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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	87
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv96mll

7 MCG Specifications

Location:	Table A-12 Page 436

The f_{int_ut} in the rating of the f_{dco_t} should be f_{int_t}. The correct rating should be:

Num	C	Rating		Symbol	Min	Typical	Max	Unit
6	P	DCO output frequency range - trimmed[2]	Low range (DRS=0, DMX32=0) $f_{dco\_t} = 512 \times f_{int\_t}$	f_{dco_t}	16	—	20	MHz
	P		Mid range (DRS=1, DMX32=0) $f_{dco\_t} = 1024 \times f_{int\_t}$		32	—	40	

Chapter 13
Serial Peripheral Interface (S08SPIV3)

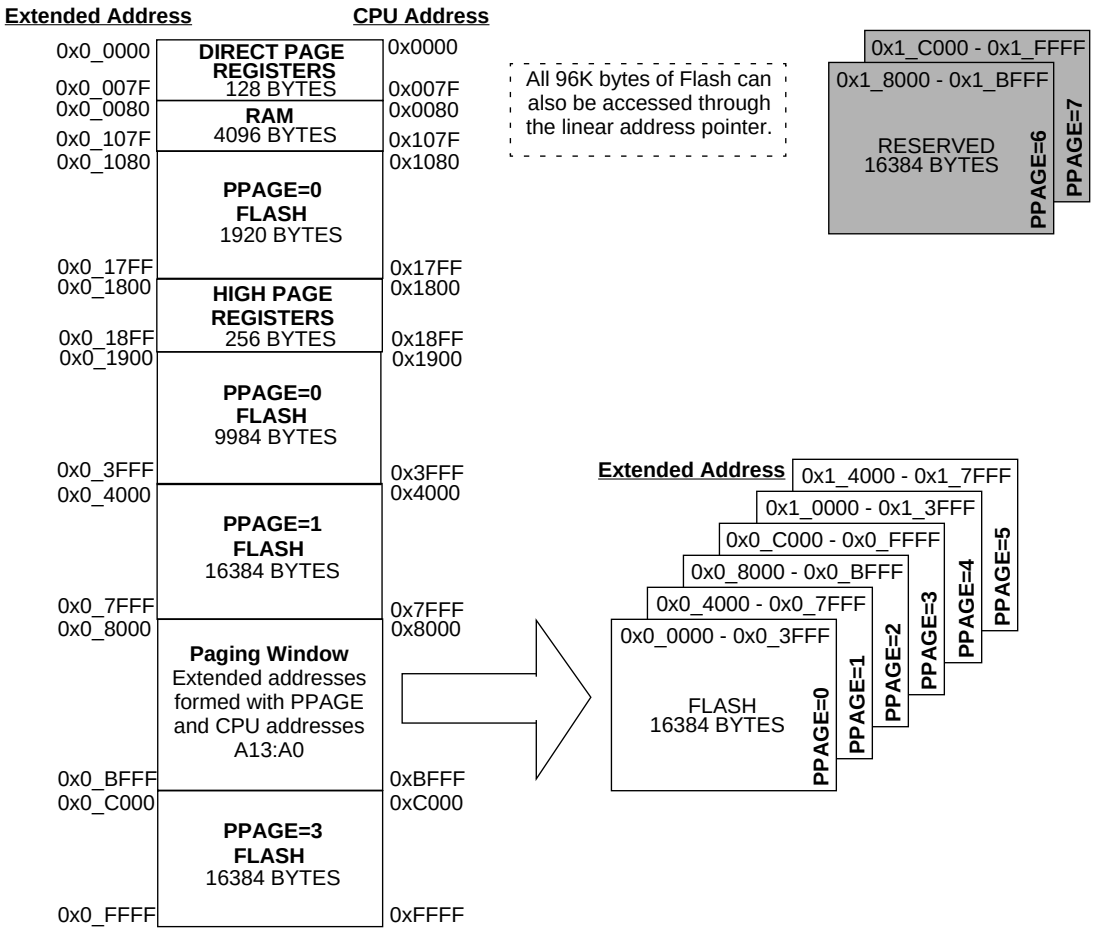

Figure 4-4. MC9S08DV96 Memory Map

4.2 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the MC9S08DZ128 Series equate file provided by Freescale Semiconductor.

