



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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	87
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	Internal
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/s9s08dz128f2mllr

Addendum Rev.2 to Rev. 1 of the MC9S08DZ128 Series Data Sheet

This addendum identifies changes to Rev. 1 of the MC9S08DZ128 Series Data Sheet. The changes described in this addendum have not been implemented in the specified pages.

1 MCG Control Register 3 Field Descriptions

Location:	Table 8-7, Page 176
------------------	---------------------

The last sentence of bit 4 (DIV32) description should be changed from "Writes to this bit are ignored if PLLS bit is set." to "DIV32 must be cleared when the PLL is selected." The correct description should be:

Field	Description
4 DIV32	Divide-by-32 Enable — Controls an additional divide-by-32 factor to the external reference clock for the FLL when RANGE bit is set. When the RANGE bit is 0, this bit has no effect. DIV32 must be cleared when the PLL is selected. 0 Divide-by-32 is disabled. 1 Divide-by-32 is enabled when RANGE=1.

2 Initializing the MCG

Location:	Section 8.5.1.1, Page 186
------------------	---------------------------

The last sentence in the note after step 6 should be removed. The note should be

2.2 Recommended System Connections

Figure 2-4 shows pin connections that are common to MC9S08DZ128 Series application systems.

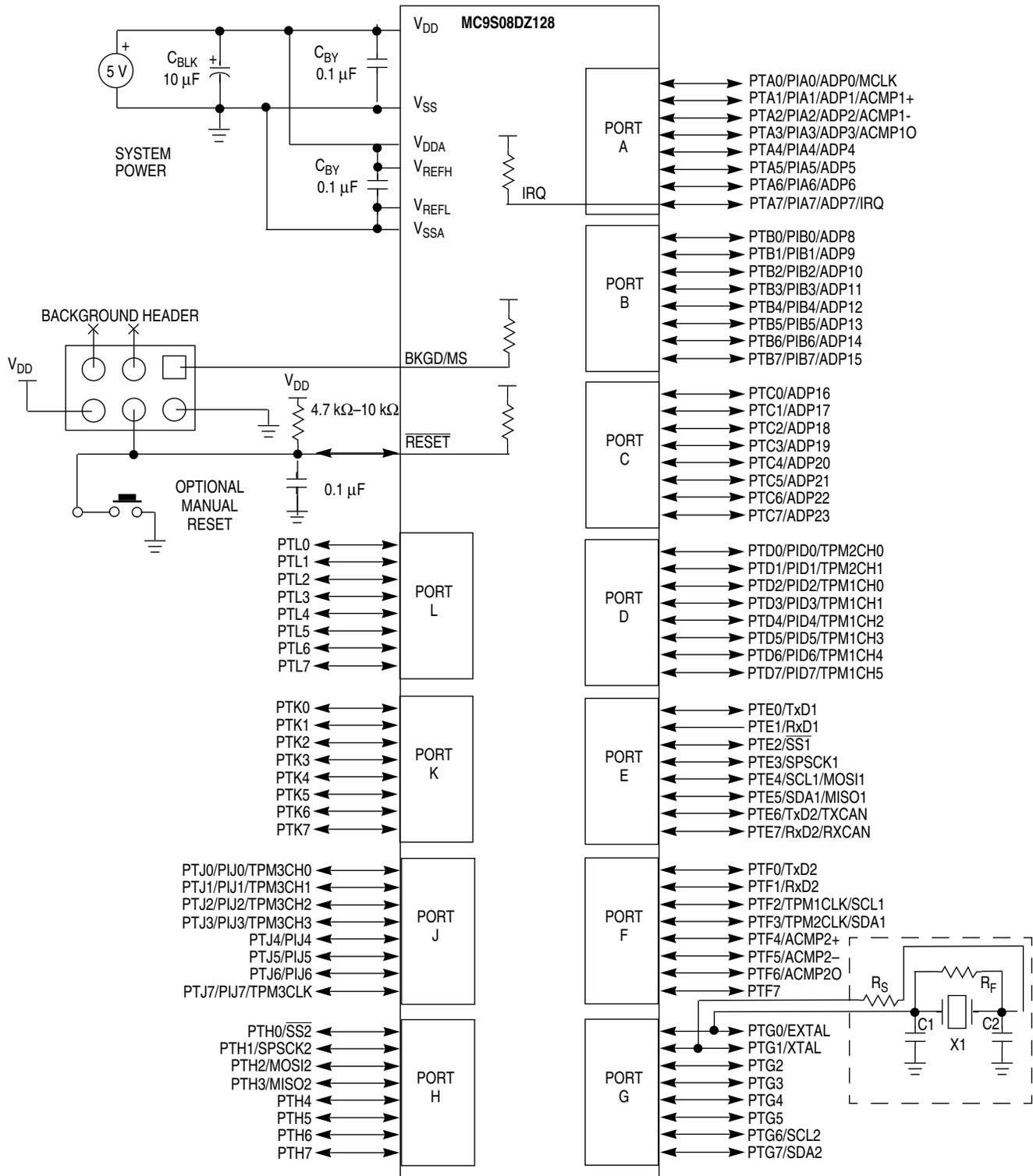


Figure 2-4. Basic System Connections (Shown in 100Pin Package)

3.6 Stop Modes

One of two stop modes is entered upon execution of a STOP instruction when the STOPE bit in SOPT1 register is set. In both stop modes, all internal clocks are halted. The MCG module can be configured to leave the reference clocks running. See Chapter 8, “Multi-Purpose Clock Generator (S08MCGV2),” for more information.

Table 3-1 shows all of the control bits that affect stop mode selection and the mode selected under various conditions. The selected mode is entered following the execution of a STOP instruction.

Table 3-1. Stop Mode Selection

STOPE	ENBDM ¹	LVDE	LVDSE	PPDC	Stop Mode
0	x	x		x	Stop modes disabled; illegal opcode reset if STOP instruction executed
1	1	x		x	Stop3 with BDM enabled ²
1	0	Both bits must be 1		x ³	Stop3 with voltage regulator active
1	0	Either bit a 0		0	Stop3
1	0	Either bit a 0		1	Stop2

¹ ENBDM is located in the BDCSCR, which is only accessible through BDC commands, see the Development Support chapter.

² When in Stop3 mode with BDM enabled, The S_{IDD} will be near R_{IDD} levels because internal clocks are enabled.

³ If LVD = 1 in stop, the MCU enters stop3, regardless of the configuration of PPDC.

