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	HCS12
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, EBI/EMI, I²C, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	59
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12hz64caa

1.5.6.19 PM2 / RXCAN0 — Port M I/O Pin 2

PM2 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN0 of the scalable controller area network controller 0 (CAN0)

1.5.6.20 PP5 / PWM5 — Port P I/O Pin 5

PP5 is a general-purpose input or output pin. It can be configured as pulse width modulator (PWM) channel output PWM5 or the serial clock pin SCL of the inter-IC bus interface (IIC).

1.5.6.21 PP4 / PWM4 — Port P I/O Pin 4

PP4 is a general-purpose input or output pin. It can be configured as pulse width modulator (PWM) channel output PWM4 or the serial data pin SDA of the inter-IC bus interface (IIC).

1.5.6.22 PP3 / PWM3 — Port P I/O Pin 3

PP3 is a general-purpose input or output pin. It can be configured as pulse width modulator (PWM) channel output PWM3.

1.5.6.23 PP2 / PWM2 — Port P I/O Pin 2

PP2 is a general-purpose input or output pin. It can be configured as pulse width modulator (PWM) channel output PWM2 or the receive pin RXD1 of the serial communication interface 1 (SCI1).

1.5.6.24 PP1 / PWM1 — Port P I/O Pin 1

PP1 is a general-purpose input or output pin. It can be configured as pulse width modulator (PWM) channel output PWM1.

1.5.6.25 PP0 / PWM0 — Port P I/O Pin 0

PP0 is a general-purpose input or output pin. It can be configured as pulse width modulator (PWM) channel output PWM0 or the transmit pin TXD1 of the serial communication interface 1 (SCI1).

1.5.6.26 PS7 / $\overline{SS}$ — Port S I/O Pin 7

PS7 is a general-purpose input or output pin. It can be configured as slave select pin $\overline{SS}$ of the serial peripheral interface (SPI).

1.5.6.27 PS6 / SCK — Port S I/O Pin 6

PS6 is a general-purpose input or output pin. It can be configured as serial clock pin SCK of the serial peripheral interface (SPI).

2.3.1 Module Memory Map

The Flash memory map is shown in Figure 2-2. The HCS12 architecture places the Flash memory addresses between 0x4000 and 0xFFFF which corresponds to three 16-Kbyte pages. The content of the HCS12 core PPAGE register is used to map the logical middle page ranging from address 0x8000 to 0xBFFF to any physical 16 Kbyte page in the Flash memory. By placing 0x3E or 0x3F in the HCS12 Core PPAGE register, the associated 16 Kbyte pages appear twice in the MCU memory map.

The FPROT register, described in Section 2.3.2.5, “Flash Protection Register (FPROT)”, can be set to globally protect a Flash block. However, three separate memory regions, one growing upward from the first address in the next-to-last page in the Flash block (called the lower region), one growing downward from the last address in the last page in the Flash block (called the higher region), and the remaining addresses in the Flash block, can be activated for protection. The Flash locations of these protectable regions are shown in Table 2-2. The higher address region of Flash block 0 is mainly targeted to hold the boot loader code because it covers the vector space. The lower address region of any Flash block can be used for EEPROM emulation in an MCU without an EEPROM module because it can remain unprotected while the remaining addresses are protected from program or erase.

Security information that allows the MCU to restrict access to the Flash module is stored in the Flash configuration field found in Flash block 0, described in Table 2-1.

Table 2-1. Flash Configuration Field

Unpaged Flash Address	Paged Flash Address (PPAGE 0x3F)	Size (Bytes)	Description
0xFF00 – 0xFF07	0xBF00 – 0xBF07	8	Backdoor Comparison Key Refer to Section 2.6.1, “Unsecuring the MCU using Backdoor Key Access”
0xFF08 – 0xFF0B	0xBF08 – 0xBF0B	4	Reserved
0xFF0C	0xBF0C	1	Block 1 Flash Protection Byte Refer to Section 2.3.2.7, “Flash Status Register (FSTAT)”
0xFF0D	0xBF0D	1	Block 0 Flash Protection Byte Refer toSection 2.3.2.7, “Flash Status Register (FSTAT)”
0xFF0E	0xBF0E	1	Flash Nonvolatile Byte Refer to Section 2.3.2.9, “Flash Control Register (FCTL)”
0xFF0F	0xBF0F	1	Flash Security Byte Refer to Section 2.3.2.2, “Flash Security Register (FSEC)”

