD1} , V_{LCD2} , V_{LCD3})

The voltage V_{LCD} is from the V_{LCD} pin and must not exceed V_{DD} . V_{LCD1} , V_{LCD2} , and V_{LCD3} are internal bias voltages for the LCD driver waveforms. They are derived from V_{LCD} using a resistor ladder (see [Figure 17-3](#)).

The relative potential of the LCD voltages are:

- $V_{LCD} = V_{DD}$
- $V_{LCD1} = 2/3 \times (V_{LCD} - V_{bias})$
- $V_{LCD2} = 1/3 \times (V_{LCD} - V_{bias})$
- $V_{LCD3} = V_{bias}$

The V_{LCD3} bias voltage, V_{bias} , is controlled by the LCD contrast control bits, $LCCON[2:0]$.

17.5.3 LCD Cycle Frame

The LCD driver module uses the CGMXCLK (see [Section 7. Oscillator \(OSC\)](#)) as the input reference clock. This clock is divided to produce the LCD waveform base clock, LCDCLK, by configuring the LCLK[2:0] bits in the LCD clock register. The LCDCLK clocks the backplane and the frontplane output waveforms.

When DDRFx is a logic 1, reading address \$000A reads the PTFx data latch. When DDRFx is a logic 0, reading address \$000A reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-7 summarizes the operation of the port F pins.

Table 18-7. Port F Pin Functions

DDRF Bit	PTF Bit	I/O Pin Mode	Accesses to DDRF	Accesses to PTF	
			Read/Write	Read	Write
0	X ⁽¹⁾	Input, Hi-Z ⁽²⁾	DDRF[7:0]	Pin	PTF[7:0] ⁽³⁾
1	X	Output	DDRF[7:0]	PTF[7:0]	PTF[7:0]

- Notes:**
1. X = don't care; except.
 2. Hi-Z = high impedance.
 3. Writing affects data register, but does not affect input.

18.8.3 Port F LED Control Register (LEDF)

Port-F LED control register (LEDF) controls the direct LED drive capability on PTF7–PTF0 pins. Each bit is individually configurable and requires that the data direction register, DDRF, bit be configured as an output.

Address: \$000F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LEDF7	LEDF6	LEDF5	LEDF4	LEDF3	LEDF2	LEDF1	LEDF0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 18-23. Port F LED Control Register (LEDF)

LEDF[7:0] — Port F LED Drive Enable Bits

These read/write bits are software programmable to enable the direct LED drive on an output port pin.

- 1 = Corresponding port F pin is configured for direct LED drive, with 15mA current sinking capability
- 0 = Corresponding port F pin is configured for standard drive



19.4 Functional Description

A logic 0 applied to the external interrupt pin can latch a CPU interrupt request. [Figure 19-1](#) shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the IRQ latch.
- Software clear — Software can clear the interrupt latch by writing to the acknowledge bit in the interrupt status and control register (INTSCR). Writing a logic 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears the interrupt latch.

The external interrupt pin is falling-edge-triggered and is software-configurable to be either falling-edge or low-level-triggered. The MODE bit in the INTSCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

When the interrupt pin is edge-triggered only, the CPU interrupt request remains set until a vector fetch, software clear, or reset occurs.

When the interrupt pin is both falling-edge and low-level-triggered, the CPU interrupt request remains set until both of the following occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic 1

The vector fetch or software clear may occur before or after the interrupt pin returns to logic 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK bit in the INTSCR mask all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the IMASK bit is clear.

NOTE: *The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests.*

IMASKK — Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests. Reset clears the IMASKK bit.

- 1 = Keyboard interrupt requests masked
- 0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins. Reset clears MODEK.

- 1 = Keyboard interrupt requests on falling edges and low levels
- 0 = Keyboard interrupt requests on falling edges only

20.6.2 Keyboard Interrupt Enable Register

The keyboard interrupt enable register individually enables or disables the PTA0/KBI0–PTA3/KBI3 and PTD4/KBI4–PTD7/KBI7 pins to operate as a keyboard interrupt pin.

Address: \$001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Reset:	0	0	0	0	0	0	0	0

Figure 20-4. Keyboard Interrupt Enable Register (KBIER)

KBIE7–KBIE0 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch interrupt requests. Reset clears the keyboard interrupt enable register.

- 1 = KBIE pin enabled as keyboard interrupt pin
- 0 = KBIE pin not enabled as keyboard interrupt pin

NOTE: *KBIE5–KBIE0 pin has an internal pullup device when KBIE_x is set.
KBIE7–KBIE6 pin does not have an internal pullup device.*

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

- 1 = (When read) Break address match
- 0 = (When read) No break address match

23.6.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

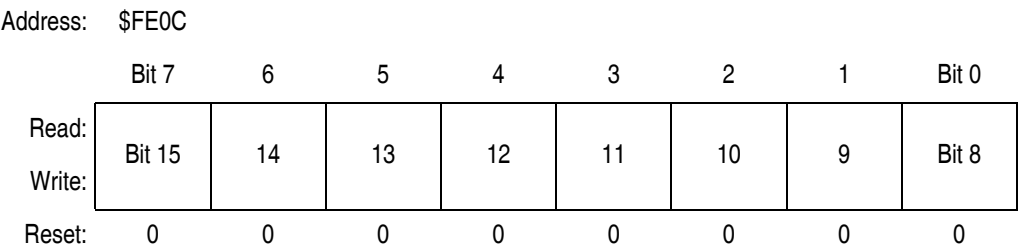


Figure 23-4. Break Address Register High (BRKH)

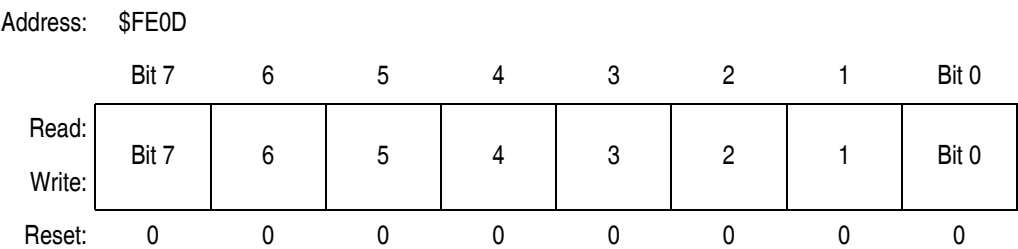


Figure 23-5. Break Address Register Low (BRKL)

23.6.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.