3.6.1 Stop3 Mode

Stop3 mode is entered by executing a STOP instruction under the conditions as shown in Table 3-1. The states of all of the internal registers and logic, RAM contents, and I/O pin states are maintained.

Exit from stop3 is done by asserting $\overline{\text{RESET}}$ or an asynchronous interrupt pin. The asynchronous interrupt pins are IRQ, PIA0–PIA7, PIB0–PIB7, PID0–PID7, and PIJ0–PIJ7. Exit from stop3 can also be done by the low voltage detection (LVD) reset, the low voltage warning (LVW) interrupt, the ADC conversion complete interrupt, the analog comparator interrupt, the real-time clock (RTC) interrupt, the MSCAN wake-up interrupt, or the SCI receiver interrupt.

If stop3 is exited by means of the $\overline{\text{RESET}}$ pin, the MCU will be reset and operation will resume after fetching the reset vector. Exit by means of an interrupt will result in the MCU fetching the appropriate interrupt vector.

3.6.1.1 LVD Enabled in Stop3 Mode

The LVD system is capable of generating either an interrupt or a reset when the supply voltage drops below the LVD voltage. If the LVD is enabled in stop (LVDE and LVDSE bits in SPMSC1 both set) at the time the CPU executes a STOP instruction, then the voltage regulator remains active during stop mode.

For the ADC to operate or for the ACMP to be used when comparing with an internal voltage, the LVD must be left enabled when entering stop3.

NOTE

The FCBEF flag will not set after launching the sector erase abort command. If an attempt is made to start a new command write sequence with a sector erase abort operation active, the FACCERR flag in the FSTAT register will be set. A new command write sequence may be started after clearing the ACCERR flag, if set.

NOTE

The sector erase abort command should be used sparingly since a sector erase operation that is aborted counts as a complete program/erase cycle.

4.6.6 Access Errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will cause the access error flag (FACCERR) in FSTAT to be set. FACCERR must be cleared by writing a 1 to FACCERR in FSTAT before any command can be processed.

- Writing to a FLASH address before the internal FLASH and EEPROM clock frequency has been set by writing to the FCDIV register.
- Writing to a FLASH address while FCBEF is not set. (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a FLASH address before launching the previous command. (There is only one write to FLASH for every command.)
- Writing a second time to FCMD before launching the previous command. (There is only one write to FCMD for every command.)
- Writing to any FLASH control register other than FCMD after writing to a FLASH address.
- Writing any command code other than the six allowed codes (0x05, 0x20, 0x25, 0x40, 0x41, or 0x47) to FCMD.
- Accessing (read or write) any FLASH control register other than to write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD.
- The MCU enters stop mode while a program or erase command is in progress. (The command is aborted.)
- Writing the byte program, burst program, sector erase or sector erase abort command code (0x20, 0x25, 0x40, or 0x47) with a background debug command while the MCU is secured. (The background debug controller can do blank check and mass erase commands only when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command.

4.6.10 EEPROM Mapping

Only half of the EEPROM is in the memory map. The EPGSEL bit in FCNFG register selects which half of the array can be accessed in foreground while the other half can not be accessed in background. There are two mapping mode options that can be selected to configure the 8-byte EEPROM sectors: 4-byte mode and 8-byte mode. Each mode is selected by the EPGMOD bit in the FOPT register.

In 4-byte sector mode (EPGMOD = 0), each 8-byte sector splits four bytes on foreground and four bytes on background but on the same addresses. The EPGSEL bit selects which four bytes can be accessed. During a sector erase, the entire 8-byte sector (four bytes in foreground and four bytes in background) is erased.

In 8-byte sector mode (EPGMOD = 1), each entire 8-byte sector is in a single page. The EPGSEL bit selects which sectors are on background. During a sector erase, the entire 8-byte sector in foreground is erased.

4.6.11 FLASH and EEPROM Registers and Control Bits

The FLASH and EEPROM modules have seven 8-bit registers in the high-page register space and three locations in the nonvolatile register space in FLASH memory. Two of those locations are copied into two corresponding high-page control registers at reset. There is also an 8-byte comparison key in FLASH memory. Refer to [Table 4-3](#) and [Table 4-5](#) for the absolute address assignments for all FLASH and EEPROM registers. This section refers to registers and control bits only by their names. A Freescale Semiconductor-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

4.6.11.1 FLASH and EEPROM Clock Divider Register (FCDIV)

Bit 7 of this register is a read-only status flag. Bits 6 through 0 may be read at any time but can be written only one time. Before any erase or programming operations are possible, write to this register to set the frequency of the clock for the nonvolatile memory system within acceptable limits.

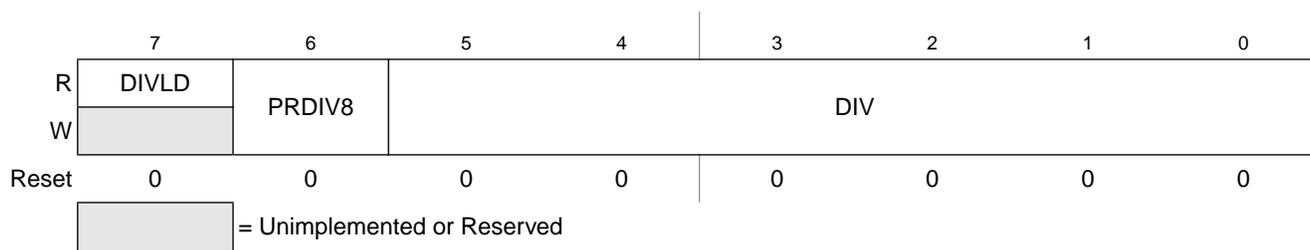


Figure 4-14. FLASH and EEPROM Clock Divider Register (FCDIV)

