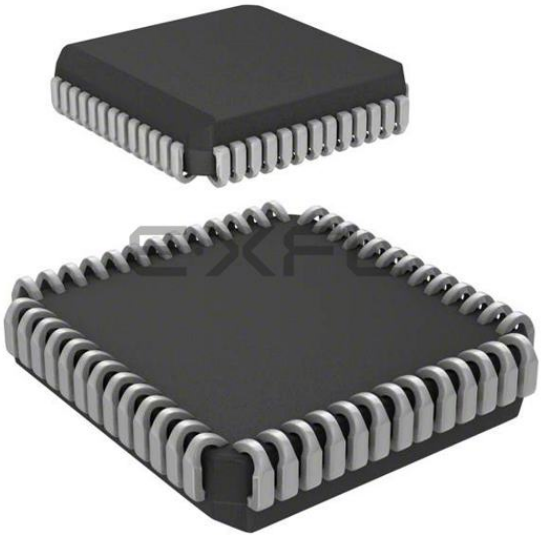


Welcome to E-XFL.COM

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	8.4MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	42
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 15x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908as60acfne

Chapter 24
Keyboard Module (KBI)

Chapter 25
Timer Interface Module A (TIMA)

2.4 Vector Addresses and Priority

Addresses in the range $FFCC to $FFFF contain the user-specified vector locations. The vector addresses are shown in Table 2-1. Please note that certain vector addresses differ between the MC68HC908AS60A and the MC68HC908AZ60A as shown in the table. It is recommended that all vector addresses are defined.

Table 2-1. Vector Addresses

		Vector	
	Address	**MC68HC908AZ60A**	**MC68HC908AS60A**
Lowest Priority	$FFCC	TIMA Channel 5 Vector (High)	Reserved
	$FFCD	TIMA Channel 5 Vector (Low)	Reserved
	$FFCE	TIMA Channel 4 Vector (High)	Reserved
	$FFCF	TIMA Channel 4 Vector (Low)	Reserved
	$FFD0	ADC Vector (High)	Reserved
	$FFD1	ADC Vector (Low)	Reserved
	$FFD2	Keyboard Vector (High)	
	$FFD3	Keyboard Vector (Low)	
	$FFD4	SCI Transmit Vector (High)	Reserved
	$FFD5	SCI Transmit Vector (Low)	Reserved
	$FFD6	SCI Receive Vector (High)	Reserved
	$FFD7	SCI Receive Vector (Low)	Reserved
	$FFD8	SCI Error Vector (High)	Reserved
	$FFD9	SCI Error Vector (Low)	Reserved
	$FFDA	CAN Transmit Vector (High)	PIT Vector (High)
	$FFDB	CAN Transmit Vector (Low)	PIT Vector (Low)
	$FFDC	CAN Receive Vector (High)	BDLC Vector (High)
	$FFDD	CAN Receive Vector (Low)	BDLC Vector (Low)
	$FFDE	CAN Error Vector (High)	ADC Vector (High)
	$FFDF	CAN Error Vector (Low)	ADC Vector (Low)
	$FFE0	CAN Wakeup Vector (High)	SCI Transmit Vector (High)
	$FFE1	CAN Wakeup Vector (Low)	SCI Transmit Vector (Low)
	$FFE2	SPI Transmit Vector (High)	SCI Receive Vector (High)
	$FFE3	SPI Transmit Vector (Low)	SCI Receive Vector (Low)
	$FFE4	SPI Receive Vector (High)	SCI Error Vector (High)
	$FFE5	SPI Receive Vector (Low)	SCI Error Vector (Low)
	$FFE6	TIMB Overflow Vector (High)	SPI Transmit Vector (High)
	$FFE7	TIMB Overflow Vector (Low)	SPI Transmit Vector (Low)
	$FFE8	TIMB CH1 Vector (High)	SPI Receive Vector (High)
	$FFE9	TIMB CH1 Vector (Low)	SPI Receive Vector (Low)
	$FFEA	TIMB CH0 Vector (High)	TIMA Overflow Vector (High)
	$FFEB	TIMB CH0 Vector (Low)	TIMA Overflow Vector (Low)
	$FFEC	TIMA Overflow Vector (High)	TIMA Channel 5 Vector (High)

4.3 FLASH-1 Control and Block Protect Registers

The FLASH-1 array has two registers that control its operation, the FLASH-1 Control Register (FL1CR) and the FLASH-1 Block Protect Register (FL1BPR).

4.3.1 FLASH-1 Control Register

The FLASH-1 Control Register (FL1CR) controls FLASH-1 program and erase operations.

