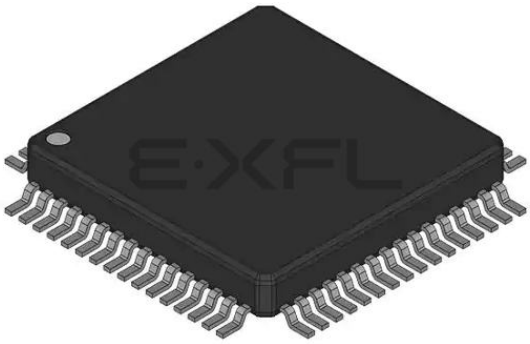


Welcome to E-XFL.COM

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8.4MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	52
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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908az60amfue

Chapter 4
FLASH-1 Memory

Chapter 5
FLASH-2 Memory

Chapter 6
EEPROM-1 Memory

4. Wait for time, t_{NVS}.
5. Set the HVEN bit.
6. Wait for time, t_{PGS}.
7. Write data byte to the FLASH-2 address to be programmed.
8. Wait for time, t_{PROG}.
9. Repeat step 7 and 8 until all the bytes within the row are programmed.
10. Clear the PGM bit.
11. Wait for time, t_{NVH}.
12. Clear the HVEN bit.
13. Wait for a time, t_{RCV}, after which the memory can be accessed in normal read mode.

The FLASH Programming Algorithm Flowchart is shown in Figure 5-4.

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.*

D. *Do not exceed t_{PROG} maximum or t_{HV} maximum. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase. t_{HV} must satisfy this condition: $t_{NVS}+ t_{NVH} + t_{PGS} + (t_{PROG}X\ 64) \leq t_{HV}$ max. Please also see 28.1.14 FLASH Memory Characteristics.*

E. *The time between each FLASH address change (step 7 to step 7), or the time between the last FLASH address programmed to clearing the PGM bit (step 7 to step 10) must not exceed the maximum programming time, t_{PROG} max.*

F. *Be cautious when programming the FLASH-2 array to ensure that non-FLASH locations are not used as the address that is written to when selecting either the desired row address range in step 3 of the algorithm or the byte to be programmed in step 7 of the algorithm. This applies particularly to:*

$0450-$047F: First row of FLASH-2 (48 bytes)

6.4.3 EEPROM-1 Program/Erase Protection

The EEPROM has a special feature that designates the 16 bytes of addresses from $08F0 to $08FF to be permanently secured. This program/erase protect option is enabled by programming the EEPRTCT bit in the EEPROM-1 Nonvolatile Register (EE1NVR) to a logic zero.

Once the EEPRTCT bit is programmed to 0 for the first time:

- Programming and erasing of secured locations $08F0 to $08FF is permanently disabled.
- Secured locations $08F0 to $08FF can be read as normal.
- Programming and erasing of EE1NVR is permanently disabled.
- Bulk and Block Erase operations are disabled for the unprotected locations $0800-$08EF, $0900-$09FF.
- Single byte program and erase operations are still available for locations $0800-$08EF and $0900-$09FF for all bytes that are not protected by the EEPROM-1 Block Protect EEBPx bits (see 6.4.4 EEPROM-1 Block Protection and 6.5.2 EEPROM-1 Array Configuration Register)

NOTE

Once armed, the protect option is permanently enabled. As a consequence, all functions in the EE1NVR will remain in the state they were in immediately before the security was enabled.

6.4.4 EEPROM-1 Block Protection

The 512 bytes of EEPROM-1 are divided into four 128-byte blocks. Each of these blocks can be protected from erase/program operations by setting the EEBPx bit in the EE1NVR. Table 6-1 shows the address ranges for the blocks.

Table 6-1. EEPROM-1 Array Address Blocks

Block Number (EEBPx)	Address Range
EEBP0	$0800–$087F
EEBP1	$0880–$08FF
EEBP2	$0900–$097F
EEBP3	$0980–$09FF

These bits are effective after a reset or a upon read of the EE1NVR register. The block protect configuration can be modified by erasing/programming the corresponding bits in the EE1NVR register and then reading the EE1NVR register. Please see 6.5.2 EEPROM-1 Array Configuration Register for more information.

NOTE

Once EEDIVSECD in the EE1DIVHNVR is programmed to 0 and after a system reset, the EE1DIV security feature is permanently enabled because the EEDIVSECD bit in the EE1DIVH is always loaded with 0 thereafter. Once this security feature is armed, erase and program mode are disabled for EE1DIVHNVR and EE1DIVLNVR. Modifications to the EE1DIVH and EE1DIVL registers are also disabled. Therefore, be cautious on programming a value into the EE1DIVHNVR.

7.4 Functional Description

The 512 bytes of EEPROM-2 are located at $0600-$07FF and can be programmed or erased without an additional external high voltage supply. The program and erase operations are enabled through the use of an internal charge pump. For each byte of EEPROM, the write/erase endurance is 10,000 cycles.