Table 4-20. FLASH Block Protection

FPS	FPOP	Protected Address Range Relative to Flash Array base		Protected Size
		Flash Array 0	Flash Array 1	
0x1F	1	No protection	No protection	0 Kbytes
0x1E		0x0_E000–0x0_FFFF		8 Kbytes
0x1D		0x0_C000–0x0_FFFF		16 Kbytes
0x1C		0x0_A000–0x0_FFFF		24 Kbytes
0x1B		0x0_8000–0x0_FFFF		32 Kbytes
0x1A		0x0_6000–0x0_FFFF		40 Kbytes
0x19		0x0_4000–0x0_FFFF		48 Kbytes
0x18		0x0_2000–0x0_FFFF		56 Kbytes
0x17		0x0_0000–0x0_FFFF		64 Kbytes
0x16		0x0_0000–0x0_FFFF	0x1_F000–0x1_FFFF	68 Kbytes
0x15		0x0_0000–0x0_FFFF	0x1_E000–0x1_FFFF	72 Kbytes
0x14		0x0_0000–0x0_FFFF	0x1_D000–0x1_FFFF	76 Kbytes
0x13		0x0_0000–0x0_FFFF	0x1_C000–0x1_FFFF	80 Kbytes
0x12		0x0_0000–0x0_FFFF	0x1_B000–0x1_FFFF	84 Kbytes
0x11		0x0_0000–0x0_FFFF	0x1_A000–0x1_FFFF	88 Kbytes
0x10		0x0_0000–0x0_FFFF	0x1_9000–0x1_FFFF	92 Kbytes
0x0F		0x0_0000–0x0_FFFF	0x1_8000–0x1_FFFF	96 Kbytes
0x0E		0x0_0000–0x0_FFFF	0x1_7800–0x1_FFFF	98 Kbytes
0x0D		0x0_0000–0x0_FFFF	0x1_7000–0x1_FFFF	100 Kbytes
0x0C		0x0_0000–0x0_FFFF	0x1_6800–0x1_FFFF	102 Kbytes
...	
0x05		0x0_0000–0x0_FFFF	0x1_3000–0x1_FFFF	116 Kbytes
0x04		0x0_0000–0x0_FFFF	0x1_2800–0x1_FFFF	118 Kbytes
0x03		0x0_0000–0x0_FFFF	0x1_2000–0x1_FFFF	120 Kbytes
0x02		0x0_0000–0x0_FFFF	0x1_1800–0x1_FFFF	122 Kbytes
0x01		0x0_0000–0x0_FFFF	0x1_1000–0x1_FFFF	124 Kbytes
0x00		0x0_0000–0x0_FFFF	0x1_0800–0x1_FFFF	126 Kbytes
-	0	0x0_0000–0x0_FFFF	0x1_0000–0x1_FFFF	128 Kbytes

6.5.4.3 Port D Pull Enable Register (PTDPE)

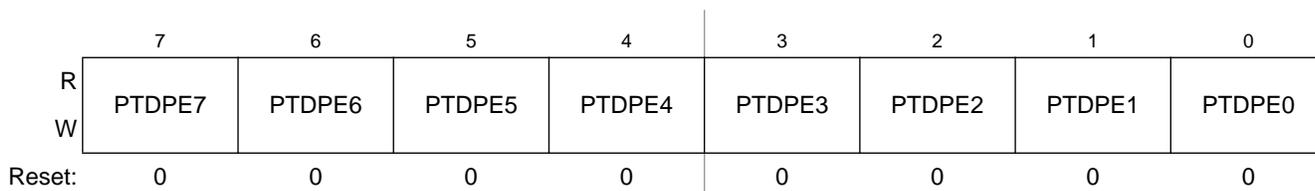


Figure 6-26. Internal Pull Enable for Port D Register (PTDPE)

Table 6-24. PTDPE Register Field Descriptions

Field	Description
7:0 PTDPE[7:0]	<p>Internal Pull Enable for Port D Bits — Each of these control bits determines if the internal pull-up or pull-down device is enabled for the associated PTD pin. For port D pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled.</p> <p>0 Internal pull-up/pull-down device disabled for port D bit n. 1 Internal pull-up/pull-down device enabled for port D bit n.</p>

NOTE

Pull-down devices only apply when using pin interrupt functions, when corresponding edge select and pin select functions are configured.

6.5.4.4 Port D Slew Rate Enable Register (PTDSE)

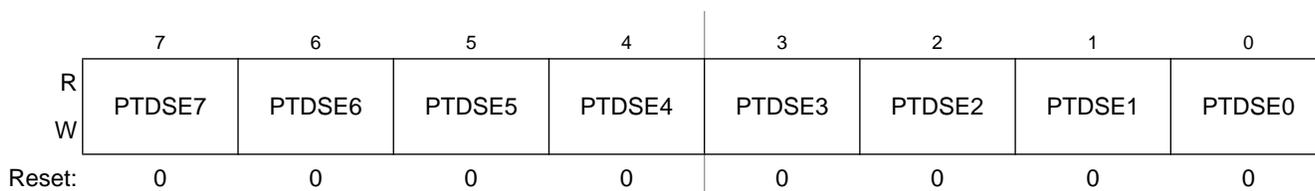


Figure 6-27. Slew Rate Enable for Port D Register (PTDSE)

Table 6-25. PTDSE Register Field Descriptions

Field	Description
7:0 PTDSE[7:0]	<p>Output Slew Rate Enable for Port D Bits — Each of these control bits determines if the output slew rate control is enabled for the associated PTD pin. For port D pins that are configured as inputs, these bits have no effect.</p> <p>0 Output slew rate control disabled for port D bit n. 1 Output slew rate control enabled for port D bit n.</p>

Note: Slew rate reset default values may differ between engineering samples and final production parts. Always initialize slew rate control to the desired value to ensure correct operation.

Table 8-2. FLL External Reference Divide Factor

RDIV	Divide Factor		
	RANGE:DIV32 0:X	RANGE:DIV32 1:0	RANGE:DIV32 1:1
0	1	1	32
1	2	2	64
2	4	4	128
3	8	8	256
4	16	16	512
5	32	32	1024
6	64	64	Reserved
7	128	128	Reserved

Table 8-3. PLL External Reference Divide Factor

RDIV	Divide Factor
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

9.1.2 Features

The ACMP has the following features:

- Full rail to rail supply operation.
- Selectable interrupt on rising edge, falling edge, or either rising or falling edges of comparator output.
- Option to compare to fixed internal bandgap reference voltage.
- Option to allow comparator output to be visible on a pin, ACMPxO.
- Can operate in stop3 mode

9.1.3 Modes of Operation

This section defines the ACMP operation in wait, stop and background debug modes.

9.1.3.1 ACMP in Wait Mode

The ACMP continues to run in wait mode if enabled before executing the WAIT instruction. Therefore, the ACMP can be used to bring the MCU out of wait mode if the ACMP interrupt, ACIE is enabled. For lowest possible current consumption, the ACMP should be disabled by software if not required as an interrupt source during wait mode.