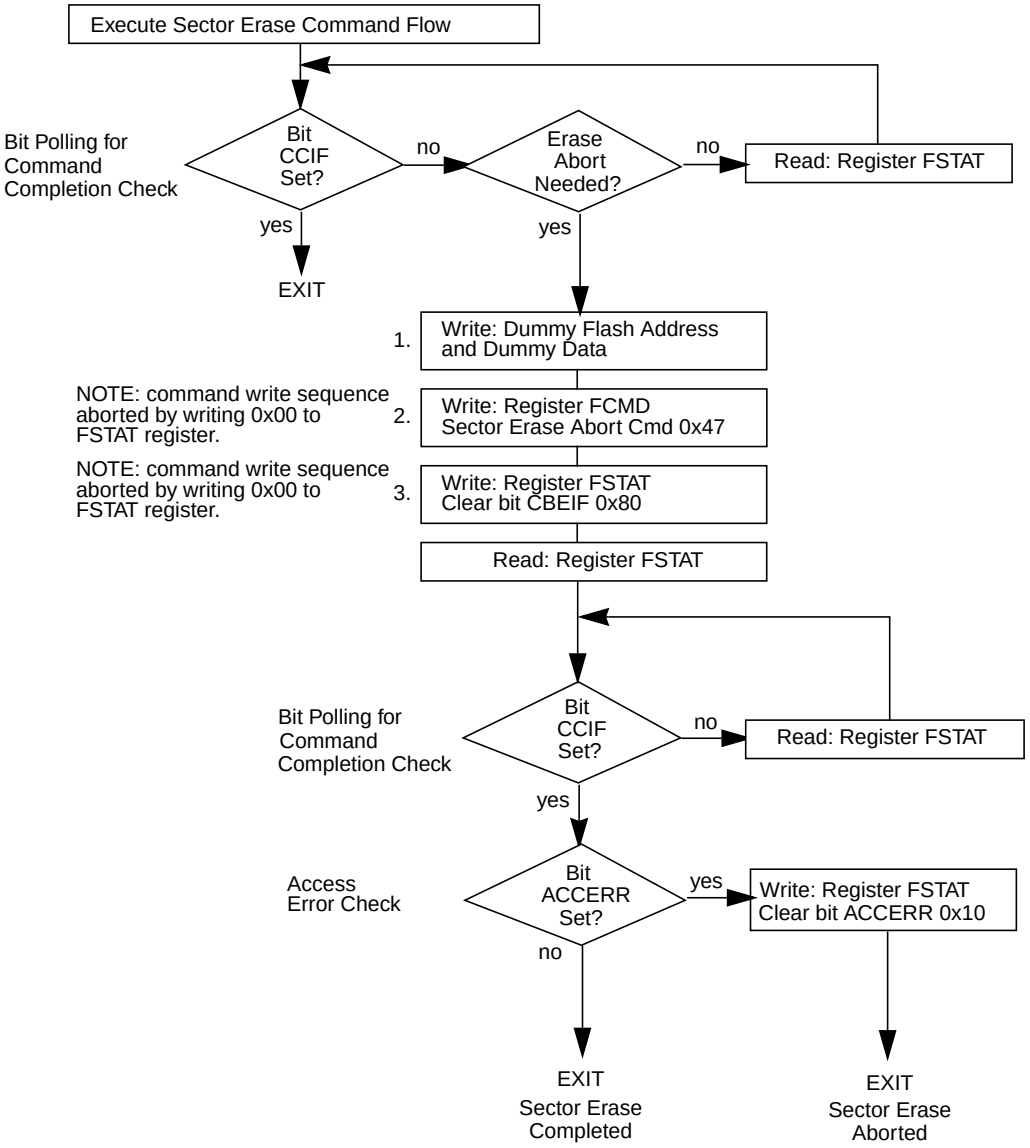

Figure 2-28. Example Sector Erase Abort Command Flow

To change the EEPROM protection that will be loaded on reset, the upper sector of EEPROM must first be unprotected, then the EEPROM protect byte located at address 0x07FD must be written to.

A protected EEPROM sector is disabled by the EPDIS bit while the size of the protected sector is defined by the EP bits in the EPROT register.

Trying to alter any of the protected areas will result in a protect violation error and PVIOL flag will be set in the ESTAT register. A mass erase of a whole EEPROM block is only possible when protection is fully disabled by setting the EPOPEN and EPDIS bits. An attempt to mass erase an EEPROM block while protection is enabled will set the PVIOL flag in the ESTAT register.

Table 3-5. EPROT Field Descriptions

Field	Description
7 EPOPEN	**Opens EEPROM for Program or Erase** 0 The whole EEPROM array is protected. In this case, the EPDIS and EP bits within the protection register are ignored. 1 The EEPROM sectors not protected are enabled for program or erase.
6:4 NV[6:4]	**Nonvolatile Flag Bits** — These three bits are available to the user as nonvolatile flags.
3 EPDIS	**EEPROM Protection Address Range Disable** — The EPDIS bit determines whether there is a protected area in the space of the EEPROM address map. 0 Protection enabled 1 Protection disabled
2:0 EP[2:0]	**EEPROM Protection Address Size** — The EP[2:0] bits determine the size of the protected sector. Refer to Table 3-6.