Table 4-1. Reset and Interrupt Vectors

Address (High:Low)	Vector	Vector Name
0xFF80:0xFF81 - 0xFF8E:0xFF8F	Reserved	Reserved
0xFF90:0xFF91	Port J	Vportj
0xFF92:0xFF93	IIC2	Viic2
0xFF94:0xFF95	SPI2	Vspi2
0xFF96:0xFF97	TPM3 Overflow	Vtpm3ovf
0xFF98:0xFF99	TPM3 Channel 3	Vtpm3ch3
0xFF9A:0xFF9B	TPM3 Channel 2	Vtpm3ch2

Accessing either the LBP or LWP registers allows a user program to read successive memory locations without re-writing the linear address pointer. Accessing LBP or LWP does the exact same function. However, because of the address mapping of the registers with LBP following LWP, a user can do word accesses in the extended address space using the LDHX or STHX instructions to access location LWP.

The MMU supports the addition of a 2s complement value to the linear address pointer without using any math instructions or memory resources. Writes to LAPAB with a 2s complement value will cause the MMU to add that value to the existing value in LAP2:LAP0.

4.4.3 MMU Registers and Control Bits

4.4.3.1 Program Page Register (PPAGE)

The HCS08 Core architecture limits the CPU addressable space available to 64K bytes. The address space can be extended to 128K bytes using a paging window scheme. The Program Page (PPAGE) allows for selecting one of the 16K byte blocks to be accessed through the Program Page Window located at 0x8000-0xBFFF. The CALL and RTC instructions can load or store the value of PPAGE onto or from the stack during program execution. After any reset, PPAGE is set to PAGE 2.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	XA16	XA15	XA14
W								
Reset:	0	0	0	0	0	0	1	0

Figure 4-5. Program Page Register (PPAGE)

Table 4-6. Program Page Register Field Descriptions

Field	Description
2:0 XA16:XA14	When the CPU addresses the paging window, 0x8000-0xBFFF, the value in the PPAGE register along with the CPU addresses A13:A0 are used to create a 17-bit extended address.

4.4.3.2 Linear Address Pointer Registers 2:0 (LAP2:LAP0)

The three registers, LAP2:LAP0 contain the 17-bit linear address that allows the user to access any FLASH location in the extended address map. This register is used in conjunction with the data registers, linear byte (LB), linear byte post increment (LBP) and linear word post increment (LWP). The contents of LAP2:LAP0 will auto-increment when accessing data using the LBP and LWP registers. The contents of LAP2:LAP0 can be increased by writing an 8-bit value to LAPAB.

Table 4-13. FCDIV Register Field Descriptions

Field	Description
7 DIVLD	**Divisor Loaded Status Flag** — When set, this read-only status flag indicates that the FCDIV register has been written since reset. Reset clears this bit and the first write to this register causes this bit to become set regardless of the data written. 0 FCDIV has not been written since reset; erase and program operations disabled for FLASH and EEPROM. 1 FCDIV has been written since reset; erase and program operations enabled for FLASH and EEPROM.
6 PRDIV8	**Prescale (Divide) FLASH and EEPROM Clock by 8** (This bit is write once.) 0 Clock input to the FLASH and EEPROM clock divider is the bus rate clock. 1 Clock input to the FLASH and EEPROM clock divider is the bus rate clock divided by 8.
5:0 DIV	**Divisor for FLASH and EEPROM Clock Divider** — The FLASH and EEPROM clock divider divides the bus rate clock (or the bus rate clock divided by 8 if PRDIV8 = 1) by the value in the 6-bit DIV field plus one. The resulting frequency of the internal FLASH and EEPROM clock must fall within the range of 200 kHz to 150 kHz for proper FLASH and EEPROM operations. Program/Erase timing pulses are one cycle of this internal FLASH and EEPROM clock which corresponds to a range of 5 µs to 6.7 µs. The automated programming logic uses an integer number of these pulses to complete an erase or program operation. See Equation 4-1 and Equation 4-2.

$$\text{if PRDIV8} = 0 \text{ — } f_{FCLK} = f_{Bus} \div (\text{DIV} + 1) \qquad \textit{Eqn. 4-1}$$

$$\text{if PRDIV8} = 1 \text{ — } f_{FCLK} = f_{Bus} \div (8 \times (\text{DIV} + 1)) \qquad \textit{Eqn. 4-2}$$

Table 4-14 shows the appropriate values for PRDIV8 and DIV for selected bus frequencies.

Table 4-14. FLASH and EEPROM Clock Divider Settings

f_{Bus}	PRDIV8 (Binary)	DIV (Decimal)	f_{FCLK}	Program/Erase Timing Pulse (5 µs Min, 6.7 µs Max)
20 MHz	1	12	192.3 kHz	5.2 µs
10 MHz	0	49	200 kHz	5 µs
8 MHz	0	39	200 kHz	5 µs
4 MHz	0	19	200 kHz	5 µs
2 MHz	0	9	200 kHz	5 µs
1 MHz	0	4	200 kHz	5 µs
200 kHz	0	0	200 kHz	5 µs
150 kHz	0	0	150 kHz	6.7 µs

4.6.11.2 FLASH and EEPROM Options Register (FOPT and NVOPT)

During reset, the contents of the nonvolatile location NVOPT are copied from FLASH into FOPT. To change the value in this register, erase and reprogram the NVOPT location in FLASH memory as usual and then issue a new MCU reset.