9.1.3.2 ACMP in Stop Modes

9.1.3.2.1 Stop3 Mode Operation

The ACMP continues to operate in Stop3 mode if enabled and compare operation remains active. If ACOPE is enabled, comparator output operates as in the normal operating mode and comparator output is placed onto the external pin. The MCU is brought out of stop when a compare event occurs and ACIE is enabled; ACF flag sets accordingly.

If stop is exited with a reset, the ACMP will be put into its reset state.

9.1.3.2.2 Stop2 and Stop1 Mode Operation

During either Stop2 and Stop1 mode, the ACMP module will be fully powered down. Upon wake-up from Stop2 or Stop1 mode, the ACMP module will be in the reset state.

9.1.3.3 ACMP in Active Background Mode

When the microcontroller is in active background mode, the ACMP will continue to operate normally.

9.1.4 Block Diagram

The block diagram for the Analog Comparator module is shown [Figure 9-2](#).

11.1.4 Block Diagram

Figure 11-2 is a block diagram of the IIC.

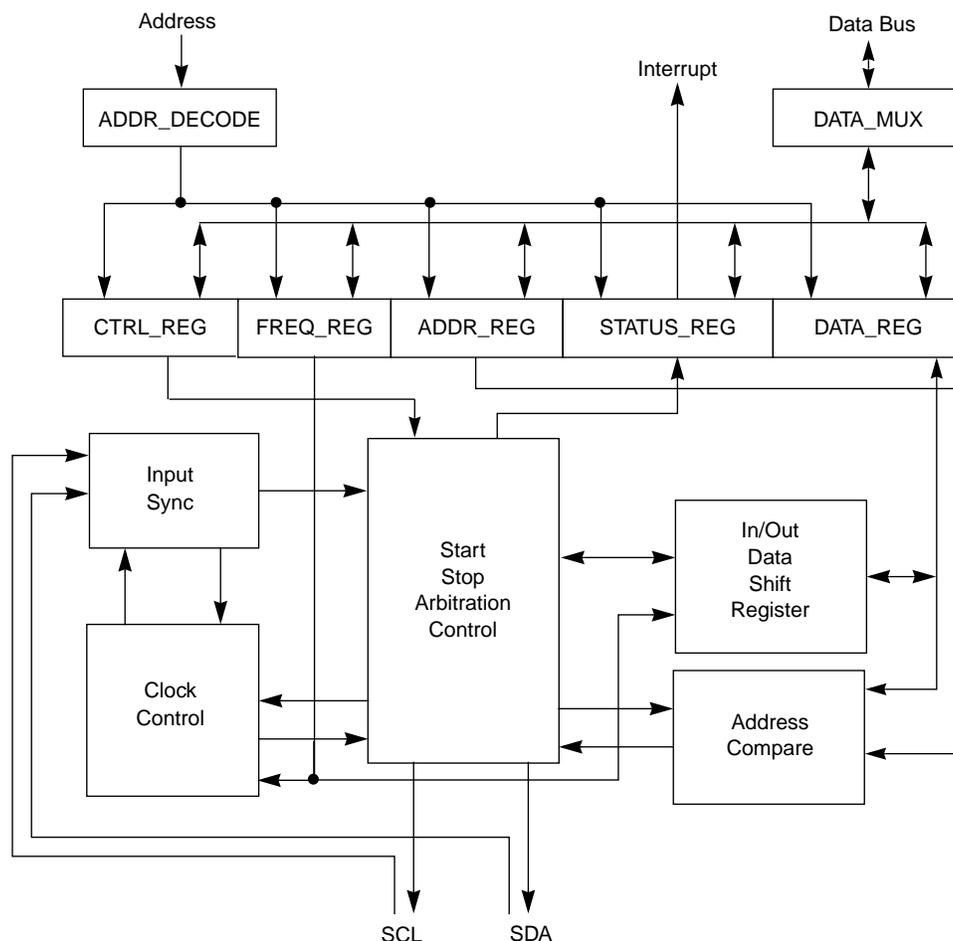


Figure 11-2. IIC Functional Block Diagram

11.2 External Signal Description

This section describes each user-accessible pin signal.

11.2.1 SCL — Serial Clock Line

The bidirectional SCL is the serial clock line of the IIC system.

11.2.2 SDA — Serial Data Line

The bidirectional SDA is the serial data line of the IIC system.

11.3 Register Definition

This section consists of the IIC register descriptions in address order.

11.3.3 IIC Control Register (IICxC1)

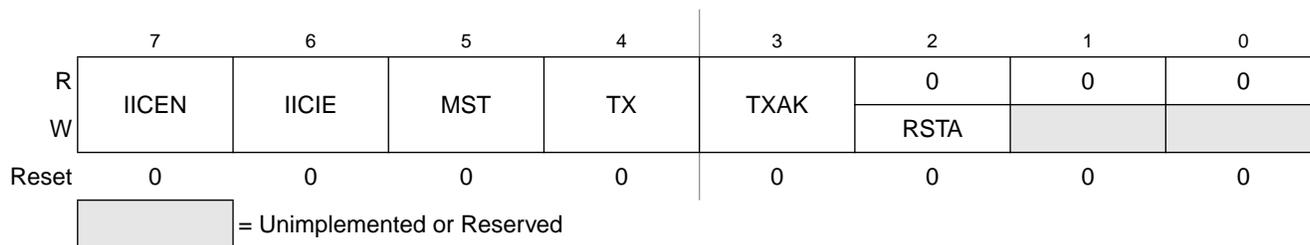


Figure 11-5. IIC Control Register (IICxC1)

Table 11-6. IICxC1 Field Descriptions

Field	Description
7 IICEN	IIC Enable. The IICEN bit determines whether the IIC module is enabled. 0 IIC is not enabled 1 IIC is enabled
6 IICIE	IIC Interrupt Enable. The IICIE bit determines whether an IIC interrupt is requested. 0 IIC interrupt request not enabled 1 IIC interrupt request enabled
5 MST	Master Mode Select. The MST bit changes from a 0 to a 1 when a start signal is generated on the bus and master mode is selected. When this bit changes from a 1 to a 0 a stop signal is generated and the mode of operation changes from master to slave. 0 Slave mode 1 Master mode
4 TX	Transmit Mode Select. The TX bit selects the direction of master and slave transfers. In master mode, this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit is always high. When addressed as a slave, this bit should be set by software according to the SRW bit in the status register. 0 Receive 1 Transmit
3 TXAK	Transmit Acknowledge Enable. This bit specifies the value driven onto the SDA during data acknowledge cycles for master and slave receivers. 0 An acknowledge signal is sent out to the bus after receiving one data byte 1 No acknowledge signal response is sent
2 RSTA	Repeat start. Writing a 1 to this bit generates a repeated start condition provided it is the current master. This bit is always read as cleared. Attempting a repeat at the wrong time results in loss of arbitration.

field of the CAN frame, is received into the next available RxBG. If the MSCAN receives an invalid message in its RxBG (wrong identifier, transmission errors, etc.) the actual contents of the buffer will be over-written by the next message. The buffer will then not be shifted into the FIFO.