Table 3-6. EEPROM Address Range Protection

EP[2:0]	Protected Address Range	Protected Size
000	0x07C0-0x07FF	64 bytes
001	0x0780-0x07FF	128 bytes
010	0x0740-0x07FF	192 bytes
011	0x0700-0x07FF	256 bytes
100	0x06C0-0x07FF	320 bytes
101	0x0680-0x07FF	384 bytes
110	0x0640-0x07FF	448 bytes
111	0x0600-0x07FF	512 bytes

3.3.2.6 EEPROM Status Register (ESTAT)

The ESTAT register defines the EEPROM state machine command status and EEPROM array access, protection and erase verify status.

4.3.3.5 Port M Pull Device Enable Register (PERM)

	7	6	5	4	3	2	1	0
R	0	0	PERM5	PERM4	PERM3	PERM2	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Reserved or Unimplemented

Figure 4-20. Port M Pull Device Enable Register (PERM)

Read: Anytime. Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated on configured input or wired-or output pins. If a pin is configured as push-pull output, the corresponding Pull Device Enable Register bit has no effect.

Table 4-15. PERM Field Descriptions

Field	Description
5:2 PERM[5:2]	**Pull Device Enable Port M** 0 Pull-up or pull-down device is disabled. 1 Pull-up or pull-down device is enabled.

4.3.3.6 Port M Polarity Select Register (PPSM)

	7	6	5	4	3	2	1	0
R	0	0	PPSM5	PPSM4	PPSM3	PPSM2	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Reserved or Unimplemented

Figure 4-21. Port M Polarity Select Register (PPSM)

Read: Anytime. Write: Anytime.

The Port M Polarity Select Register selects whether a pull-down or a pull-up device is connected to the pin. The Port M Polarity Select Register is effective only when the corresponding Data Direction Register bit is set to 0 (input) and the corresponding Pull Device Enable Register bit is set to 1.

If a CAN module is enabled, a pull-up device can be activated on the receiver pin, and on the transmitter pin if the corresponding wired-OR mode bit is set. Pull-down devices can not be activated on CAN pins.

of 4 OSCCLK plus 4 PLLCLK cycles to make the transition. During the transition, all clocks freeze and CPU activity ceases.

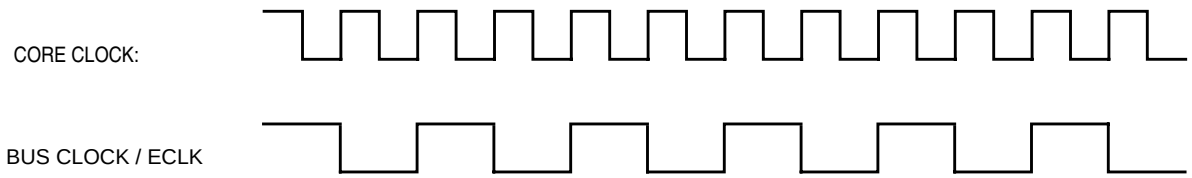

Figure 5-18. Core Clock and Bus Clock Relationship

5.4.3 Clock Monitor (CM)

If no OSCCLK edges are detected within a certain time, the clock monitor within the oscillator block generates a clock monitor fail event. The CRG then asserts self-clock mode or generates a system reset depending on the state of SCME bit. If the clock monitor is disabled or the presence of clocks is detected no failure is indicated by the oscillator block.The clock monitor function is enabled/disabled by the CME control bit.

5.4.4 Clock Quality Checker

The clock monitor performs a coarse check on the incoming clock signal. The clock quality checker provides a more accurate check in addition to the clock monitor.

A clock quality check is triggered by any of the following events:

- Power-on reset (POR)
- Low voltage reset (LVR)
- Wake-up from full stop mode (exit full stop)
- Clock monitor fail indication (CM fail)

A time window of 50000 VCO clock cycles[1] is called *check window*.

A number greater equal than 4096 rising OSCCLK edges within a *check window* is called *osc ok*. Note that *osc ok* immediately terminates the current *check window*. See Figure 5-19 as an example.

1. VCO clock cycles are generated by the PLL when running at minimum frequency f_{SCM}.

7.3.2.6 ATD Control Register 5 (ATDCTL5)

This register selects the type of conversion sequence and the analog input channels sampled. Writes to this register will abort current conversion sequence and start a new conversion sequence. If external trigger is enabled (ETRIGE = 1) an initial write to ATDCTL5 is required to allow starting of a conversion sequence which will then occur on each trigger event. Start of conversion means the beginning of the sampling phase.