Address: $FF88

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 4-1. FLASH-1 Control Register (FL1CR)

HVEN — High-Voltage Enable Bit

This read/write bit enables the charge pump to drive high voltages for program and erase operations in the array. HVEN can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

1 = High voltage enabled to array and charge pump on
0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

Setting this read/write bit configures the FLASH-1 array for mass or page erase operation.

1 = Mass erase operation selected
0 = Page erase operation selected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be set at the same time.

1 = Erase operation selected
0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Program operation selected
0 = Program operation unselected

4.6 FLASH-1 Page Erase Operation

Use this step-by-step procedure to erase a page (128 bytes) of FLASH-1 memory to read as logic 1:

1. Set the ERASE bit and clear the MASS bit in the FLASH-1 Control Register (FL1CR).
2. Read the FLASH-1 Block Protect Register (FL1BPR).
3. Write any data to any FLASH-1 address within the address range of the page (128 byte block) to be erased.
4. Wait for time, t_{NVS}.
5. Set the HVEN bit.
6. Wait for time, t_{ERASE}.
7. Clear the ERASE bit.
8. Wait for time, t_{NVH}.
9. Clear the HVEN bit.
10. Wait for a time, t_{RCV}, after which the memory can be accessed in normal read mode.

NOTE

A. *Programming and erasing of FLASH locations can not be performed by code being executed from the same FLASH array.*

B. *While these operations must be performed in the order shown, other unrelated operations may occur between the steps. Care must be taken however to ensure that these operations do not access any address within the FLASH array memory space such as the COP Control Register (COPCTL) at $FFFF.*

C. *It is highly recommended that interrupts be disabled during program/erase operations.*

4.7 FLASH-1 Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 64 consecutive bytes with address ranges as follows:

- $XX00 to $XX3F
- $XX40 to $XX7F
- $XX80 to $XXBF
- $XXC0 to $XXFF

During the programming cycle, make sure that all addresses being written to fit within one of the ranges specified above. Attempts to program addresses in different row ranges in one programming cycle will fail.

Use this step-by-step procedure to program a row of FLASH-1 memory.

NOTE

In order to avoid program disturbs, the row must be erased before any byte on that row is programmed.

1. Set the PGM bit in the FLASH-1 Control Register (FL1CR). This configures the memory for program operation and enables the latching of address and data programming.
2. Read the FLASH-1 Block Protect Register (FL1BPR).

6.4.5.3 EEPROM-1 Erasing

The programmed state of an EEPROM bit is logic 0. Erasing changes the state to a logic 1. Only EEPROM-1 bytes in the non-protected blocks and the EE1NVR register can be erased.

Use the following procedure to erase a byte, block or the entire EEPROM-1 array:

1. Configure EERAS1 and EERAS0 for byte, block or bulk erase; set EELAT in EE1CR.(A)

NOTE

If using the AUTO mode, also set the AUTO bit in Step 1.

2. Byte erase: write any data to the desired address.(B)
   Block erase: write any data to an address within the desired block.(B)
   Bulk erase: write any data to an address within the array.(B)
3. Set the EEPGM bit.(C) Go to Step 7 if AUTO is set.
4. Wait for a time: t_{EEBYTE} for byte erase; $t_{EEBLOCK}$ for block erase; $t_{EEBULK.}$ for bulk erase.
5. Clear EEPGM bit.
6. Wait for a time, t_{EEFPV}, for the erasing voltage to fall. Go to Step 8.
7. Poll the EEPGM bit until it is cleared by the internal timer.(D)
8. Clear EELAT bits.(E)

NOTE

A. *Setting the EELAT bit configures the address and data buses to latch data for erasing the array. Only valid EEPROM-1 addresses will be latched. If EELAT is set, other writes to the EE1CR will be allowed after a valid EEPROM-1 write.*

B. *If more than one valid EEPROM write occurs, the last address and data will be latched overriding the previous address and data. Once data is written to the desired address, do not read EEPROM-1 locations other than the written location. (Reading an EEPROM location returns the latched data and causes the read address to be latched).*

C. *The EEPGM bit cannot be set if the EELAT bit is cleared or a non-valid EEPROM address is latched. This is to ensure proper programming sequence. Once EEPGM is set, do not read any EEPROM-1 locations; otherwise, the current program cycle will be unsuccessful. When EEPGM is set, the on-board programming sequence will be activated.*

D. *The delay time for the EEPGM bit to be cleared in AUTO mode is less than t_{EEBYTE}/$t_{EEBLOCK}$/t_{EEBULK}. However, on other MCUs, this delay time may be different. For forward compatibility, software should not make any dependency on this delay time.*