When the MSCAN module is transmitting, the MSCAN receives its own transmitted messages into the background receive buffer, RxBG, but does not shift it into the receiver FIFO, generate a receive interrupt, or acknowledge its own messages on the CAN bus. The exception to this rule is in loopback mode (see Section 12.3.2, “MSCAN Control Register 1 (CANCTL1)”) where the MSCAN treats its own messages exactly like all other incoming messages. The MSCAN receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN must be prepared to become a receiver.

An overrun condition occurs when all receive message buffers in the FIFO are filled with correctly received messages with accepted identifiers and another message is correctly received from the CAN bus with an accepted identifier. The latter message is discarded and an error interrupt with overrun indication is generated if enabled (see Section 12.5.7.5, “Error Interrupt”). The MSCAN remains able to transmit messages while the receiver FIFO is full, but all incoming messages are discarded. As soon as a receive buffer in the FIFO is available again, new valid messages will be accepted.

12.5.3 Identifier Acceptance Filter

The MSCAN identifier acceptance registers (see Section 12.3.11, “MSCAN Identifier Acceptance Control Register (CANIDAC)”) define the acceptable patterns of the standard or extended identifier (ID[10:0] or ID[28:0]). Any of these bits can be marked ‘don’t care’ in the MSCAN identifier mask registers (see Section 12.3.16, “MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)”).

A filter hit is indicated to the application software by a set receive buffer full flag (RXF = 1) and three bits in the CANIDAC register (see Section 12.3.11, “MSCAN Identifier Acceptance Control Register (CANIDAC)”). These identifier hit flags (IDHIT[2:0]) clearly identify the filter section that caused the acceptance. They simplify the application software’s task to identify the cause of the receiver interrupt. If more than one hit occurs (two or more filters match), the lower hit has priority.

A very flexible programmable generic identifier acceptance filter has been introduced to reduce the CPU interrupt loading. The filter is programmable to operate in four different modes (see Bosch CAN 2.0A/B protocol specification):

- Two identifier acceptance filters, each to be applied to:
 - The full 29 bits of the extended identifier and to the following bits of the CAN 2.0B frame:
 - Remote transmission request (RTR)
 - Identifier extension (IDE)
 - Substitute remote request (SRR)
 - The 11 bits of the standard identifier plus the RTR and IDE bits of the CAN 2.0A/B messages¹. This mode implements two filters for a full length CAN 2.0B compliant extended identifier. Figure 12-39 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces a filter 0 hit. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces a filter 1 hit.

¹.Although this mode can be used for standard identifiers, it is recommended to use the four or eight identifier acceptance filters for standard identifiers