	7	6	5	4	3	2	1	0
R W	DJM	DSGN	SCAN	MULT	CD	CC	CB	CA
Reset	0	0	0	0	0	0	0	0

Figure 7-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

Table 7-14. ATDCTL5 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — This bit controls justification of conversion data in the result registers. See Section 7.3.2.16, “ATD Conversion Result Registers (ATDDRx)” for details. 0 Left justified data in the result registers. 1 Right justified data in the result registers.
6 DSGN	**Result Register Data Signed or Unsigned Representation** — This bit selects between signed and unsigned conversion data representation in the result registers. Signed data is represented as 2’s complement. Signed data is not available in right justification. See <st-bold>7.3.2.16 ATD Conversion Result Registers (ATDDRx) for details. 0 Unsigned data representation in the result registers. 1 Signed data representation in the result registers. Table 7-15 summarizes the result data formats available and how they are set up using the control bits. Table 7-16 illustrates the difference between the signed and unsigned, left justified output codes for an input signal range between 0 and 5.12 Volts.
5 SCAN	**Continuous Conversion Sequence Mode** — This bit selects whether conversion sequences are performed continuously or only once. If external trigger is enabled (ETRIGE=1) setting this bit has no effect, that means each trigger event starts a single conversion sequence. 0 Single conversion sequence 1 Continuous conversion sequences (scan mode)
4 MULT	**Multi-Channel Sample Mode** — When MULT is 0, the ATD sequence controller samples only from the specified analog input channel for an entire conversion sequence. The analog channel is selected by channel selection code (control bits CD/CC/CB/CA located in ATDCTL5). When MULT is 1, the ATD sequence controller samples across channels. The number of channels sampled is determined by the sequence length value (S8C, S4C, S2C, S1C). The first analog channel examined is determined by channel selection code (CC, CB, CA control bits); subsequent channels sampled in the sequence are determined by incrementing the channel selection code or wrapping around to AN0 (channel 0. 0 Sample only one channel 1 Sample across several channels

Table 8-8. LCD Duty and Bias

Duty	LCDCR0 Register		Backplanes				Bias (BIAS = 0)			Bias (BIAS = 1)		
	DUTY1	DUTY0	BP3	BP2	BP1	BP0	1/1	1/2	1/3	1/1	1/2	1/3
1/1	0	1	OFF	OFF	OFF	BP0	YES	NA	NA	YES	NA	NA
1/2	1	0	OFF	OFF	BP1	BP0	NA	YES	NA	NA	NA	YES
1/3	1	1	OFF	BP2	BP1	BP0	NA	NA	YES	NA	NA	YES
1/4	0	0	BP3	BP2	BP1	BP0	NA	NA	YES	NA	NA	YES

8.4.2 Operation in Wait Mode

The LCD32F4B driver system operation during wait mode is controlled by the LCD stop in wait (LCDSWAI) bit in the LCD control register 1 (LCDCR1). If LCDSWAI is reset, the LCD32F4B driver system continues to operate during wait mode. If LCDSWAI is set, the LCD32F4B driver system is turned off during wait mode. In this case, the LCD waveform generation clocks are stopped and the LCD32F4B drivers pull down to VSSX those frontplane and backplane pins that were enabled before entering wait mode. The contents of the LCD RAM and the LCD registers retain the values they had prior to entering wait mode.

8.4.3 Operation in Pseudo Stop Mode

The LCD32F4B driver system operation during pseudo stop mode is controlled by the LCD run in pseudo stop (LCDRPSTP) bit in the LCD control register 1 (LCDCR1). If LCDRPSTP is reset, the LCD32F4B driver system is turned off during pseudo stop mode. In this case, the LCD waveform generation clocks are stopped and the LCD32F4B drivers pull down to VSSX those frontplane and backplane pins that were enabled before entering pseudo stop mode. If LCDRPSTP is set, the LCD32F4B driver system continues to operate during pseudo stop mode. The contents of the LCD RAM and the LCD registers retain the values they had prior to entering pseudo stop mode.

8.4.4 Operation in Stop Mode

All LCD32F4B driver system clocks are stopped, the LCD32F4B driver system pulls down to VSSX those frontplane and backplane pins that were enabled before entering stop mode. Also, during stop mode, the contents of the LCD RAM and the LCD registers retain the values they had prior to entering stop mode. As a result, after exiting from stop mode, the LCD32F4B driver system clocks will run (if LCDEN = 1) and the frontplane and backplane pins retain the functionality they had prior to entering stop mode.

8.4.5 LCD Waveform Examples

Figure 8-9 through Figure 8-13 show the timing examples of the LCD output waveforms for the available modes of operation.

to a logic high state. PWM output on M2C1M results in a positive current flow through coil 1 when M2C1P is driven to a logic high state.