	7	6	5	4	3	2	1	0
R	KEYEN	FNORED	EPGMOD	0	0	0	SEC	
W								
Reset	F	F	F	0	0	0	F	F

= Unimplemented or Reserved

Figure 4-15. FLASH and EEPROM Options Register (FOPT)

Table 4-15. FOPT Register Field Descriptions

Field	Description
7 KEYEN	**Backdoor Key Mechanism Enable** — When this bit is 0, the backdoor key mechanism cannot be used to disengage security. The backdoor key mechanism is accessible only from user (secured) firmware. BDM commands cannot be used to write key comparison values that would unlock the backdoor key. For more detailed information about the backdoor key mechanism, refer to Section 4.6.9, “Security.” 0 No backdoor key access allowed. 1 If user firmware writes an 8-byte value that matches the nonvolatile backdoor key (NVBACKKEY through NVBACKKEY+7 in that order), security is temporarily disengaged until the next MCU reset.
6 FNORED	**Vector Redirection Disable** — When this bit is 1, vector redirection is disabled. 0 Vector redirection enabled. 1 Vector redirection disabled.

Table 4-15. FOPT Register Field Descriptions

Field	Description
5 EPGMOD	**EEPROM Sector Mode** — When this bit is 0, each sector is split into two pages (4-byte mode). When this bit is 1, each sector is in a single page (8-byte mode). 0 Half of each EEPROM sector is in Page 0 and the other half is in Page 1. 1 Each sector is in a single page.
1:0 SEC	**Security State Code** — This 2-bit field determines the security state of the MCU as shown in Table 4-16. When the MCU is secure, the contents of RAM, EEPROM and FLASH memory cannot be accessed by instructions from any unsecured source including the background debug interface. SEC changes to 1:0 after successful backdoor key entry or a successful blank check of FLASH. For more detailed information about security, refer to Section 4.6.9, “Security.”

Table 4-16. Security States[1]

SEC[1:0]	Description
0:0	secure
0:1	secure
1:0	unsecured
1:1	secure

1 SEC changes to 1:0 after successful backdoor key entry or a successful blank check of FLASH.

4.6.11.3 FLASH and EEPROM Configuration Register (FCNFG)

	7	6	5	4	3	2	1	0
R	0	EPGSEL	KEYACC	1	0	0	0	1
W								
Reset	0	0	0	1	0	0	0	1

= Unimplemented or Reserved

Figure 4-16. FLASH and EEPROM Configuration Register (FCNFG)

Table 4-17. FCNFG Register Field Descriptions

Field	Description
6 EPGSEL	**EEPROM Page Select** — This bit selects which EEPROM page is accessed in the memory map. 0 Page 0 is in foreground of memory map. Page 1 is in background and can not be accessed. 1 Page 1 is in foreground of memory map. Page 0 is in background and can not be accessed.
5 KEYACC	**Enable Writing of Access Key** — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to Section 4.6.9, “Security.” 0 Writes to 0xFFB0–0xFFB7 are interpreted as the start of a FLASH programming or erase command. 1 Writes to NVBACKKEY (0xFFB0–0xFFB7) are interpreted as comparison key writes.

5.8.7 System Power Management Status and Control 1 Register (SPMSC1)

This high page register contains status and control bits to support the low-voltage detect function, and to enable the bandage voltage reference for use by the ADC and ACMP modules. This register should be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

	7	6	5	4	3	2	1	0
R	LVWF[1]	0	LVWIE	LVDRE[2]	LVDSE	LVDE[2]	0	BGBE
W		LVWACK						
Reset:	0	0	0	1	1	1	0	0

= Unimplemented or Reserved

1 LVWF will be set in the case when V_{Supply} transitions below the trip point or after reset and V_{Supply} is already below V_{LVW}.