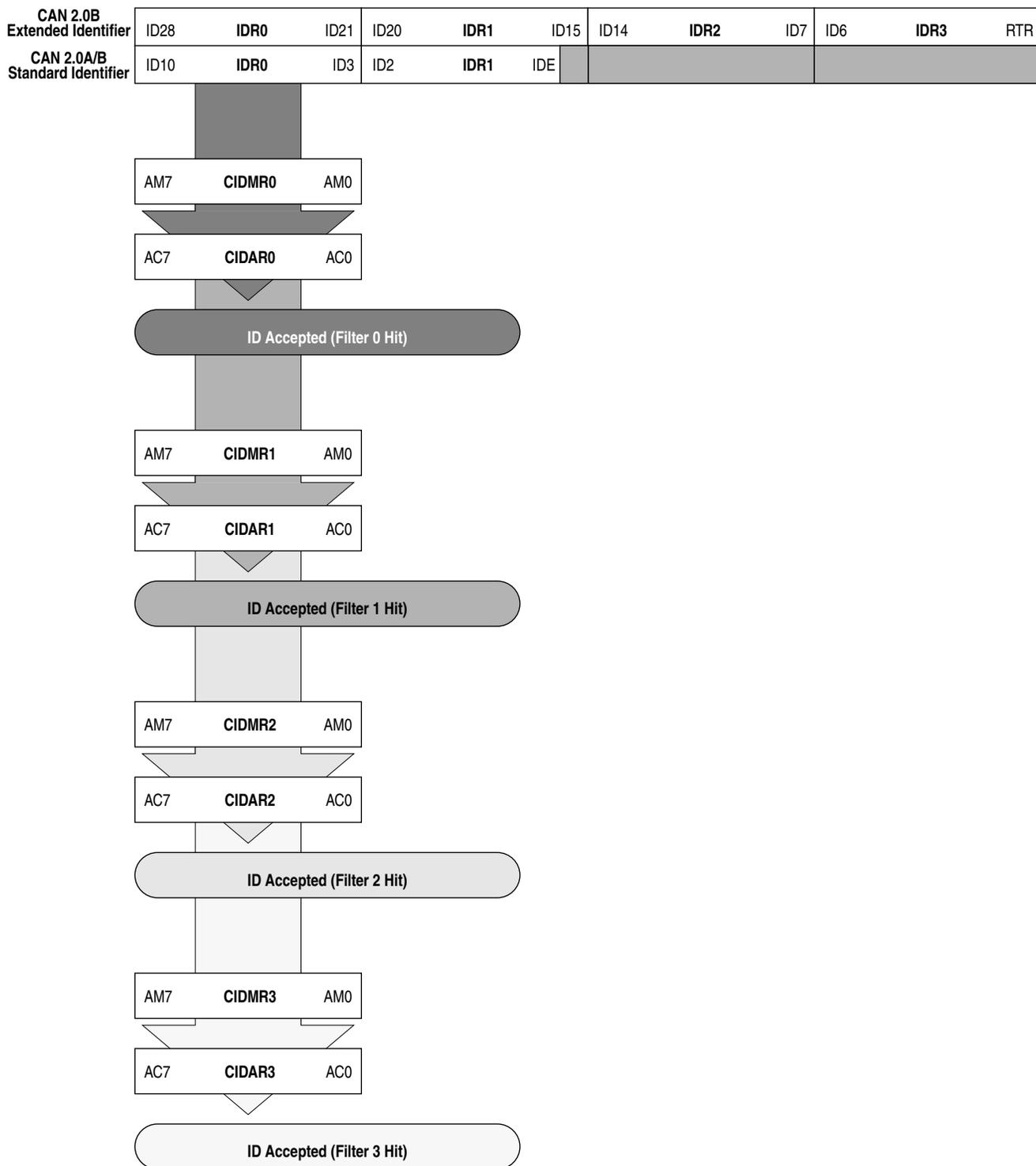


Figure 12-41. 8-bit Maskable Identifier Acceptance Filters

MSCAN filter uses three sets of registers to provide the filter configuration. Firstly, the CANIDAC register determines the configuration of the banks into filter sizes and number of filters. Secondly, registers CANIDMR0/1/2/3 determine those bits on which the filter will operate by placing a '0' at the appropriate

13.5 Functional Description

An SPI transfer is initiated by checking for the SPI transmit buffer empty flag (SPTEF = 1) and then writing a byte of data to the SPI data register (SPIxD) in the master SPI device. When the SPI shift register is available, this byte of data is moved from the transmit data buffer to the shifter, SPTEF is set to indicate there is room in the buffer to queue another transmit character if desired, and the SPI serial transfer starts.

During the SPI transfer, data is sampled (read) on the MISO pin at one SPSCCK edge and shifted, changing the bit value on the MOSI pin, one-half SPSCCK cycle later. After eight SPSCCK cycles, the data that was in the shift register of the master has been shifted out the MOSI pin to the slave while eight bits of data were shifted in the MISO pin into the master's shift register. At the end of this transfer, the received data byte is moved from the shifter into the receive data buffer and SPRF is set to indicate the data can be read by reading SPIxD. If another byte of data is waiting in the transmit buffer at the end of a transfer, it is moved into the shifter, SPTEF is set, and a new transfer is started.

Normally, SPI data is transferred most significant bit (MSB) first. If the least significant bit first enable (LSBFE) bit is set, SPI data is shifted LSB first.

When the SPI is configured as a slave, its \overline{SS} pin must be driven low before a transfer starts and \overline{SS} must stay low throughout the transfer. If a clock format where CPHA = 0 is selected, \overline{SS} must be driven to a logic 1 between successive transfers. If CPHA = 1, \overline{SS} may remain low between successive transfers. See [Section 13.5.1, "SPI Clock Formats"](#) for more details.

Because the transmitter and receiver are double buffered, a second byte, in addition to the byte currently being shifted out, can be queued into the transmit data buffer, and a previously received character can be in the receive data buffer while a new character is being shifted in. The SPTEF flag indicates when the transmit buffer has room for a new character. The SPRF flag indicates when a received character is available in the receive data buffer. The received character must be read out of the receive buffer (read SPIxD) before the next transfer is finished or a receive overrun error results.

In the case of a receive overrun, the new data is lost because the receive buffer still held the previous character and was not ready to accept the new data. There is no indication for such an overrun condition so the application system designer must ensure that previous data has been read from the receive buffer before a new transfer is initiated.

13.5.1 SPI Clock Formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a clock polarity (CPOL) bit and a clock phase (CPHA) control bit to select one of four clock formats for data transfers. CPOL selectively inserts an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data.

[Figure 13-10](#) shows the clock formats when CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCCK edge and bit 8 ending one-half SPSCCK cycle after the sixteenth SPSCCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCCK 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

Table 16-2. TPMxSC Field Descriptions (continued)

Field	Description
4–3 CLKS[B:A]	Clock source selects. As shown in Table 16-3, this 2-bit field is used to disable the TPM system or select one of three clock sources to drive the counter prescaler. The fixed system clock source is only meaningful in systems with a PLL-based or FLL-based system clock. When there is no PLL or FLL, the fixed-system clock source is the same as the bus rate clock. The external source is synchronized to the bus clock by TPM module, and the fixed system clock source (when a PLL or FLL is present) is synchronized to the bus clock by an on-chip synchronization circuit. When a PLL or FLL is present but not enabled, the fixed-system clock source is the same as the bus-rate clock.
2–0 PS[2:0]	Prescale factor select. This 3-bit field selects one of 8 division factors for the TPM clock input as shown in Table 16-4. This prescaler is located after any clock source synchronization or clock source selection so it affects the clock source selected to drive the TPM system. The new prescale factor will affect the clock source on the next system clock cycle after the new value is updated into the register bits.

Table 16-3. TPM-Clock-Source Selection

CLKSB:CLKSA	TPM Clock Source to Prescaler Input
00	No clock selected (TPM counter disable)
01	Bus rate clock
10	Fixed system clock
11	External source

Table 16-4. Prescale Factor Selection

PS2:PS1:PS0	TPM Clock Source Divided-by
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

16.3.2 TPM-Counter Registers (TPMxCNTH:TPMxCNTL)

The two read-only TPM counter registers contain the high and low bytes of the value in the TPM counter. Reading either byte (TPMxCNTH or TPMxCNTL) latches the contents of both bytes into a buffer where they remain latched until the other half is read. This allows coherent 16-bit reads in either big-endian or little-endian order which makes this more friendly to various compiler implementations. The coherency mechanism is automatically restarted by an MCU reset or any write to the timer status/control register (TPMxSC).