7.4.1 EEPROM-2 Configuration

The 8-bit EEPROM-2 Nonvolatile Register (EE2NVR) and the 16-bit EEPROM-2 Timebase Divider Nonvolatile Register (EE2DIVNVR) contain the default settings for the following EEPROM configurations:

- EEPROM-2 Timebase Reference
- EEPROM-2 Security Option
- EEPROM-2 Block Protection

EE2NVR and EE2DIVNVR are nonvolatile EEPROM registers. They are programmed and erased in the same way as EEPROM bytes. The contents of these registers are loaded into their respective volatile registers during a MCU reset. The values in these read/write volatile registers define the EEPROM-2 configurations.

For EE2NVR, the corresponding volatile register is the EEPROM-2 Array Configuration Register (EE2ACR). For the EE2DIVNCR (two 8-bit registers: EE2DIVHNVR and EE2DIVLNVR), the corresponding volatile register is the EEPROM-2 Divider Register (EE2DIV: EE2DIVH and EE2 DIVL).

7.4.2 EEPROM-2 Timebase Requirements

A 35μs timebase is required by the EEPROM-2 control circuit for program and erase of EEPROM content. This timebase is derived from dividing the CGMXCLK or bus clock (selected by EEDIVCLK bit in CONFIG-2 Register) using a timebase divider circuit controlled by the 16-bit EEPROM-2 Timebase Divider EE2DIV Register (EE2DIVH and EE2DIVL).

As the CGMXCLK or bus clock is user selected, the EEPROM-2 Timebase Divider Register must be configured with the appropriate value to obtain the 35 μs. The timebase divider value is calculated by using the following formula:

$$EE2DIV = INT[Reference\ Frequency(Hz) \times 35 \times 10^{-6} + 0.5]$$

This value is written to the EEPROM-2 Timebase Divider Register (EE2DIVH and EE2DIVL) or programmed into the EEPROM-2 Timebase Divider Nonvolatile Register prior to any EEPROM program or erase operations (7.4.1 EEPROM-2 Configuration and 7.4.2 EEPROM-2 Timebase Requirements).

7.4.5.3 EEPROM-2 Erasing

The programmed state of an EEPROM bit is logic 0. Erasing changes the state to a logic 1. Only EEPROM-2 bytes in the non-protected blocks and the EE2NVR register can be erased.

Use the following procedure to erase a byte, block or the entire EEPROM-2 array:

1. Configure EERAS1 and EERAS0 for byte, block or bulk erase; set EELAT in EE2CR.(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-2 addresses will be latched. If EELAT is set, other writes to the EE2CR will be allowed after a valid EEPROM-2 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-2 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-2 locations; otherwise, the current program cycle will be unsuccessful. When EEPGM is set, the on-board programming sequence will be activated.*

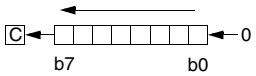
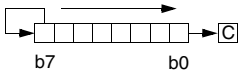
D. *The delay time for the EEPGM bit to be cleared in AUTO mode is less than t_{EEBYTE}/$t_{EEBLOCK}$/t_{EEBULK}. However, on other MCUs, this delay time may be different. For forward compatibility, software should not make any dependency on this delay time.*

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-2 array.*

8.7 Instruction Set Summary

Table 8-1 provides a summary of the M68HC08 instruction set.

Table 8-1. Instruction Set Summary (Sheet 1 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X ADD *opr*,SP ADD *opr*,SP	Add without Carry	A ← (A) + (M)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #*opr*	Add Immediate Value (Signed) to SP	SP ← (SP) + (16 « M)	–	–	–	–	–	–	IMM	A7	ii	2
AIX #*opr*	Add Immediate Value (Signed) to H:X	H:X ← (H:X) + (16 « M)	–	–	–	–	–	–	IMM	AF	ii	2
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X AND *opr*,SP AND *opr*,SP	Logical AND	A ← (A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X ASL *opr*,SP	Arithmetic Shift Left (Same as LSL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR *opr*,X ASR *opr*,SP	Arithmetic Shift Right		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	4 1 1 4 3 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + rel ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BCLR *n*, *opr*	Clear Bit n in M	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? (Z) = 1	–	–	–	–	–	–	REL	27	rr	3
BGE *opr*	Branch if Greater Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) = 0	–	–	–	–	–	–	REL	90	rr	3
BGT *opr*	Branch if Greater Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) ǀ (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	PC ← (PC) + 2 + *rel* ? (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	PC ← (PC) + 2 + *rel* ? (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? (C) ǀ (Z) = 0	–	–	–	–	–	–	REL	22	rr	3

9.6.2 Stop Mode

In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset also causes an exit from stop mode.

The SIM disables the clock generator module outputs (CGMOUT and CGMXCLK) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the configuration register (CONFIG-1). If SSREC is set, stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32. This is ideal for applications using canned oscillators that do not require long startup times from stop mode.