E. *Any attempt to clear both EEPGM and EELAT bits with a single instruction will only clear EEPGM. This is to allow time for removal of high voltage from the EEPROM-1 array.*

Chapter 9
System Integration Module (SIM)

9.1 Introduction

This chapter describes the system integration module (SIM), which supports up to 32 external and/or internal interrupts. Together with the central processor unit (CPU), the SIM controls all MCU activities. A block diagram of the SIM is shown in Figure 9-1. Figure 9-2 is a summary of the SIM input/output (I/O) registers. The SIM is a system state controller that coordinates CPU and exception timing. The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing

NOTE

Software can select the CGMVCLK divided by two as the CGMOUT source even if the PLL is not locked (LOCK = 0). Therefore, software should make sure the PLL is locked before setting the BCS bit.

10.7 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

10.7.1 Wait Mode

The CGM remains active in wait mode. Before entering wait mode, software can disengage and turn off the PLL by clearing the BCS and PLLON bits in the PLL control register (PCTL). Less power-sensitive applications can disengage the PLL without turning it off. Applications that require the PLL to wake the MCU from wait mode also can deselect the PLL output without turning off the PLL.

10.7.2 Stop Mode

The STOP instruction disables the CGM and holds low all CGM outputs (CGMXCLK, CGMOUT, and CGMINT).

If CGMOUT is being driven by CGMVCLK and a STOP instruction is executed; the PLL will clear the BCS bit in the PLL control register, causing CGMOUT to be driven by CGMXCLK. When the MCU recovers from STOP, the crystal clock divided by two drives CGMOUT and BCS remains clear.

10.8 CGM During Break Interrupts

The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See Chapter 13 Break Module (BRK).

To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the PLLF bit during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write the PLL control register during the break state without affecting the PLLF bit.

10.9 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

10.9.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5% acquisition time tolerance. If a command instructs the system to change from 0 Hz to

Chapter 16
Low-Voltage Inhibit (LVI)

16.1 Introduction

This chapter describes the low-voltage inhibit module (LVI47, Version A), which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls to the LVI trip voltage.

16.2 Features

Features of the LVI module include:

- Programmable LVI Reset
- Programmable Power Consumption
- Digital Filtering of V_{DD} Pin Level

NOTE

If a low voltage interrupt (LVI) occurs during programming of EEPROM or Flash memory, then adequate programming time may not have been allowed to ensure the integrity and retention of the data. It is the responsibility of the user to ensure that in the event of an LVI any addresses being programmed receive specification programming conditions.

16.3 Functional Description

Figure 16-1 shows the structure of the LVI module. The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. The LVI power bit, LVIPWR, enables the LVI to monitor V_{DD} voltage. The LVI reset bit, LVIRST, enables the LVI module to generate a reset when V_{DD} falls below a voltage, LVI_{TRIPF}, and remains at or below that level for nine or more consecutive CPU cycles.

Note that short V_{DD} spikes may not trip the LVI. It is the user's responsibility to ensure a clean V_{DD} signal within the specified operating voltage range if normal microcontroller operation is to be guaranteed.

LVISTOP, enables the LVI module during stop mode. This will ensure when the STOP instruction is implemented, the LVI will continue to monitor the voltage level on V_{DD}. LVIPWR, LVISTOP, and LVIRST are in the configuration register, CONFIG-1 (see Chapter 11 Configuration Register (CONFIG-1)).

Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, LVI_{TRIPR}. V_{DD} must be above LVI_{TRIPR} for only one CPU cycle to bring the MCU out of reset (see 16.3.2 Forced Reset Operation). The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR).

An LVI reset also drives the $\overline{RST}$ pin low to provide low-voltage protection to external peripheral devices.

19.5.3 Transmission Format When CPHA = 1

Figure 19-5 shows an SPI transmission in which CPHA (SPCR) is logic 1. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ($\overline{SS}$) is low, so that only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 19.6.2 Mode Fault Error). When CPHA = 1, the master begins driving its MOSI pin on the first SPSCK edge. Therefore, the slave uses the first SPSCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions. This format may be preferable in systems having only one master and only one slave driving the MISO data line.