Table 17-1. BDC Command Summary

Command Mnemonic	Active BDM/ Non-intrusive	Coding Structure	Description
SYNC	Non-intrusive	n/a ¹	Request a timed reference pulse to determine target BDC communication speed
ACK_ENABLE	Non-intrusive	D5/d	Enable acknowledge protocol. Refer to Freescale document order no. HCS08RMv1/D.
ACK_DISABLE	Non-intrusive	D6/d	Disable acknowledge protocol. Refer to Freescale document order no. HCS08RMv1/D.
BACKGROUND	Non-intrusive	90/d	Enter active background mode if enabled (ignore if ENBDM bit equals 0)
READ_STATUS	Non-intrusive	E4/SS	Read BDC status from BDCSCR
WRITE_CONTROL	Non-intrusive	C4/CC	Write BDC controls in BDCSCR
READ_BYTE	Non-intrusive	E0/AAAA/d/RD	Read a byte from target memory
READ_BYTE_WS	Non-intrusive	E1/AAAA/d/SS/RD	Read a byte and report status
READ_LAST	Non-intrusive	E8/SS/RD	Re-read byte from address just read and report status
WRITE_BYTE	Non-intrusive	C0/AAAA/WD/d	Write a byte to target memory
WRITE_BYTE_WS	Non-intrusive	C1/AAAA/WD/d/SS	Write a byte and report status
READ_BKPT	Non-intrusive	E2/RBKP	Read BDCBKPT breakpoint register
WRITE_BKPT	Non-intrusive	C2/WBKP	Write BDCBKPT breakpoint register
GO	Active BDM	08/d	Go to execute the user application program starting at the address currently in the PC
TRACE1	Active BDM	10/d	Trace 1 user instruction at the address in the PC, then return to active background mode
TAGGO	Active BDM	18/d	Same as GO but enable external tagging (HCS08 devices have no external tagging pin)
READ_A	Active BDM	68/d/RD	Read accumulator (A)
READ_CCR	Active BDM	69/d/RD	Read condition code register (CCR)
READ_PC	Active BDM	6B/d/RD16	Read program counter (PC)
READ_HX	Active BDM	6C/d/RD16	Read H and X register pair (H:X)
READ_SP	Active BDM	6F/d/RD16	Read stack pointer (SP)
READ_NEXT	Active BDM	70/d/RD	Increment H:X by one then read memory byte located at H:X
READ_NEXT_WS	Active BDM	71/d/SS/RD	Increment H:X by one then read memory byte located at H:X. Report status and data.
WRITE_A	Active BDM	48/WD/d	Write accumulator (A)
WRITE_CCR	Active BDM	49/WD/d	Write condition code register (CCR)
WRITE_PC	Active BDM	4B/WD16/d	Write program counter (PC)
WRITE_HX	Active BDM	4C/WD16/d	Write H and X register pair (H:X)
WRITE_SP	Active BDM	4F/WD16/d	Write stack pointer (SP)
WRITE_NEXT	Active BDM	50/WD/d	Increment H:X by one, then write memory byte located at H:X
WRITE_NEXT_WS	Active BDM	51/WD/d/SS	Increment H:X by one, then write memory byte located at H:X. Also report status.

¹ The SYNC command is a special operation that does not have a command code.

maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (for instance, either V_{SS} or V_{DD}).

Table A-2. Absolute Maximum Ratings

Num	Rating	Symbol	Value	Unit
1	Supply voltage	V_{DD}	-0.3 to + 5.8	V
2	Input voltage	V_{In}	- 0.3 to $V_{DD} + 0.3$	V
3	Instantaneous maximum current Single pin limit (applies to all port pins) ^{1, 2, 3}	I_D	± 25	mA
4	Maximum current into V_{DD}	I_{DD}	120	mA
5	Storage temperature	T_{stg}	-55 to +150	°C

- ¹ Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive (V_{DD}) and negative (V_{SS}) clamp voltages, then use the larger of the two resistance values.
- ² All functional non-supply pins are internally clamped to V_{SS} and V_{DD} .
- ³ Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{In} > V_{DD}$) is greater than I_{DD} , the injection current may flow out of V_{DD} and could result in external power supply going out of regulation. Ensure external V_{DD} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power. Examples are: if no system clock is present, or if the clock rate is very low which would reduce overall power consumption.

A.4 Thermal Characteristics

This section provides information about operating temperature range, power dissipation, and package thermal resistance. Power dissipation on I/O pins is usually small compared to the power dissipation in on-chip logic and it is user-determined rather than being controlled by the MCU design. In order to take $P_{I/O}$ into account in power calculations, determine the difference between actual pin voltage and V_{SS} or V_{DD} and multiply by the pin current for each I/O pin. Except in cases of unusually high pin current (heavy loads), the difference between pin voltage and V_{SS} or V_{DD} will be very small.

Table A-3. Thermal Characteristics

Num	C	Rating	Symbol	Value	Unit	
1	D	Operating temperature range (packaged)	T_A	T_L to T_H	°C	
				P^1		-40 to 85
				V		-40 to 105
				M		-40 to 125
2	T	Maximum junction temperature	T_J	135		
3	D	Thermal resistance ^{2,3} Single-layer board		θ_{JA}	°C/W	
		100-pin LQFP	61			
		64-pin LQFP	67			
		48-pin LQFP	75			
4	D	Thermal resistance ^{2,3} Four-layer board		θ_{JA}	°C/W	
		100-pin LQFP	48			
		64-pin LQFP	49			
		48-pin LQFP	52			

¹ Freescale may eliminate a test insertion at a particular temperature from the production test flow once sufficient data has been collected and is approved.

² Junction temperature is a function of die size, on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, air flow, power dissipation of other components on the board, and board thermal resistance.

³ Junction to Ambient Natural Convection

The average chip-junction temperature (T_J) in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad \text{Eqn. A-1}$$

where:

T_A = Ambient temperature, °C

θ_{JA} = Package thermal resistance, junction-to-ambient, °C/W

$P_D = P_{int} + P_{I/O}$

$P_{int} = I_{DD} \times V_{DD}$, Watts — chip internal power

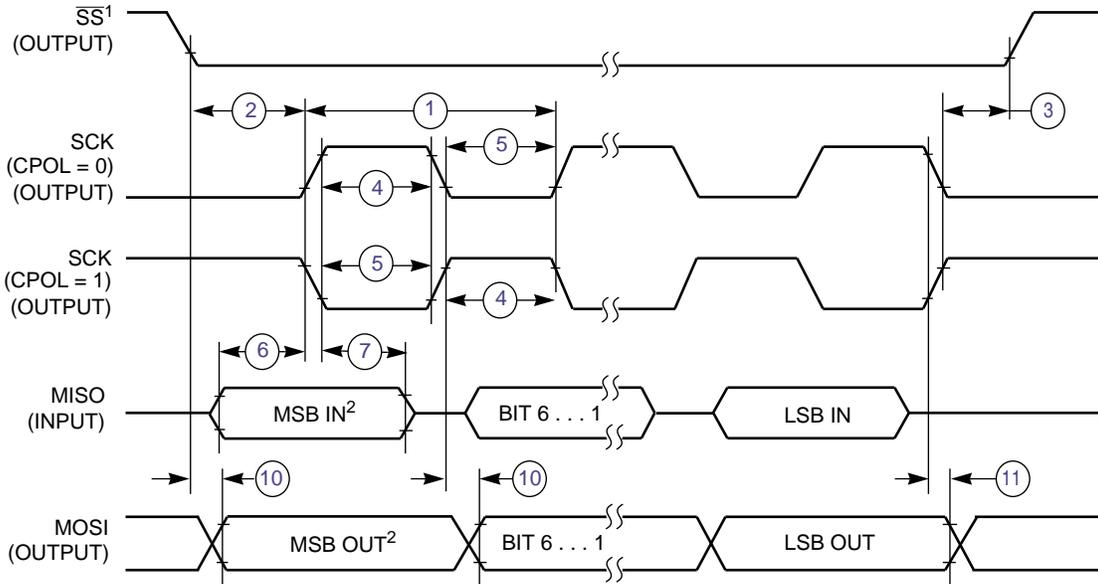
$P_{I/O}$ = Power dissipation on input and output pins — user determined