NOTE

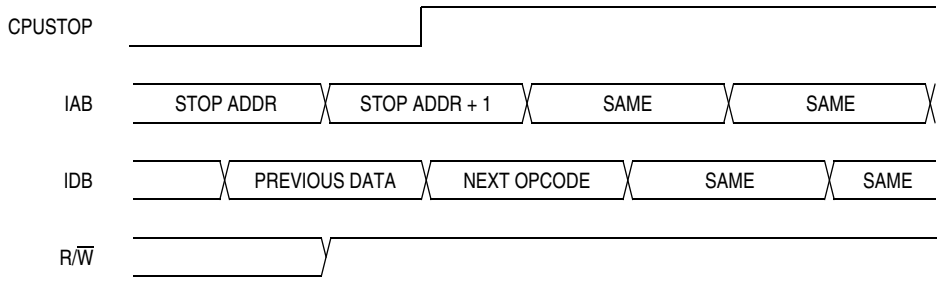
External crystal applications should use the full stop recovery time by clearing the SSREC bit.

The break module is inactive in Stop mode. The STOP instruction does not affect break module register states.

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. Figure 9-15 shows stop mode entry timing.

NOTE

To minimize stop current, all pins configured as inputs should be driven to a logic 1 or logic 0.



NOTE: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 9-15. Stop Mode Entry Timing

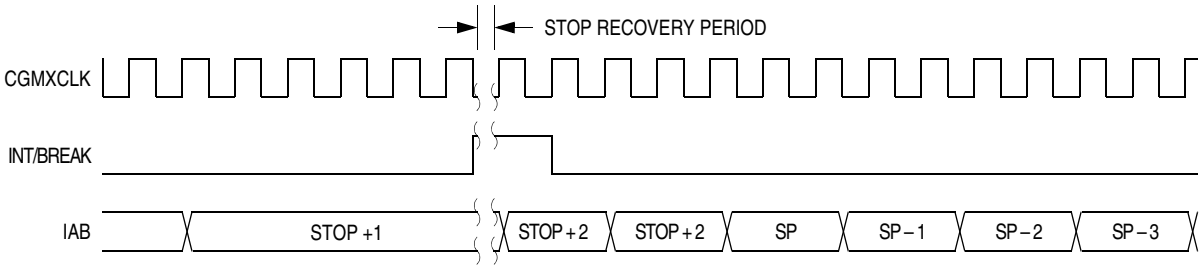


Figure 9-16. Stop Mode Recovery from Interrupt or Break

factor L is programmed to a 0. This value would set up a condition inconsistent with the operation of the PLL, so that the PLL would be disabled and the crystal clock would be forced as the source of the base clock.

10.3.4 CGM External Connections

In its typical configuration, the CGM requires seven external components. Five of these are for the crystal oscillator and two are for the PLL.

The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in Figure 10-3. Figure 10-3 shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

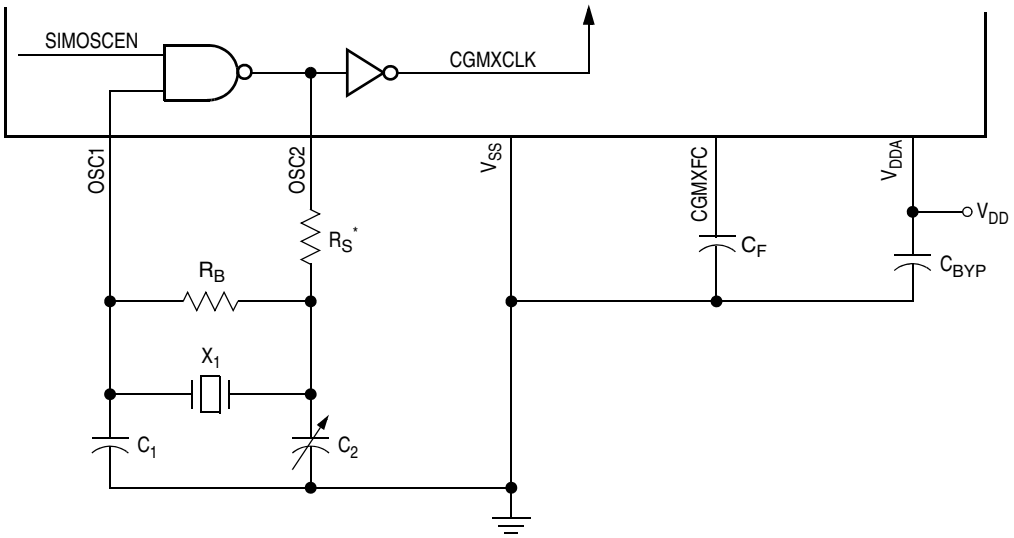
- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S (optional)

The series resistor (R_S) may not be required for all ranges of operation, especially with high-frequency crystals. Refer to the crystal manufacturer's data for more information.