9.2.4 M3C0M/M3C0P/M3C1M/M3C1P — PWM Output Pins for Motor 3

High current PWM output pins that can be used for motor drive. These pins interface to the coils of motor 3. PWM output on M3C0M results in a positive current flow through coil 0 when M3C0P is driven to a logic high state. PWM output on M3C1M results in a positive current flow through coil 1 when M3C1P is driven to a logic high state.

9.3 Memory Map and Register Definition

This section provides a detailed description of all registers of the 10-bit 8-channel motor controller module.

9.3.1 Module Memory Map

Figure 9-2 shows the memory map of the 10-bit 8-channel motor controller module.

Figure 9-2. MC10B8C Memory Map

Offset	Register	Access
0x0000	Motor Controller Control Register 0 (MCCTL0)	RW
0x0001	Motor Controller Control Register 1 (MCCTL1)	RW
0x0002	Motor Controller Period Register (High Byte)	RW
0x0003	Motor Controller Period Register (Low Byte)	RW
0x0004	Reserved[1]	—
0x0005	Reserved	—
0x0006	Reserved	—
0x0007	Reserved	—
0x0008	Reserved	—
0x0009	Reserved	—
0x000A	Reserved	—
0x000B	Reserved	—
0x000C	Reserved	—
0x000D	Reserved	—
0x000E	Reserved	—
0x000F	Reserved	—
0x0010	Motor Controller Channel Control Register 0 (MCCC0)	RW
0x0011	Motor Controller Channel Control Register 1 (MCCC1)	RW
0x0012	Motor Controller Channel Control Register 2 (MCCC2)	RW
0x0013	Motor Controller Channel Control Register 3 (MCCC3)	RW
0x0014	Motor Controller Channel Control Register 4 (MCCC4)	RW
0x0015	Motor Controller Channel Control Register 5 (MCCC5)	RW

10.4.3 Operation in Low Power Modes

The SSD block can be configured for lower MCU power consumption in three different ways.

- Stop mode powers down the sigma-delta converter and halts clock to the modulus counter. Exit from Stop enables the sigma-delta converter and the clock to the modulus counter but due to the converter recovery time, the integration result should be ignored.
- Wait mode with SSDWAI bit set powers down the sigma-delta converter and halts the clock to the modulus counter. Exit from Wait enables the sigma-delta converter and clock to the modulus counter but due to the converter recovery time, the integration result should be ignored.
- Clearing SDCPU bit powers down the sigma-delta converter.

10.4.4 Stall Detection Flow

Figure 10-15 shows a flowchart and software setup for stall detection of a stepper motor. To control a second stepper motor, the SMS bit must be toggled during the SSD initialization.

Table 11-6. IIC Divider and Hold Values (Sheet 3 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
4F	136	26	60	70
50	96	18	36	50
51	112	18	44	58
52	128	26	52	66
53	144	26	60	74
54	160	34	68	82
55	176	34	76	90
56	208	42	92	106
57	256	42	116	130
58	160	18	76	82
59	192	18	92	98
5A	224	34	108	114
5B	256	34	124	130
5C	288	50	140	146
5D	320	50	156	162
5E	384	66	188	194
5F	480	66	236	242
60	320	34	156	162
61	384	34	188	194
62	448	66	220	226
63	512	66	252	258
64	576	98	284	290
65	640	98	316	322
66	768	130	380	386
67	960	130	476	482
68	640	66	316	322
69	768	66	380	386
6A	896	130	444	450
6B	1024	130	508	514
6C	1152	194	572	578
6D	1280	194	636	642
6E	1536	258	764	770
6F	1920	258	956	962
70	1280	130	636	642
71	1536	130	764	770
72	1792	258	892	898
73	2048	258	1020	1026
74	2304	386	1148	1154
75	2560	386	1276	1282
76	3072	514	1532	1538
77	3840	514	1916	1922
78	2560	258	1276	1282
79	3072	258	1532	1538
7A	3584	514	1788	1794
7B	4096	514	2044	2050

Table 11-6. IIC Divider and Hold Values (Sheet 5 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
B0	2560	260	1272	1284
B1	3072	260	1528	1540
B2	3584	516	1784	1796
B3	4096	516	2040	2052
B4	4608	772	2296	2308
B5	5120	772	2552	2564
B6	6144	1028	3064	3076
B7	7680	1028	3832	3844
B8	5120	516	2552	2564
B9	6144	516	3064	3076
BA	7168	1028	3576	3588
BB	8192	1028	4088	4100
BC	9216	1540	4600	4612
BD	10240	1540	5112	5124
BE	12288	2052	6136	6148
BF	15360	2052	7672	7684

11.3.2.3 IIC Control Register (IBCR)

	7	6	5	4	3	2	1	0
R	IBEN	IBIE	MS/SL	Tx/Rx	TXAK	0	0	IBSWAI
W						RSTA		
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 11-5. IIC Bus Control Register (IBCR)

Read and write anytime

flag by writing another byte to the transmitter buffer (SCIDRH/SCIDRL), while the shift register is shifting out the first byte.