For most applications, $P_{I/O} \ll P_{int}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273^\circ\text{C}) \quad \text{Eqn. A-2}$$

Solving equations 1 and 2 for K gives:

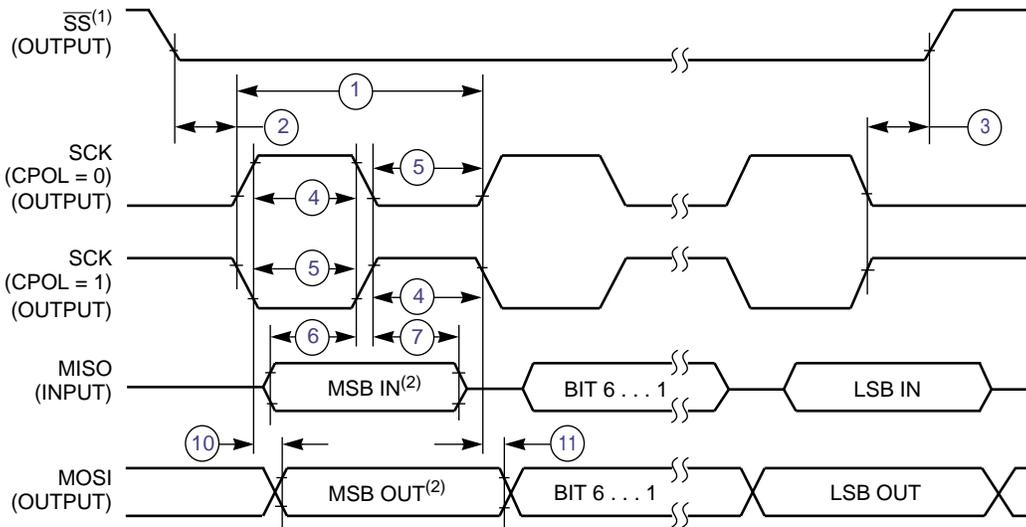
$$K = P_D \times (T_A + 273^\circ\text{C}) + \theta_{JA} \times (P_D)^2 \quad \text{Eqn. A-3}$$



NOTES:

1. \overline{SS} output mode (MODFEN = 1, SSOE = 1).
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure A-15. SPI Master Timing (CPHA = 0)



NOTES:

1. \overline{SS} output mode (MODFEN = 1, SSOE = 1).
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure A-16. SPI Master Timing (CPHA = 1)

Appendix B

Ordering Information and Mechanical Drawings

B.1 Ordering Information

This section contains ordering information for MC9S08DZ128 Series devices.

Example of the device numbering system:

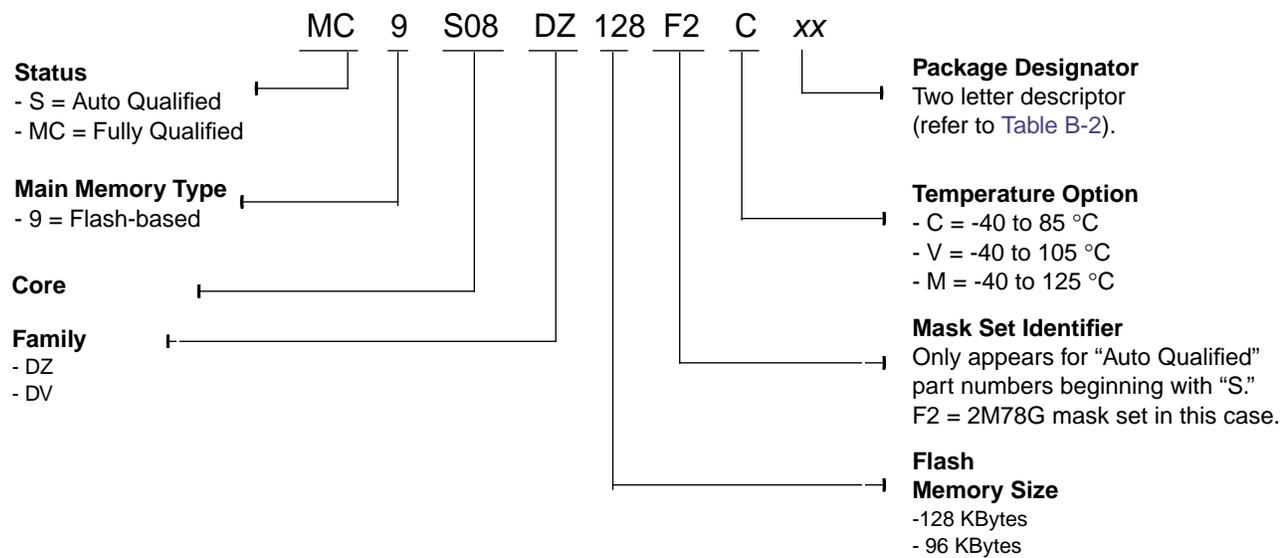


Figure 18-18. Device Numbering Scheme

B.1.1 MC9S08DZ128 Series Devices

Table B-1. Devices in the MC9S08DZ128 Series

Device Number	Memory			Available Packages ¹
	FLASH	RAM	EEPROM	
MC9S08DZ128	128K	8K	2K	100 LQFP 64 LQFP 48 LQFP
MC9S08DZ96	96K	6K	2K	
MC9S08DV128	128K	6K	—	
MC9S08DV96	96K	4K	—	

¹ See Table B-2 for package information.