Figure 10-3 also shows the external components for the PLL:

- Bypass capacitor, C_{BYP}
- Filter capacitor, C_F

Routing should be done with great care to minimize signal cross talk and noise. (See 10.9 Acquisition/Lock Time Specifications for routing information and more information on the filter capacitor's value and its effects on PLL performance).



*R_S can be 0 (shorted) when used with higher-frequency crystals. Refer to manufacturer's data.

Figure 10-3. CGM External Connections

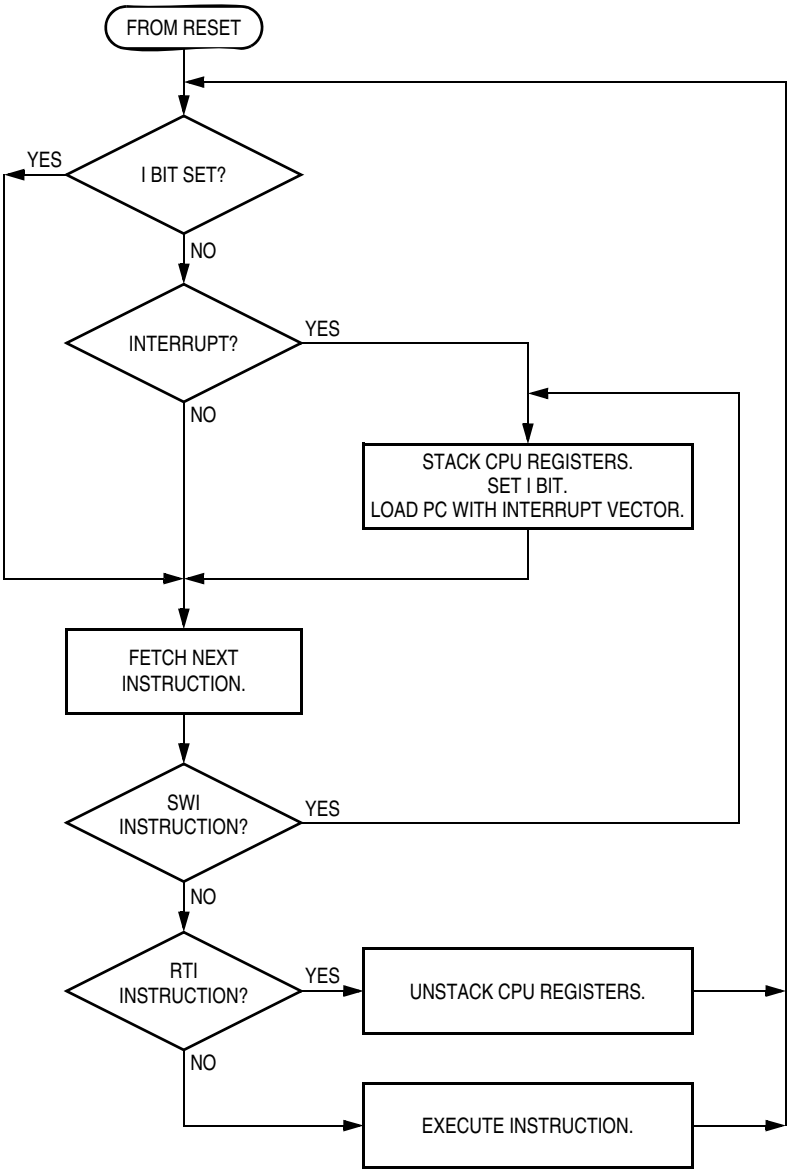


Figure 17-3. IRQ Interrupt Flowchart

The first part of Figure 19-7 shows how to read the SPSCR and SPDR to clear the SPRF without problems. However, as illustrated by the second transmission example, the OVRF flag can be set in between the time that SPSCR and SPDR are read.

In this case, an overflow can be easily missed. Since no more SPRF interrupts can be generated until this OVRF is serviced, it will not be obvious that bytes are being lost as more transmissions are completed. To prevent this, either enable the OVRF interrupt or do another read of the SPSCR after the read of the SPDR. This ensures that the OVRF was not set before the SPRF was cleared and that future transmissions will complete with an SPRF interrupt. Figure 19-8 illustrates this process. Generally, to avoid this second SPSCR read, enable the OVRF to the CPU by setting the ERRIE bit (SPSCR).