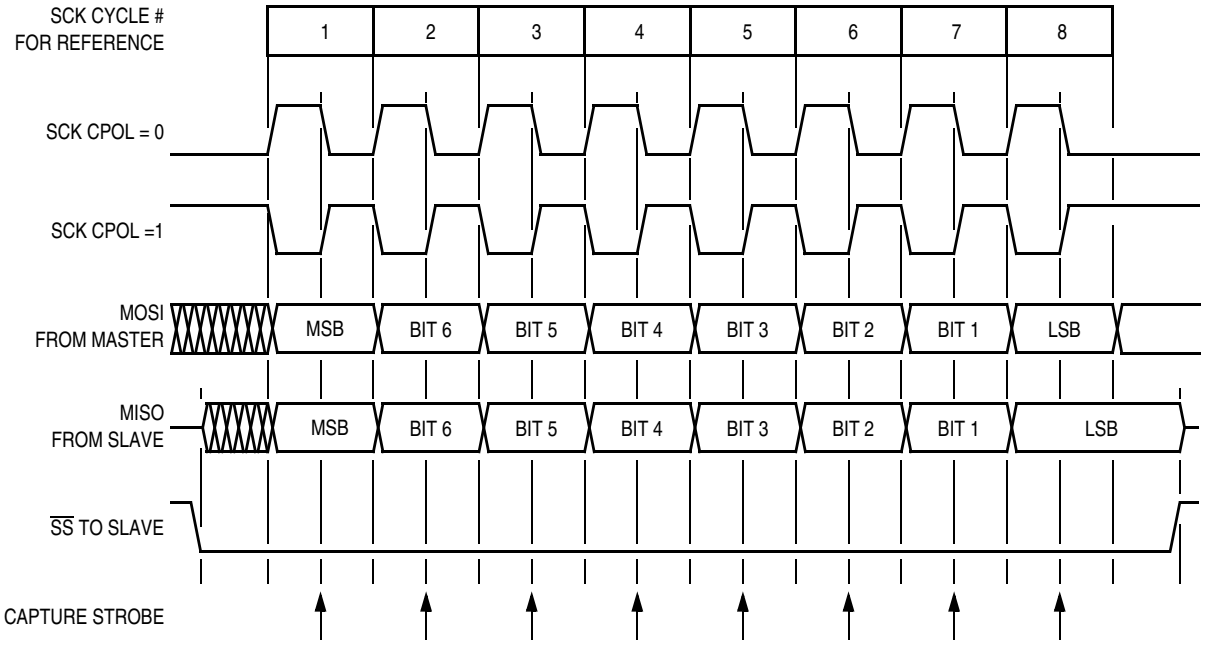


Figure 19-5. Transmission Format (CPHA = 1)

19.5.4 Transmission Initiation Latency

When the SPI is configured as a master (SPMSTR = 1), transmissions are started by a software write to the SPDR ($0012). CPHA has no effect on the delay to the start of the transmission, but it does affect the initial state of the SCK signal. When CPHA = 0, the SCK signal remains inactive for the first half of the first SCK cycle. When CPHA = 1, the first SCK cycle begins with an edge on the SCK line from its inactive to its active level. The SPI clock rate (selected by SPR1–SPR0) affects the delay from the write to SPDR and the start of the SPI transmission. (See Figure 19-6). The internal SPI clock in the master is a free-running derivative of the internal MCU clock. It is only enabled when both the SPE and SPMSTR bits (SPCR) are set to conserve power. SCK edges occur half way through the low time of the internal MCU clock. Since the SPI clock is free-running, it is uncertain where the write to the SPDR will occur relative to the slower SCK. This uncertainty causes the variation in the initiation delay shown in Figure 19-6. This

Chapter 22
Input/Output Ports

22.1 Introduction

On the MC68HC908AZ60A and 64-pin MC68HC908AS60A, fifty bidirectional input/output (I/O) form seven parallel ports. On the52-pin MC68HC908AS60A, forty bidirectional input/output (I/O) form six parallel ports. All I/O pins are programmable as inputs or outputs.

NOTE

Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
$0001	Port B Data Register (PTB)	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
$0002	Port C Data Register (PTC)	0	0	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
$0003	Port D Data Register (PTD)	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
$0004	Data Direction Register A (DDRA)	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
$0005	Data Direction Register B (DDRB)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
$0006	Data Direction Register C (DDRC)	MCLKEN	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
$0007	Data Direction Register D (DDRD)	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
$0008	Port E Data Register (PTE)	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
$0009	Port F Data Register (PTF)	0	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
$000A	Port G Data Register (PTG)	0	0	0	0	0	PTG2	PTG1	PTG0
$000B	Port H Data Register (PTH)	0	0	0	0	0	0	PTH1	PTH0
$000C	Data Direction Register E (DDRE)	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
$000D	Data Direction Register F (DDRF)	0	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
$000E	Data Direction Register G (DDRG)	0	0	0	0	0	DDRG2	DDRG1	DDRG0
$000F	Data Direction Register H (DDRH)	0	0	0	0	0	0	DDRH1	DDRH0

Figure 22-1. I/O Port Register Summary

22.5 Port D

Port D is an 8-bit general-purpose I/O port. Note that PTD7 is only available on 64-pin package options.