To initiate an SCI transmission:

1. Configure the SCI:
   a) Select a baud rate. Write this value to the SCI baud registers (SCIBDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is 0. Writing to the SCIBDH has no effect without also writing to SCIBDL.
   b) Write to SCICR1 to configure word length, parity, and other configuration bits (LOOPS, RSRC, M, WAKE, ILT, PE, and PT).
   c) Enable the transmitter, interrupts, receive, and wake up as required, by writing to the SCICR2 register bits (TIE, TCIE, RIE, ILIE, TE, RE, RWU, and SBK). A preamble or idle character will now be shifted out of the transmitter shift register.
2. Transmit procedure for each byte:
   a) Poll the TDRE flag by reading the SCISR1 or responding to the TDRE interrupt. Keep in mind that the TDRE bit resets to 1.
   b) If the TDRE flag is set, write the data to be transmitted to SCIDRH/L, where the ninth bit is written to the T8 bit in SCIDRH if the SCI is in 9-bit data format. A new transmission will not result until the TDRE flag has been cleared.
3. Repeat step 2 for each subsequent transmission.

NOTE

The TDRE flag is set when the shift register is loaded with the next data to be transmitted from SCIDRH/L, which happens, generally speaking, a little over half-way through the stop bit of the previous frame. Specifically, this transfer occurs 9/16ths of a bit time AFTER the start of the stop bit of the previous frame.

Writing the TE bit from 0 to a 1 automatically loads the transmit shift register with a preamble of 10 logic 1s (if M = 0) or 11 logic 1s (if M = 1). After the preamble shifts out, control logic transfers the data from the SCI data register into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

Hardware supports odd or even parity. When parity is enabled, the most significant bit (MSB) of the data character is the parity bit.

The transmit data register empty flag, TDRE, in SCI status register 1 (SCISR1) becomes set when the SCI data register transfers a byte to the transmit shift register. The TDRE flag indicates that the SCI data register can accept new data from the internal data bus. If the transmit interrupt enable bit, TIE, in SCI control register 2 (SCICR2) is also set, the TDRE flag generates a transmitter interrupt request.

When the transmit shift register is not transmitting a frame, the TXD pin goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

14.4.2 Slave Mode

The SPI operates in slave mode when the MSTR bit in SPI Control Register1 is clear.

- SCK Clock

  In slave mode, SCK is the SPI clock input from the master.
- MISO and MOSI Pins

  In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI Control Register 2.
- $\overline{SS}$ Pin

  The $\overline{SS}$ pin is the slave select input. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be low. $\overline{SS}$ must remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state.

  The $\overline{SS}$ input also controls the serial data output pin, if $\overline{SS}$ is high (not selected), the serial data output pin is high impedance, and, if $\overline{SS}$ is low the first bit in the SPI Data Register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{SS}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register takes place.

Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

NOTE

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave's serial data output line.

As long as no more than one slave device drives the system slave's serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI Control Register 1 is clear, odd numbered edges on the SCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

If the CPHA bit is set, even numbered edges on the SCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

When CPHA is set, the first edge is used to get the first data bit onto the serial data output pin. When CPHA is clear and the $\overline{SS}$ input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the eighth shift, the transfer is considered complete and the received data is transferred into the SPI Data Register. To indicate transfer is complete, the SPIF flag in the SPI Status Register is set.

NOTE

A change of the bits CPOL, CPHA, SSOE, LSBFE, MODFEN, SPC0 and BIDIROE with SPC0 set in slave mode will corrupt a transmission in progress and has to be avoided.

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

The SCK is output for the master mode and input for the slave mode.

The $\overline{SS}$ is the input or output for the master mode, and it is always the input for the slave mode.

The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

NOTE

In bidirectional master mode, with mode fault enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to slave mode, in this case MISO becomes occupied by the SPI and MOSI is not used. This has to be considered, if the MISO pin is used for other purpose.

14.4.6 Error Conditions

The SPI has one error condition:

- Mode fault error

14.4.6.1 Mode Fault Error

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation, the MODF bit in the SPI Status Register is set automatically provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO and MOSI pins are forced to be high impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI Status Register (with MODF set) followed by a write to SPI Control Register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

— Data associated with event B trigger modes
— Detail report mode stores address and data for all cycles except program (P) and free (f) cycles
— Current instruction address when in profiling mode
— BGND is not considered a change-of-flow (cof) by the debugger

19.1.2 Modes of Operation

There are two main modes of operation: breakpoint mode and debug mode. Each one is mutually exclusive of the other and selected via a software programmable control bit.