2 This bit can be written only one time after reset. Additional writes are ignored.

Figure 5-9. System Power Management Status and Control 1 Register (SPMSC1)

Table 5-10. SPMSC1 Register Field Descriptions

Field	Description
7 LVWF	**Low-Voltage Warning Flag** — The LVWF bit indicates the low-voltage warning status. 0 low-voltage warning is not present. 1 low-voltage warning is present or was present.
6 LVWACK	**Low-Voltage Warning Acknowledge** — If LVWF = 1, a low-voltage condition has occurred. To acknowledge this low-voltage warning, write 1 to LVWACK, which will automatically clear LVWF to 0 if the low-voltage warning is no longer present.
5 LVWIE	**Low-Voltage Warning Interrupt Enable** — This bit enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when LVWF = 1.
4 LVDRE	**Low-Voltage Detect Reset Enable** — This write-once bit enables LVD events to generate a hardware reset (provided LVDE = 1). 0 LVD events do not generate hardware resets. 1 Force an MCU reset when an enabled low-voltage detect event occurs.
3 LVDSE	**Low-Voltage Detect Stop Enable** — Provided LVDE = 1, this read/write bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode. 1 Low-voltage detect enabled during stop mode.
2 LVDE	**Low-Voltage Detect Enable** — This write-once bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled. 1 LVD logic enabled.
0 BGBE	**Bandgap Buffer Enable** — This bit enables an internal buffer for the bandgap voltage reference for use by the ADC and ACMP modules on one of its internal channels. 0 Bandgap buffer disabled. 1 Bandgap buffer enabled.

6.5.11 Port L Registers

Port L is controlled by the registers listed below.

6.5.11.1 Port L Data Register (PTLD)

	7	6	5	4	3	2	1	0
R W	PTLD7	PTLD6	PTLD5	PTLD4	PTLD3	PTLD2	PTLD1	PTLD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-65. Port L Data Register (PTLD)

Table 6-63. PTLD Register Field Descriptions

Field	Description
7:0 PTLD[7:0]	**Port L Data Register Bits** — For port L pins that are inputs, reads return the logic level on the pin. For port L pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port L pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTLD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pull-ups disabled.

6.5.11.2 Port L Data Direction Register (PTLDD)

	7	6	5	4	3	2	1	0
R W	PTLDD7	PTLDD6	PTLDD5	PTLDD4	PTLDD3	PTLDD2	PTLDD1	PTLDD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-66. Port L Data Direction Register (PTLDD)

Table 6-64. PTLDD Register Field Descriptions

Field	Description
7:0 PTLDD[7:0]	**Data Direction for Port L Bits** — These read/write bits control the direction of port L pins and what is read for PTLD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port L bit n and PTLD reads return the contents of PTLDn.

8.1.1 Features

Key features of the MCG module are:

- Frequency-locked loop (FLL)
  - Internal or external reference can be used to control the FLL
- Phase-locked loop (PLL)
  - Voltage-controlled oscillator (VCO)
  - Modulo VCO frequency divider
  - Phase/Frequency detector
  - Integrated loop filter
  - Lock detector with interrupt capability
- Internal reference clock
  - Nine trim bits for accuracy
  - Can be selected as the clock source for the MCU
- External reference clock
  - Control for external oscillator
  - Clock monitor with reset capability
  - Can be selected as the clock source for the MCU
- Reference divider is provided
- Clock source selected can be divided down by 1, 2, 4, or 8
- BDC clock (MCGLCLK) is provided as a constant divide by 2 of the DCO output whether in an FLL or PLL mode.
- Two selectable digitally controlled oscillators (DCOs) optimized for different frequency ranges.
- Option to maximize DCO output frequency for a 32,768 Hz external reference clock source.