22.5.1 Port D Data Register

Port D is a 8-bit special function port that shares seven of its pins with the analog to digital converter and two with the timer interface modules.

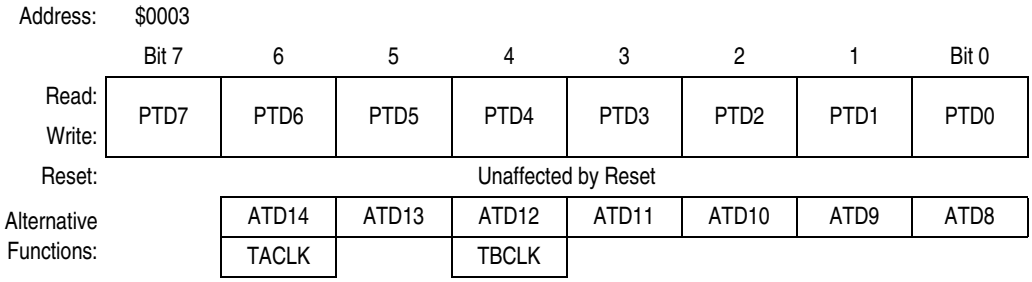


Figure 22-11. Port D Data Register (PTD)

PTD[7:0] — Port D Data Bits

PTD[7:0] are read/write, software programmable bits. Data direction of PTD[7:0] pins are under the control of the corresponding bit in data direction register D.

ATD[14:8] — ADC Channel Status Bits

PTD6/ATD14/TACLK–PTD0/ATD8 are seven of the 15 analog-to-digital converter channels. The ADC channel select bits, CH[4:0], determine whether the PTD6/ATD14/TACLK–PTD0/ATD8 pins are ADC channels or general-purpose I/O pins. If an ADC channel is selected and a read of this corresponding bit in the port B data register occurs, the data will be 0 if the data direction for this bit is programmed as an input. Otherwise, the data will reflect the value in the data latch. (See Chapter 26 Analog-to-Digital Converter (ADC)).

NOTE

Data direction register D (DDRD) does not affect the data direction of port D pins that are being used by the TIMA or TIMB. However, the DDRD bits always determine whether reading port D returns the states of the latches or a 0.

TACLK/TBCLK — Timer Clock Input Bit

The PTD6/ATD14/TACLK pin is the external clock input for the TIMA. The PTD4/ATD12/TBCLK pin is the external clock input for the TIMB. The prescaler select bits, PS[2:0], select PTD6/ATD14/TACLK or PTD4/ATD12/TBCLK as the TIM clock input. (See 25.8.4 TIMA Channel Status and Control Registers and 20.8.4 TIMB Channel Status and Control Registers). When not selected as the TIM clock, PTD6/ATD14/TACLK and PTD4/ATD12/TBCLK are available for general-purpose I/O. While TACLK/TBCLK are selected corresponding DDRD bits have no effect.

TERRIF — Transmitter Error Passive Interrupt Flag

This flag is set when the MSCAN08 goes into error passive status due to the Transmit Error counter exceeding 127 and the Bus-off interrupt flag is not set[1]. If not masked, an Error interrupt is pending while this flag is set.

1 = MSCAN08 went into transmit error passive status.
0 = No transmit error passive status has been reached.

BOFFIF — Bus-Off Interrupt Flag

This flag is set when the MSCAN08 goes into bus-off status, due to the transmit error counter exceeding 255. It cannot be cleared before the MSCAN08 has monitored 128 times 11 consecutive ‘recessive’ bits on the bus. If not masked, an Error interrupt is pending while this flag is set.

1 = MSCAN08has gone into bus-off status.
0 = No bus-off status has bee reached.

OVRIF — Overrun Interrupt Flag

This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set.

1 = A data overrun has been detected since last clearing the flag.
0 = No data overrun has occurred.

RXF — Receive Buffer Full

The RXF flag is set by the MSCAN08 when a new message is available in the foreground receive buffer. This flag indicates whether the buffer is loaded with a correctly received message. After the CPU has read that message from the receive buffer the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the exchange of the background receive buffer into the foreground buffer. If not masked, a receive interrupt is pending while this flag is set.

1 = The receive buffer is full. A new message is available.
0 = The receive buffer is released (not full).

NOTE

To ensure data integrity, no registers of the receive buffer shall be read while the RXF flag is cleared.

NOTE

The CRFLG register is held in the reset state when the SFTRES bit in CMCR0 is set.