In the breakpoint mode there are two sub-modes of operation:

- Dual address mode, where a match on either of two addresses will cause the system to enter background debug mode (BDM) or initiate a software interrupt (SWI).
- Full breakpoint mode, where a match on address and data will cause the system to enter background debug mode (BDM) or initiate a software interrupt (SWI).

In debug mode, there are several sub-modes of operation.

- Trigger modes

  There are many ways to create a logical trigger. The trigger can be used to capture bus information either starting from the trigger or ending at the trigger. Types of triggers (A and B are registers):

  — A only
  — A or B
  — A then B
  — Event only B (data capture)
  — A then event only B (data capture)
  — A and B, full mode
  — A and not B, full mode
  — Inside range
  — Outside range

- Capture modes

  There are several capture modes. These determine which bus information is saved and which is ignored.

  — Normal: save change-of-flow program fetches
  — Loop1: save change-of-flow program fetches, ignoring duplicates
  — Detail: save all bus operations except program and free cycles
  — Profile: poll target from external device

19.1.3 Block Diagram

Figure 19-1 is a block diagram of this module in breakpoint mode. Figure 19-2 is a block diagram of this module in debug mode.

19.3 Memory Map and Register Definition

A summary of the registers associated with the DBG sub-block is shown in Figure 19-3. Detailed descriptions of the registers and bits are given in the subsections that follow.

19.3.1 Module Memory Map

Table 19-2. DBG Memory Map

Address Offset	Use	Access
	Debug Control Register (DBGC1)	R/W
	Debug Status and Control Register (DBGSC)	R/W
	Debug Trace Buffer Register High (DBGTBH)	R
	Debug Trace Buffer Register Low (DBGTBL)	R
4	Debug Count Register (DBGCNT)	R
5	Debug Comparator C Extended Register (DBGCCX)	R/W
6	Debug Comparator C Register High (DBGCCH)	R/W
	Debug Comparator C Register Low (DBGCCL)	R/W
8	Debug Control Register 2 (DBGC2) / (BKPCT0)	R/W
9	Debug Control Register 3 (DBGC3) / (BKPCT1)	R/W
A	Debug Comparator A Extended Register (DBGCAX) / (/BKP0X)	R/W
B	Debug Comparator A Register High (DBGCAH) / (BKP0H)	R/W
	Debug Comparator A Register Low (DBGCAL) / (BKP0L)	R/W
	Debug Comparator B Extended Register (DBGCBX) / (BKP1X)	R/W
E	Debug Comparator B Register High (DBGCBH) / (BKP1H)	R/W
F	Debug Comparator B Register Low (DBGCBL) / (BKP1L)	R/W

19.3.2 Register Descriptions

This section consists of the DBG register descriptions in address order. Most of the register bits can be written to in either BKP or DBG mode, although they may not have any effect in one of the modes. However, the only bits in the DBG module that can be written while the debugger is armed (ARM = 1) are DBGEN and ARM

Name[1]		Bit 7	6	5	4	3	2	1	Bit 0
DBGC1	R	DBGEN	ARM	TRGSEL	BEGIN	DBGBRK	0	CAPMOD	
	W								
DBGSC	R	AF	BF	CF	0	TRG			
	W								

= Unimplemented or Reserved

Figure 19-3. DBG Register Summary

Read: Anytime when register is in the map

Write: Anytime when register is in the map

Port A bits 7 through 0 are associated with address lines A15 through A8 respectively and data lines D15/D7 through D8/D0 respectively. When this port is not used for external addresses such as in single-chip mode, these pins can be used as general-purpose I/O. Data direction register A (DDRA) determines the primary direction of each pin. DDRA also determines the source of data for a read of PORTA.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

NOTE

To ensure that you read the value present on the PORTA pins, always wait at least one cycle after writing to the DDRA register before reading from the PORTA register.

21.3.2.2 Port B Data Register (PORTB)

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0
Single Chip	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Expanded Wide, Emulation Narrow with IVIS, and Peripheral	AB/DB7	AB/DB6	AB/DB5	AB/DB4	AB/DB3	AB/DB2	AB/DB1	AB/DB0
Expanded Narrow	AB7	AB6	AB5	AB4	AB3	AB2	AB1	AB0

Figure 21-3. Port A Data Register (PORTB)

Read: Anytime when register is in the map

Write: Anytime when register is in the map

Port B bits 7 through 0 are associated with address lines A7 through A0 respectively and data lines D7 through D0 respectively. When this port is not used for external addresses, such as in single-chip mode, these pins can be used as general-purpose I/O. Data direction register B (DDRB) determines the primary direction of each pin. DDRB also determines the source of data for a read of PORTB.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

NOTE

To ensure that you read the value present on the PORTB pins, always wait at least one cycle after writing to the DDRB register before reading from the PORTB register.