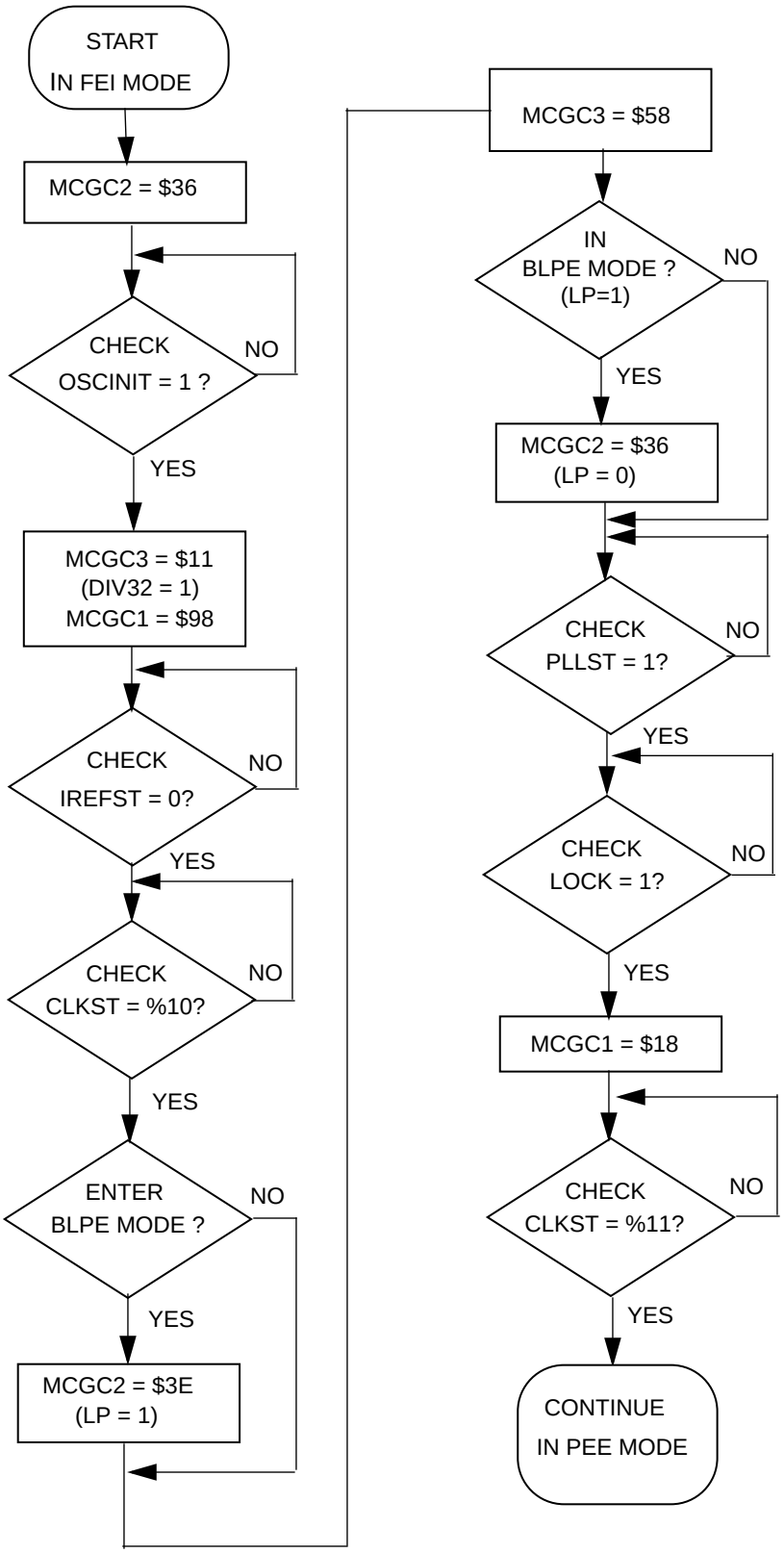

Figure 8-10. Flowchart of FEI to PEE Mode Transition using an 8 MHz crystal

11.1.2 Features

The IIC includes these distinctive features:

- Compatible with IIC bus standard
- Multi-master operation
- Software programmable for one of 64 different serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated start signal generation
- Acknowledge bit generation/detection
- Bus busy detection
- General call recognition
- 10-bit address extension

11.1.3 Modes of Operation

A brief description of the IIC in the various MCU modes is given here.

- **Run mode** — This is the basic mode of operation. To conserve power in this mode, disable the module.
- **Wait mode** — The module continues to operate while the MCU is in wait mode and can provide a wake-up interrupt.
- **Stop mode** — The IIC is inactive in stop3 mode for reduced power consumption. The stop instruction does not affect IIC register states. Stop2 resets the register contents.

11.3.5 IIC Data I/O Register (IICxD)

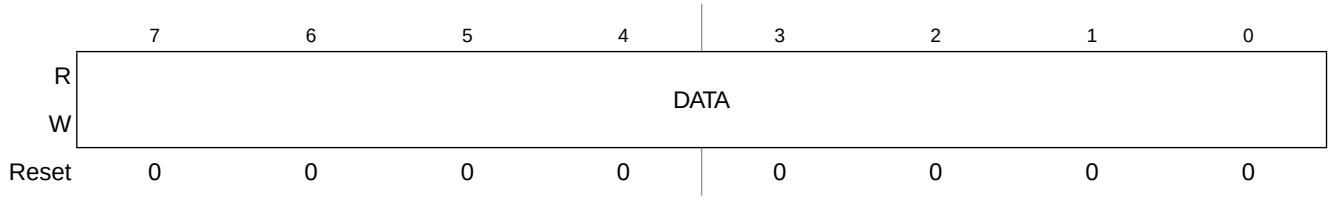

Figure 11-7. IIC Data I/O Register (IICxD)

Table 11-8. IICxD Field Descriptions

Field	Description
7–0 DATA	**Data** — In master transmit mode, when data is written to the IICxD, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data.

NOTE

When transitioning out of master receive mode, the IIC mode should be switched before reading the IICxD register to prevent an inadvertent initiation of a master receive data transfer.

In slave mode, the same functions are available after an address match has occurred.

The TX bit in IICxC must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the IIC is configured for master transmit but a master receive is desired, reading the IICxD does not initiate the receive.