1. Condition to set the flag: TERRIF = (128 ≤ TEC ≤ 255) & $\overline{\text{BOFFIF}}$

TOVx — Toggle-On-Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIMA counter overflows. When channel x is an input capture channel, TOVx has no effect. Reset clears the TOVx bit.

1 = Channel x pin toggles on TIMA counter overflow.
0 = Channel x pin does not toggle on TIMA counter overflow.

NOTE

When TOVx is set, a TIMA counter overflow takes precedence over a channel x output compare if both occur at the same time.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at logic 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 25-8 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

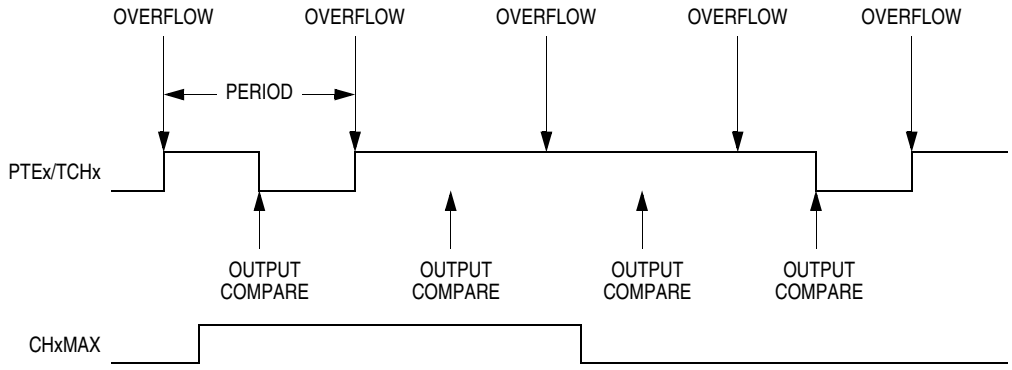


Figure 25-8. CHxMAX Latency

25.8.5 TIMA Channel Registers

These read/write registers contain the captured TIMA counter value of the input capture function or the output compare value of the output compare function. The state of the TIMA channel registers after reset is unknown.

In input capture mode (MSxB–MSxA = 0:0) reading the high byte of the TIMA channel x registers (TACHxH) inhibits input captures until the low byte (TACHxL) is read.

In output compare mode (MSxB–MSxA ≠ 0:0) writing to the high byte of the TIMA channel x registers (TACHxH) inhibits output compares and the CHxF bit until the low byte (TACHxL) is written.

Register Name and Address: TACH0H — $0027

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after Reset							

Figure 25-9. TIMA Channel Registers (TACH0H/L–TACH5H/L) (Sheet 1 of 3)

26.3.2 Voltage Conversion

When the input voltage to the ADC equals V_{REFH} (see 28.1.6 ADC Characteristics), the ADC converts the signal to $FF (full scale). If the input voltage equals $V_{SSA,}$ the ADC converts it to $00. Input voltages between V_{REFH} and V_{SSA} are a straight-line linear conversion. Conversion accuracy of all other input voltages is not guaranteed. Avoid current injection on unused ADC inputs to prevent potential conversion error.

NOTE

Input voltage should not exceed the analog supply voltages.

26.3.3 Conversion Time

Conversion starts after a write to the ADSCR (ADC status control register, $0038), and requires between 16 and 17 ADC clock cycles to complete. Conversion time in terms of the number of bus cycles is a function of ADICLK select, CGMXCLK frequency, bus frequency, and ADIV prescaler bits. For example, with a CGMXCLK frequency of 4 MHz, bus frequency of 8 MHz, and fixed ADC clock frequency of 1 MHz, one conversion will take between 16 and 17 μs and there will be between 128 bus cycles between each conversion. Sample rate is approximately 60 kHz.

Refer to 28.1.6 ADC Characteristics.

$$\text{Conversion Time} = \frac{\text{16 to 17 ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

$$\text{Number of Bus Cycles} = \text{Conversion Time} \times \text{Bus Frequency}$$

26.3.4 Continuous Conversion

In the continuous conversion mode, the ADC data register will be filled with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit (ADC status control register, $0038) is cleared. The COCO bit is set after the first conversion and will stay set for the next several conversions until the next write of the ADC status and control register or the next read of the ADC data register.

26.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes. See 28.1.6 ADC Characteristics for accuracy information.

26.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit (ADC status control register, $0038) is at logic 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

27.4 BDLC MUX Interface

The MUX interface is responsible for bit encoding/decoding and digital noise filtering between the protocol handler and the physical interface.