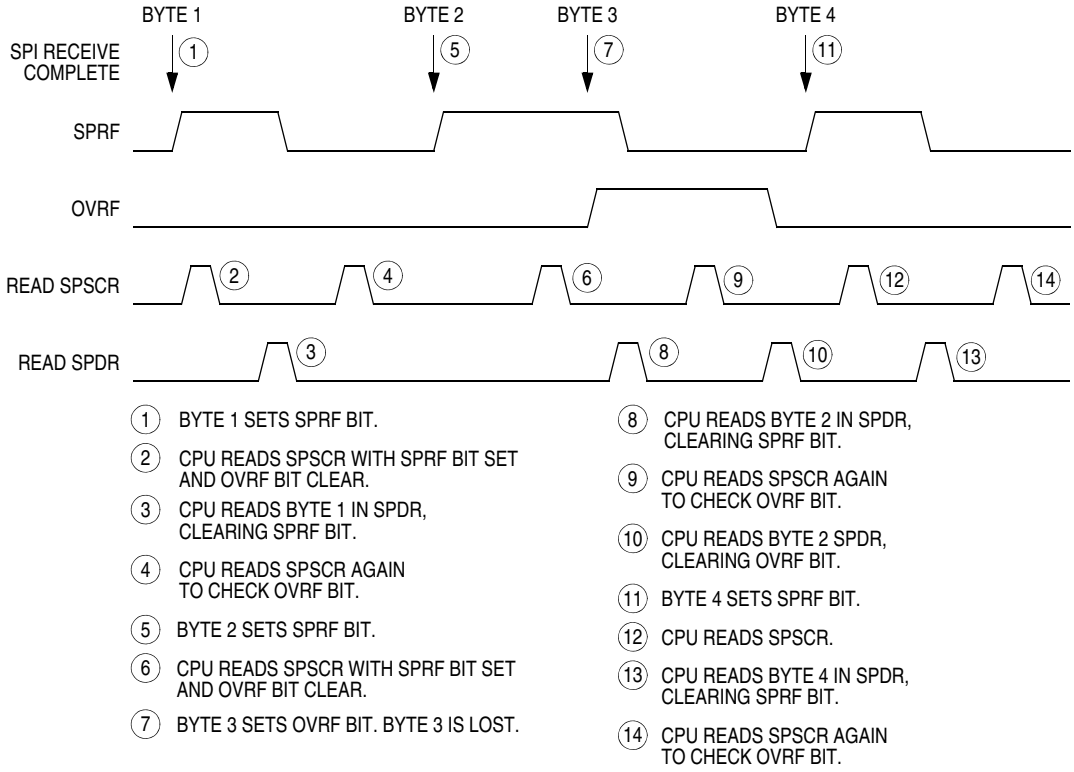


Figure 19-8. Clearing SPRF When OVRF Interrupt Is Not Enabled

19.6.2 Mode Fault Error

For the MODF flag (in SPSCR) to be set, the mode fault error enable bit (MODFEN in SPSCR) must be set. Clearing the MODFEN bit does not clear the MODF flag but does prevent MODF from being set again after MODF is cleared.

MODF generates a receiver/error CPU interrupt request if the error interrupt enable bit (ERRIE in SPSCR) is also set. The SPRF, MODF, and OVRF interrupts share the same CPU interrupt vector. MODF and OVRF can generate a receiver/error CPU interrupt request. (See Figure 19-9). It is not possible to enable only MODF or OVRF to generate a receiver/error CPU interrupt request. However, leaving MODFEN low prevents MODF from being set.

22.8 Port G

Port G is a 3-bit special function port that shares all of its pins with the keyboard interrupt module (KBD). Note that Port G is only available on 64-pin package options.

22.8.1 Port G Data Register

The port G data register contains a data latch for each of the three port G pins.

Address: $000A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	PTG2	PTG1	PTG0
Write:	R	R	R	R	R			
Reset:	Unaffected by Reset							
Alternative Function:						KBD2	KBD1	KBD0

R = Reserved

Figure 22-20. Port G Data Register (PTG)

PTG[2:0] — Port G Data Bits

These read/write bits are software programmable. Data direction of each port G pin is under the control of the corresponding bit in data direction register G. Reset has no effect on PTG[2:0].

KBD[2:0] — Keyboard Wakeup pins

The keyboard interrupt enable bits, KBIE[2:0], in the keyboard interrupt control register, enable the port G pins as external interrupt pins (See Chapter 24 Keyboard Module (KBI)). Enabling an external interrupt pin will override the corresponding DDRGx.

22.8.2 Data Direction Register G

Data direction register G determines whether each port G pin is an input or an output. Writing a logic 1 to a DDRG bit enables the output buffer for the corresponding port G pin; a logic 0 disables the output buffer.

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	DDRG2	DDRG1	DDRG0
Write:	R	R	R	R	R			
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 22-21. Data Direction Register G (DDRG)

22.9 Port H

Port H is a 2-bit special function port that shares all of its pins with the keyboard interrupt module (KBD). Note that Port H is only available on 64-pin package options.

22.9.1 Port H Data Register

The port H data register contains a data latch for each of the two port H pins.

Address: $000B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	PTH1	PTH0
Write:	R	R	R	R	R	R		
Reset:	Unaffected by Reset							
Alternative Function:							KBD4	KBD3

R = Reserved

Figure 22-23. Port H Data Register (PTH)

PTH[1:0] — Port H Data Bits

These read/write bits are software programmable. Data direction of each port H pin is under the control of the corresponding bit in data direction register H. Reset has no effect on PTH[1:0].