Reading the IICxD returns the last byte received while the IIC is configured in master receive or slave receive modes. The IICxD does not reflect every byte transmitted on the IIC bus, nor can software verify that a byte has been written to the IICxD correctly by reading it back.

In master transmit mode, the first byte of data written to IICxD following assertion of MST is used for the address transfer and should comprise of the calling address (in bit 7 to bit 1) concatenated with the required R/$\overline{W}$ bit (in position bit 0).

11.3.6 IIC Control Register 2 (IICxC2)

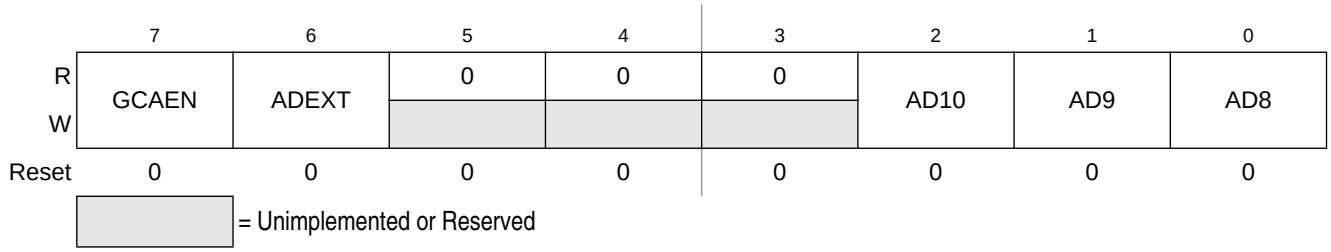

Figure 11-8. IIC Control Register (IICxC2)

Section 12.3.10, "MSCAN Transmit Buffer Selection Register (CANTBSEL)"). For receive buffers, only when RXF flag is set (see Section 12.3.4.1, "MSCAN Receiver Flag Register (CANRFLG)").

Write: For transmit buffers, anytime when TXEx flag is set (see Section 12.3.6, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 12.3.10, "MSCAN Transmit Buffer Selection Register (CANTBSEL)"). Unimplemented for receive buffers.

Reset: Undefined (0x00XX) because of RAM-based implementation

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
IDR0	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
	W								
IDR1	R	ID2	ID1	ID0	RTR[1]	IDE[2]			
	W								
IDR2	R								
	W								
IDR3	R								
	W								

= Unused, always read 'x'

Figure 12-24. Receive/Transmit Message Buffer — Standard Identifier Mapping

1 The position of RTR differs between extended and standard indentifier mapping.

2 IDE is 0.

12.4.1 Identifier Registers (IDR0–IDR3)

The identifier registers for an extended format identifier consist of a total of 32 bits; ID[28:0], SRR, IDE, and RTR bits. The identifier registers for a standard format identifier consist of a total of 13 bits; ID[10:0], RTR, and IDE bits.

12.4.1.1 IDR0–IDR3 for Extended Identifier Mapping

	7	6	5	4	3	2	1	0
R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
W								
Reset:	x	x	x	x	x	x	x	x

Figure 12-25. Identifier Register 0 (IDR0) — Extended Identifier Mapping

in LSBFE. Both variations of SPSCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output pin from a master and the MISO waveform applies to the MISO output from a slave. The $\overline{SS}$ OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master $\overline{SS}$ output goes to active low at the start of the first bit time of the transfer and goes back high one-half SPSCK cycle after the end of the eighth bit time of the transfer. The $\overline{SS}$ IN waveform applies to the slave select input of a slave.