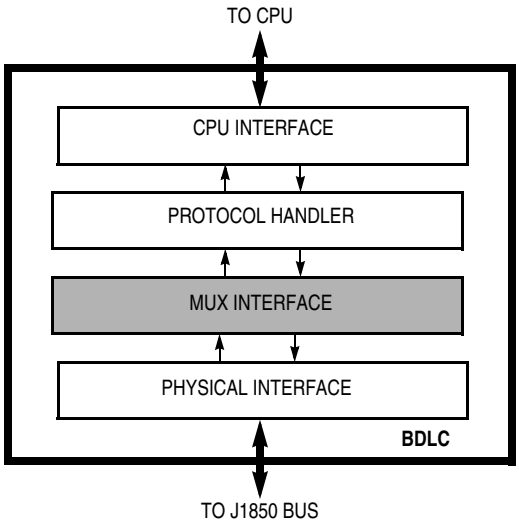


Figure 27-4. BDLC Block Diagram

27.4.1 Rx Digital Filter

The receiver section of the BDLC includes a digital low pass filter to remove narrow noise pulses from the incoming message. An outline of the digital filter is shown in Figure 27-5.

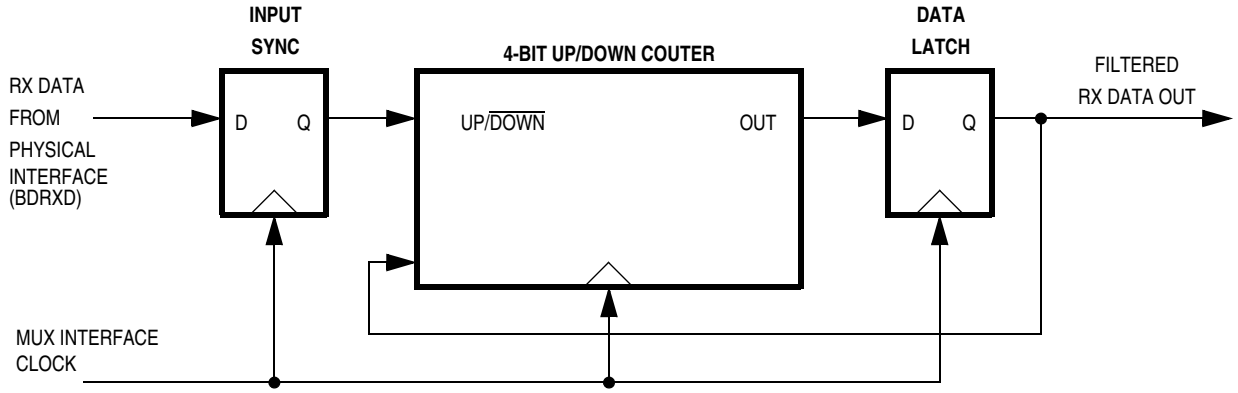


Figure 27-5. BDLC Rx Digital Filter Block Diagram

27.4.1.1 Operation

The clock for the digital filter is provided by the MUX interface clock (see f_{BDLC} parameter in Table 27-3). At each positive edge of the clock signal, the current state of the receiver physical interface (BDRxD) signal is sampled. The BDRxD signal state is used to determine whether the counter should increment or decrement at the next negative edge of the clock signal.

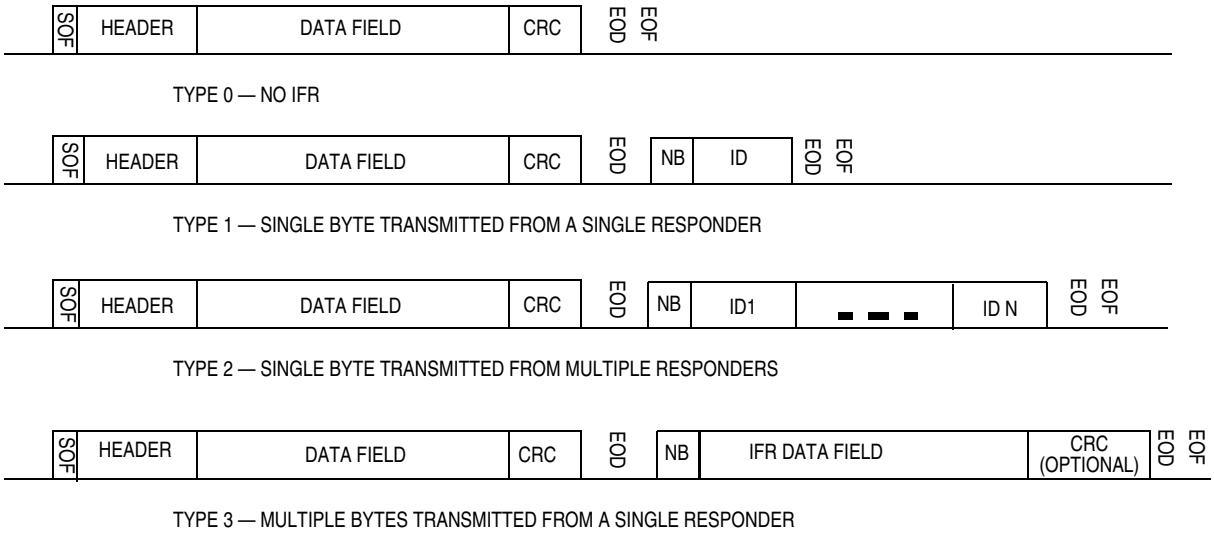
TSIFR, TMIFR1, and TMIFR0 — Transmit In-Frame Response Control Bits