Table A-20. SPI Slave Mode Timing Characteristics

Conditions are shown in Table A-4 unless otherwise noted, CLOAD = 200pF on all outputs							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Operating Frequency	f_{op}	DC		1/4	f_{bus}
1	P	SCK Period $t_{sck} = 1./f_{op}$	t_{sck}	4		2048	t_{bus}
2	D	Enable Lead Time	t_{lead}	1			t_{cyc}
3	D	Enable Lag Time	t_{lag}	1			t_{cyc}
4	D	Clock (SCK) High or Low Time	t_{wsck}	$t_{cyc} - 30$			ns
5	D	Data Setup Time (Inputs)	t_{su}	25			ns
6	D	Data Hold Time (Inputs)	t_{hi}	25			ns
7	D	Slave Access Time	t_a			1	t_{cyc}
8	D	Slave MISO Disable Time	t_{dis}			1	t_{cyc}
9	D	Data Valid (after SCK Edge)	t_v			25	ns
10	D	Data Hold Time (Outputs)	t_{ho}	0			ns
11	D	Rise Time Inputs and Outputs	t_r			25	ns
12	D	Fall Time Inputs and Outputs	t_f			25	ns

A.8 LCD_32F4B

Table A-21. LCD_32F4B Driver Electrical Characteristics

Characteristic	Symbol	Min.	Typ.	Max.	Unit
LCD Supply Voltage	VLCD	-0.25	-	VDDX + 0.25	V
LCD Output Impedance(BP[3:0],FP[31:0]) for outputs to charge to higher voltage level or to GND [1]	$Z_{BP/FP}$	-	-	5.0	kΩ
LCD Output Current (BP[3:0],FP[31:0]) for outputs to discharge to lower voltage level except GND [2]	$I_{BP/FP}$	50	-	-	uA

[1] Outputs measured one at a time, low impedance voltage source connected to the VLCD pin.

[2] Outputs measured one at a time, low impedance voltage source connected to the VLCD pin.

A.9 External Bus Timing

A timing diagram of the external multiplexed-bus is illustrated in Figure A-10 with the actual timing values shown on table Table A-22. All major bus signals are included in the diagram. While both a data write and data read cycle are shown, only one or the other would occur on a particular bus cycle.

The expanded bus timings are highly dependent on the load conditions. The timing parameters shown assume a balanced load across all outputs.

Appendix D
Derivative Differences

Table D-1 shows the different options for the MC9S12HZ256 members.

Table D-1. Package, Memory and Peripheral Options for MC9S12HZ256 members

Device	Package	Flash	ROM	RAM	EEPROM	CAN	SCI	SPI	IIC	SSD	LCD	Motor	A/D	PWM	TIM	I/O
MC9S12HZ256	112 LQFP	256K	0	12K	2K	2	2	1	1	4	32x4	16/4	1/16	6-ch	8-ch	85
MC9S12HZ128	112 LQFP	128K	0	6K	2K	2	2	1	1	4	32x4	16/4	1/16	6-ch	8-ch	85
MC9S12HZ64	112 LQFP	64K	0	4K	1K	1	1	1	0	4	24x4	16/4	1/8	4-ch	8-ch	69
	80 QFP					1	1	0	0	4	20x4	16/4	1/7	4-ch	4-ch	59
MC9S12HN64	112 LQFP	64K	0	4K	1K	0	1	1	0	4	24x4	16/4	1/8	4-ch	8-ch	69
	80 QFP					0	1	0	0	4	20x4	16/4	1/7	4-ch	4-ch	59
MC3S12HZ256	112 LQFP	0	256K	12K	0	2	2	1	1	4	32x4	16/4	1/16	6-ch	8-ch	85
MC3S12HZ128	112 LQFP	0	128K	6K	0	1	2	1	1	4	32x4	16/4	1/16	6-ch	8-ch	85
MC3S12HZ64	112 LQFP	0	64K	4K	0	1	1	1	0	4	24x4	16/4	1/8	4-ch	8-ch	69
	80 QFP					1	1	0	0	4	20x4	16/4	1/7	4-ch	4-ch	59
MC3S12HN64	112 LQFP	0	64K	4K	0	0	1	1	0	4	24x4	16/4	1/8	4-ch	8-ch	69
	80 QFP					0	1	0	0	4	20x4	16/4	1/7	4-ch	4-ch	59
MC3S12HZ32	80 QFP	0	32K	2K	0	1	1	0	0	4	20x4	16/4	1/7	4-ch	4-ch	59
MC3S12HN32	80 QFP	0	32K	2K	0	0	1	0	0	4	20x4	16/4	1/7	4-ch	4-ch	59