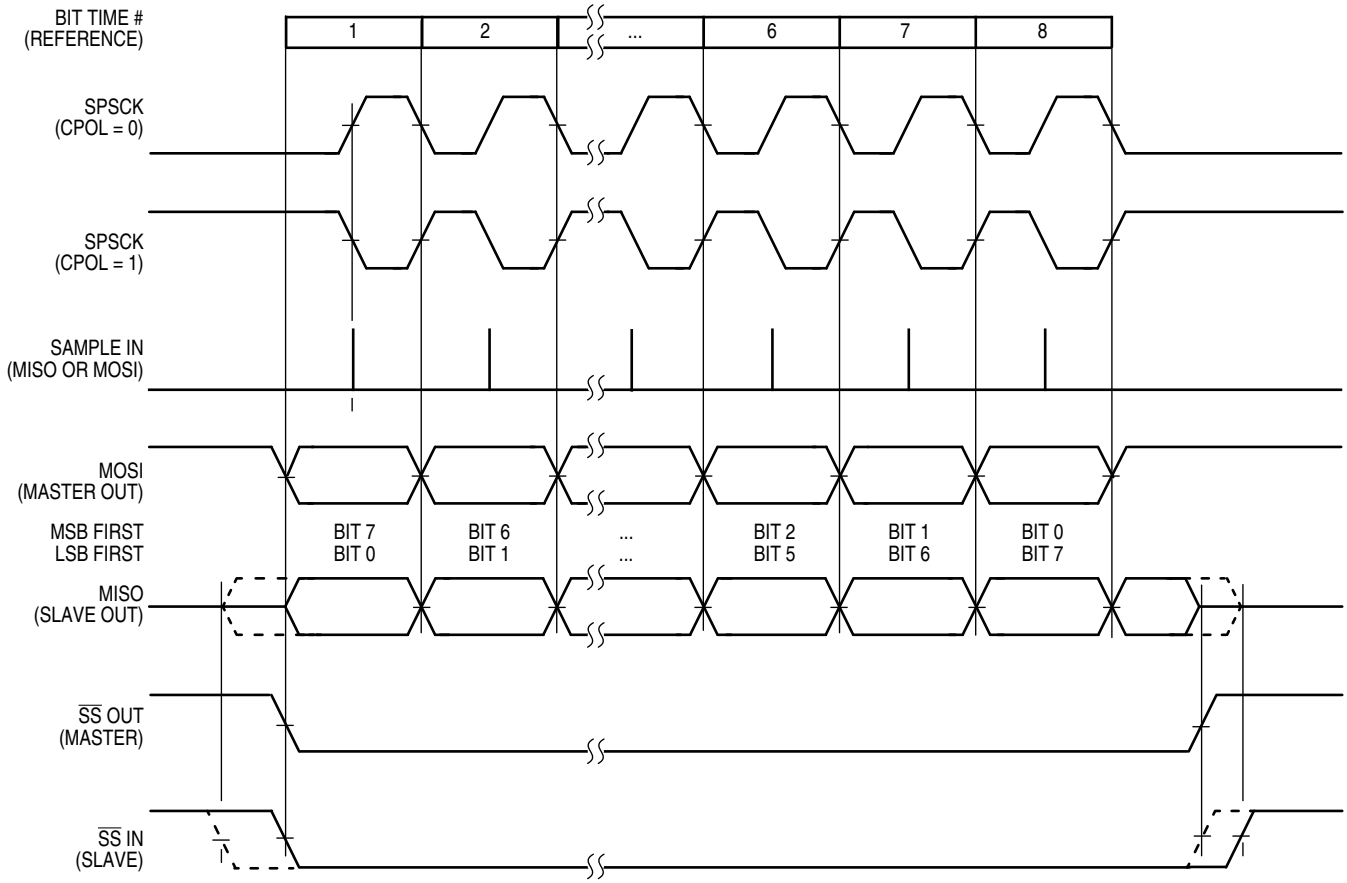

Figure 13-11. SPI Clock Formats (CPHA = 0)

When CPHA = 0, the slave begins to drive its MISO output with the first data bit value (MSB or LSB depending on LSBFE) when $\overline{SS}$ goes to active low. The first SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the second SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CPHA = 0, the slave's $\overline{SS}$ input must go to its inactive high level between transfers.

Writing 0 to TE does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity that is in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

14.3.2.1 Send Break and Queued Idle

The SBK control bit in SCIxC2 is used to send break characters which were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (10 bit times including the start and stop bits). A longer break of 13 bit times can be enabled by setting BRK13 = 1. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another Freescale Semiconductor SCI, the break characters will be received as 0s in all eight data bits and a framing error (FE = 1) occurs.

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while TE = 0, set the general-purpose I/O controls so the pin that is shared with TxD is an output driving a logic 1. This ensures that the TxD line will look like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE.

The length of the break character is affected by the BRK13 and M bits as shown below.

Table 14-9. Break Character Length

BRK13	M	Break Character Length
0	0	10 bit times
0	1	11 bit times
1	0	13 bit times
1	1	14 bit times

14.3.3 Receiver Functional Description

In this section, the receiver block diagram (Figure 14-3) is used as a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wakeup function are explained.

The receiver input is inverted by setting RXINV = 1. The receiver is enabled by setting the RE bit in SCIxC2. Character frames consist of a start bit of logic 0, eight (or nine) data bits (LSB first), and a stop bit of logic 1. For information about 9-bit data mode, refer to Section 14.3.5.1, “8- and 9-Bit Data Modes.” For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status

16.2.1.1 EXTCLK — External Clock Source

Control bits in the timer status and control register allow the user to select nothing (timer disable), the bus-rate clock (the normal default source), a crystal-related clock, or an external clock as the clock which drives the TPM prescaler and subsequently the 16-bit TPM counter. The external clock source is synchronized in the TPM. The bus clock clocks the synchronizer; the frequency of the external source must be no more than one-fourth the frequency of the bus-rate clock, to meet Nyquist criteria and allowing for jitter.