KBD[4:3] — Keyboard Wake-up pins

The keyboard interrupt enable bits, KBIE[4:3], in the keyboard interrupt control register, enable the port H pins as external interrupt pins (See Chapter 24 Keyboard Module (KBI)).

22.9.2 Data Direction Register H

Data direction register H determines whether each port H pin is an input or an output. Writing a logic 1 to a DDRH bit enables the output buffer for the corresponding port H pin; a logic 0 disables the output buffer.

Address: $000F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	DDRH1	DDRH0
Write:	R	R	R	R	R	R		
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 22-24. Data Direction Register H (DDRH)

DDRH[1:0] — Data Direction Register H Bits

These read/write bits control port H data direction. Reset clears DDRG[1:0], configuring all port H pins as inputs.

1 = Corresponding port H pin configured as output
0 = Corresponding port H pin configured as input

NOTE

Avoid glitches on port H pins by writing to the port H data register before changing data direction register H bits from 0 to 1.

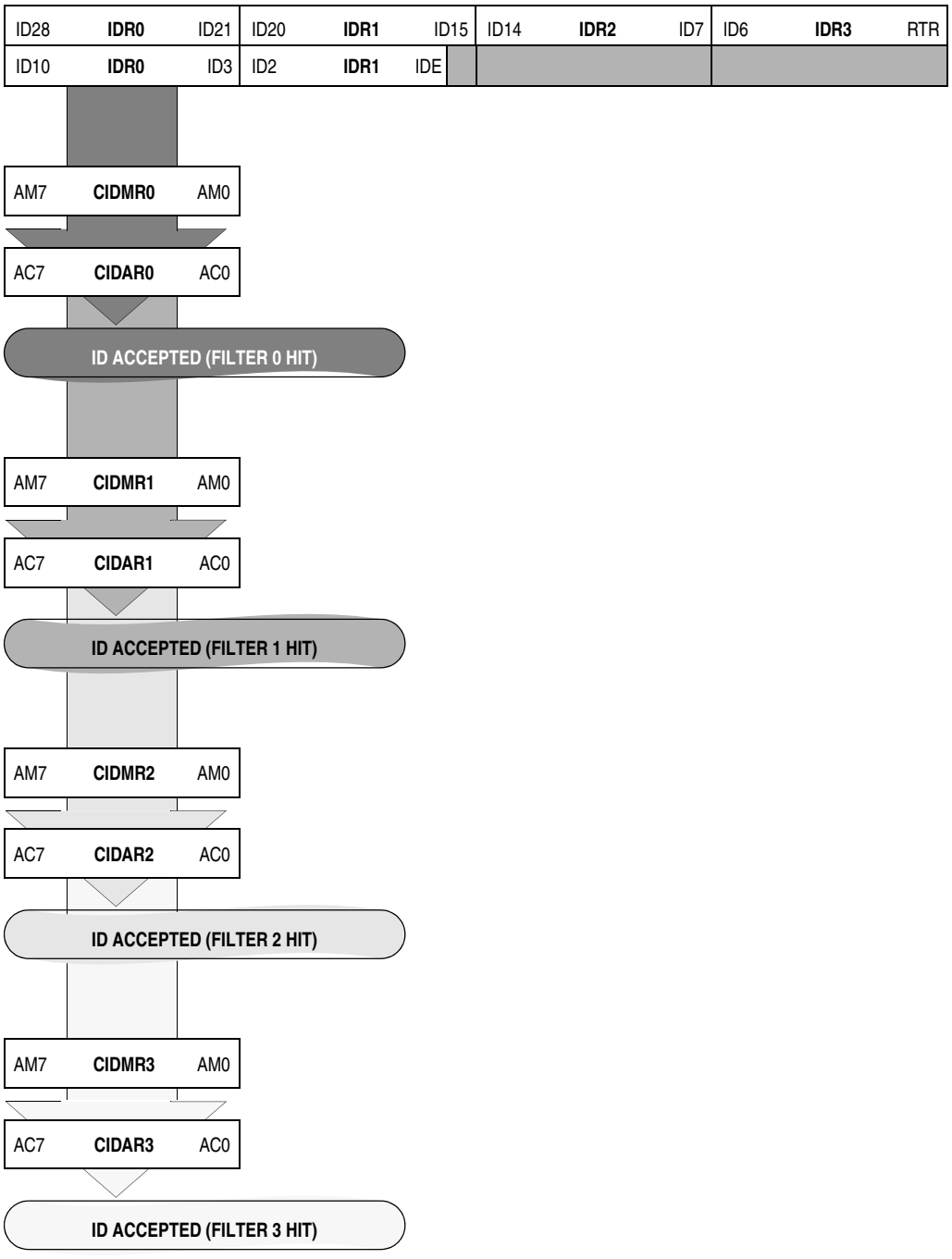


Figure 23-5. Quadruple 8-Bit Maskable Acceptance Filters

23.13.6 MSCAN08 Receiver Interrupt Enable Register

Address: $0505

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	WUPIE	RWRNIE	TWRNIE	RERRIE	TERRIE	BOFFIE	OVRIE	RXFIE
Reset:	0	0	0	0	0	0	0	0

Figure 23-20. Receiver Interrupt Enable Register (CRIER)

WUPIE — Wakeup Interrupt Enable

1 = A wakeup event will result in a wakeup interrupt.
0 = No interrupt will be generated from this event.