These three bits control the type of in-frame response being sent. The programmer should not set more than one of these control bits to a 1 at any given time. However, if more than one of these three control bits are set to 1, the priority encoding logic will force these register bits to a known value as shown in Table 27-4. For example, if 011 is written to TSIFR, TMIFR1, and TMIFR0, then internally they will be encoded as 010. However, when these bits are read back, they will read 011.

Table 27-4. BDLC Transmit In-Frame Response Control Bit Priority Encoding

Write/Read TSIFR	Write/Read TMIFR1	Write/Read TMIFR0	Actual TSIFR	Actual TMIFR1	Actual TMIFR0
0	0	0	0	0	0
1	X	X	1	0	0
0	1	X	0	1	0
0	0	1	0	0	1

The BDLC supports the in-frame response (IFR) feature of J1850 by setting these bits correctly. The four types of J1850 IFR are shown below. The purpose of the in-frame response modes is to allow multiple nodes to acknowledge receipt of the data by responding with their personal ID or physical address in a concatenated manner after they have seen the EOD symbol. If transmission arbitration is lost by a node while sending its response, it continues to transmit its ID/address until observing its unique byte in the response stream. For VPW modulation, because the first bit of the IFR is always passive, a normalization bit (active) must be generated by the responder and sent prior to its ID/address byte. When there are multiple responders on the J1850 bus, only one normalization bit is sent which assists all other transmitting nodes to sync up their response.



NB = Normalization Bit
ID = Identifier (usually the physical address of the responder(s))
HEADER = Specifies one of three frame lengths

Figure 27-18. Types of In-Frame Response (IFR)

high byte — The most significant eight bits of a word.

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

I — The interrupt mask bit in the condition code register of the CPU08. When I is set, all interrupts are disabled.

index register (H:X) — A 16-bit register in the CPU08. The upper byte of H:X is called H. The lower byte is called X. In the indexed addressing modes, the CPU uses the contents of H:X to determine the effective address of the operand. H:X can also serve as a temporary data storage location.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) and instruction.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

interrupt request — A signal from a peripheral to the CPU intended to cause the CPU to execute a subroutine.

I/O — See “input/output (I/0).”

IRQ — See "external interrupt module (IRQ)."

jitter — Short-term signal instability.

latch — A circuit that retains the voltage level (logic 1 or logic 0) written to it for as long as power is applied to the circuit.

latency — The time lag between instruction completion and data movement.

least significant bit (LSB) — The rightmost digit of a binary number.

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

low byte — The least significant eight bits of a word.

low voltage inhibit module (LVI) — A module in the M68HC08 Family that monitors power supply voltage.

LVI — See "low voltage inhibit module (LVI)."

M68HC08 — A Freescale family of 8-bit MCUs.

mark/space — The logic 1/logic 0 convention used in formatting data in serial communication.

mask — 1. A logic circuit that forces a bit or group of bits to a desired state. 2. A photomask used in integrated circuit fabrication to transfer an image onto silicon.

mask option — A optional microcontroller feature that the customer chooses to enable or disable.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — Mode of low-jitter PLL operation during which the PLL is locked on a frequency. Also see "acquisition mode."

two's complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two's complement number indicates the sign of the number (1 indicates negative). The two's complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value. Executing an opcode at an unimplemented location causes an illegal address reset.

V —The overflow bit in the condition code register of the CPU08. The CPU08 sets the V bit when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow bit.

variable — A value that changes during the course of program execution.

VCO — See "voltage-controlled oscillator."

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

X — The lower byte of the index register (H:X) in the CPU08.

Z — The zero bit in the condition code register of the CPU08. The CPU08 sets the zero bit when an arithmetic operation, logical operation, or data manipulation produces a result of $00.