The external clock signal shares the same pin as a channel I/O pin, so the channel pin will not be usable for channel I/O function when selected as the external clock source. It is the user's responsibility to avoid such settings. If this pin is used as an external clock source (CLKSB:CLKSA = 1:1), the channel can still be used in output compare mode as a software timer (ELSnB:ELSnA = 0:0).

16.2.1.2 TPMxCHn — TPM Channel n I/O Pin(s)

Each TPM channel is associated with an I/O pin on the MCU. The function of this pin depends on the channel configuration. The TPM pins share with general purpose I/O pins, where each pin has a port data register bit, and a data direction control bit, and the port has optional passive pullups which may be enabled whenever a port pin is acting as an input.

The TPM channel does not control the I/O pin when (ELSnB:ELSnA = 0:0) or when (CLKSB:CLKSA = 0:0) so it normally reverts to general purpose I/O control. When CPWMS = 1 (and ELSnB:ELSnA not = 0:0), all channels within the TPM are configured for center-aligned PWM and the TPMxCHn pins are all controlled by the TPM system. When CPWMS=0, the MSnB:MSnA control bits determine whether the channel is configured for input capture, output compare, or edge-aligned PWM.

When a channel is configured for input capture (CPWMS=0, MSnB:MSnA = 0:0 and ELSnB:ELSnA not = 0:0), the TPMxCHn pin is forced to act as an edge-sensitive input to the TPM. ELSnB:ELSnA control bits determine what polarity edge or edges will trigger input-capture events. A synchronizer based on the bus clock is used to synchronize input edges to the bus clock. This implies the minimum pulse width—that can be reliably detected—on an input capture pin is four bus clock periods (with ideal clock pulses as near as two bus clocks can be detected). TPM uses this pin as an input capture input to override the port data and data direction controls for the same pin.

When a channel is configured for output compare (CPWMS=0, MSnB:MSnA = 0:1 and ELSnB:ELSnA not = 0:0), the associated data direction control is overridden, the TPMxCHn pin is considered an output controlled by the TPM, and the ELSnB:ELSnA control bits determine how the pin is controlled. The remaining three combinations of ELSnB:ELSnA determine whether the TPMxCHn pin is toggled, cleared, or set each time the 16-bit channel value register matches the timer counter.

When the output compare toggle mode is initially selected, the previous value on the pin is driven out until the next output compare event—then the pin is toggled.

18.3.3.9 Debug Comparator A Extension Register (DBGCAX)

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R	RWAEN	RWA	PAGSEL	0	0	0	0	Bit 16
W								
POR or non-end-run	0	0	0	0	0	0	0	0
Reset end-run[1]	U	U	U	0	0	0	0	U

= Unimplemented or Reserved

Figure 18-10. Debug Comparator A Extension Register (DBGCAX)

1 In the case of an end-trace to reset where DBGEN=1 and BEGIN=0, the bits in this register do not change after reset.

Table 18-11. DBGCAX Field Descriptions

Field	Description
7 RWAEN	**Read/Write Comparator A Enable Bit** — The RWAEN bit controls whether read or write comparison is enabled for Comparator A. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
6 RWA	**Read/Write Comparator A Value Bit** — The RWA bit controls whether read or write is used in compare for Comparator A. The RWA bit is not used if RWAEN = 0. 0 Write cycle will be matched 1 Read cycle will be matched
5 PAGSEL	**Comparator A Page Select Bit** — This PAGSEL bit controls whether Comparator A will be qualified with the internal signal (mmu_ppage_sel) that indicates an extended access through the PPAGE mechanism. When mmu_ppage_sel = 1, the 17-bit core address is a paged program access, and the 17-bit core address is made up of PPAGE[2:0]:addr[13:0]. When mmu_ppage_sel = 0, the 17-bit core address is either a 16-bit CPU address with a leading 0 in bit 16, or a 17-bit linear address pointer value. 0 Match qualified by mmu_ppage_sel = 0 so address bits [16:0] correspond to a 17-bit CPU address with a leading zero at bit 16, or a 17-bit linear address pointer address 1 Match qualified by mmu_ppage_sel = 1 so address bits [16:0] compare to flash memory address made up of PPAGE[2:0]:addr[13:0]
0 Bit 16	**Comparator A Extended Address Bit 16 Compare Bit** — The Comparator A bit 16 compare bit controls whether Comparator A will compare the core address bus bit 16 to a logic 1 or logic 0. 0 Compare corresponding address bit to a logic 0 1 Compare corresponding address bit to a logic 1