RWRNIE — Receiver Warning Interrupt Enable

1 = A receiver warning status event will result in an error interrupt.
0 = No interrupt is generated from this event.

TWRNIE — Transmitter Warning Interrupt Enable

1 = A transmitter warning status event will result in an error interrupt.
0 = No interrupt is generated from this event.

RERRIE — Receiver Error Passive Interrupt Enable

1 = A receiver error passive status event will result in an error interrupt.
0 = No interrupt is generated from this event.

TERRIE — Transmitter Error Passive Interrupt Enable

1 = A transmitter error passive status event will result in an error interrupt.
0 = No interrupt is generated from this event.

BOFFIE — Bus-Off Interrupt Enable

1 = A bus-off event will result in an error interrupt.
0 = No interrupt is generated from this event.

OVRIE — Overrun Interrupt Enable

1 = An overrun event will result in an error interrupt.
0 = No interrupt is generated from this event.

RXFIE — Receiver Full Interrupt Enable

1 = A receive buffer full (successful message reception) event will result in a receive interrupt.
0 = No interrupt will be generated from this event.

NOTE

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

23.13.10 MSCAN08 Receive Error Counter

Address: $050E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 23-24. Receiver Error Counter (CRXERR)

This register reflects the status of the MSCAN08 receive error counter. The register is read only.

23.13.11 MSCAN08 Transmit Error Counter

Address: $050F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 23-25. Transmit Error Counter (CTXERR)

This register reflects the status of the MSCAN08 transmit error counter. The register is read only.

NOTE

Both error counters may only be read when in Sleep or Soft Reset mode.

27.3.1.3 Run Mode

This mode is entered from the reset mode after all MCU reset sources are no longer asserted. Run mode is entered from the BDLC wait mode whenever activity is sensed on the J1850 bus.

Run mode is entered from the BDLC stop mode whenever network activity is sensed, although messages will not be received properly until the clocks have stabilized and the CPU is in run mode also.

In this mode, normal network operation takes place. The user should ensure that all BDLC transmissions have ceased before exiting this mode.

27.3.1.4 BDLC Wait Mode

This power-conserving mode is entered automatically from run mode whenever the CPU executes a WAIT instruction and if the WCM bit in the BCR1 register is cleared previously.

In this mode, the BDLC internal clocks continue to run. The first passive-to-active transition of the bus generates a CPU interrupt request from the BDLC which wakes up the BDLC and the CPU. In addition, if the BDLC receives a valid EOF symbol while operating in wait mode, then the BDLC also will generate a CPU interrupt request which wakes up the BDLC and the CPU. See 27.7.1 Wait Mode.

27.3.1.5 BDLC Stop Mode

This power-conserving mode is entered automatically from run mode whenever the CPU executes a STOP instruction or if the CPU executes a WAIT instruction and the WCM bit in the BCR1 register is set previously.

In this mode, the BDLC internal clocks are stopped but the physical interface circuitry is placed in a low-power mode and awaits network activity. If network activity is sensed, then a CPU interrupt request will be generated, restarting the BDLC internal clocks. See 27.7.2 Stop Mode.

27.3.1.6 Digital Loopback Mode

When a bus fault has been detected, the digital loopback mode is used to determine if the fault condition is caused by failure in the node's internal circuits or elsewhere in the network, including the node's analog physical interface. In this mode, the transmit digital output pin (BDTxD) and the receive digital input pin (BDRxD) of the digital interface are disconnected from the analog physical interface and tied together to allow the digital portion of the BDLC to transmit and receive its own messages without driving the J1850 bus.

27.3.1.7 Analog Loopback Mode

Analog loopback is used to determine if a bus fault has been caused by a failure in the node's off-chip analog transceiver or elsewhere in the network. The BCLD analog loopback mode does not modify the digital transmit or receive functions of the BDLC. It does, however, ensure that once analog loopback mode is exited, the BDLC will wait for an idle bus condition before participation in network communication resumes. If the off-chip analog transceiver has a loopback mode, it usually causes the input to the output drive stage to be looped back into the receiver, allowing the node to receive messages it has transmitted without driving the J1850 bus. In this mode, the output to the J1850 bus is typically high impedance. This allows the communication path through the analog transceiver to be tested without interfering with network activity. Using the BDLC analog loopback mode in conjunction with the analog transceiver's loopback mode ensures that, once the off-chip analog transceiver has exited loopback mode, the BCLD will not begin communicating before a known condition exists on the J1850 bus.

A.1.4 CONFIG-2

CONFIG-2 register $FE09 has 2 new bits activated. Bit 3 is now a silicon hard set bit, which identifies this new A-suffix silicon (1) from the previous non-A suffix silicon (0). Bit 7 is now an EEPROM time base divider clock select bit selecting the reference clock source for the EEPROM time base divider module (refer to EEPROM changes described above).

A.1.5 Keyboard Interrupt

The keyboard module is now a feature of the MC68HC908AS60A in 64-qfp package whereas previously it was only a feature of the AZ device. Vector addresses $FFD2 and $FFD3 are now in the AS memory map in support of this option.

A.1.6 Current Consumption

Current consumption will be significantly lower in many applications. Although maximum specifications are still very dependent upon fabrication process variation and configuration of the MCU in the target application, additional values have been added to the I_{DD} specifications to provide typical current consumption data. Please see Chapter 28 Electrical Specifications for further details.

A.1.7 Illegal Address Reset

Only an opcode fetch from an illegal address will generate an illegal address reset. Data fetches from unmapped addresses will not generate a reset.

A.1.8 Monitor Mode Entry and COP Disable Voltage

The monitor mode entry and COP disable voltage specifications (V_{HI}) have been increased. Please see Chapter 28 Electrical Specifications for details.

A.1.9 Low-Voltage Inhibit (LVI)

The Low-Voltage Inhibit (LVI) specifications for trip and recovery voltage (V_{LVI}) have been altered based upon module performance on silicon. Please see for Chapter 28 Electrical Specifications details.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0012	SPI Data Register (SPDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
$0013	SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
$0014	SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
$0015	SCI Control Register 3 (SCC3)	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
		Write:								
$0016	SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
$0017	SCI Status Register 2 (SCS2)	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
$0018	SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
$0019	SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
$001A	IRQ Status and Control Register (ISCR)	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:					R	ACK		
$001B	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
$001C	PLL Control Register (PCTL)	Read:	PLLIE	PLLF	PLLON	BCS	1	1	1	1
		Write:								
$001D	PLL Bandwidth Control Register (PBWC)	Read:	AUTO	LOCK	$\overline{\text{ACQ}}$	XLD	0	0	0	0
		Write:								
$001E	PLL Programming Register (PPG)	Read:	MUL7	MUL6	MUL5	MUL4	VRS7	VRS6	VRS5	VRS4
		Write:								
$001F	Configuration Write-Once Register (CONFIG-1)	Read:	LVISTOP	R	LVIRST	LVIPWR	SSREC	COPL	STOP	COPD
		Write:								
$0020	Timer A Status and Control Register (TASC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST	R			
$0021	Keyboard Interrupt Enable Register (KBIER)	Read:	0	0	0	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
$0022	Timer A Counter Register High (TACNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0023	Timer A Counter Register Low (TACNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0024	Timer A Modulo Register High (TAMODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								

= Unimplemented R = Reserved

Figure B-2. I/O Data, Status and Control Registers (Sheet 2 of 4)

Revision History

Major Change Between Revision 5.0 and Revision 6.0

The following table lists the major change between the current revision of the MC68HC908AZ60A Technical Data Book, Rev 6.0, and the previous revision, Rev 5.0.

Section affected	Description of change
Appendix B. MC68HC908AZ60E	Added chapter describing the MC68HC908AZ60E.

Major Changes Between Revision 5.0 and Revision 4.0

The following table lists the major changes between the current revision of the MC68HC908AZ60A Technical Data Book, Rev 5.0, and the previous revision, Rev 4.0.

Section affected	Description of change
Throughout	Updated to meet Freescale identity guidelines.
Chapter 24 Keyboard Module (KBI)	Addresses for KBSCR and KBIER registers corrected to $001B and $0021 repectively.
Chapter 28 Electrical Specifications	Updated values for 28.1.8 CGM Operating Conditions.

Major Changes Between Revision 4.0 and Revision 3.0

The following table lists the major changes between the current revision of the MC68HC908AZ60A Technical Data Book, Rev 4.0, and the previous revision, Rev 3.0.

Section affected	Description of change
Electrical Specifications	Updated case outline drawing for 64-Pin Quad Flat Pack (Case 840B)

Major Changes Between Revision 3.0 and Revision 2.0

The following table lists the major changes between the current revision of the MC68HC908AZ60A Technical Data Book, Rev 3.0, and the previous revision, Rev 2.0.

Section affected	Description of change
Keyboard Module (KBD)	In Table 24-1, addresses for KBSCR and KBIER registers corrected to $001B and $0021 repectively In first bullet on page 333, vector addresses corrected to $00D2 and $00D3.