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	53
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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08dv96clh

Chapter 1
Device Overview

MC9S08DZ128 Series devices are members of the low-cost, high-performance HCS08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced HCS08 core and are available with a variety of modules, memory sizes, memory types, and package types.

1.1 Devices in the MC9S08DZ128 Series

This data sheet covers members of the MC9S08DZ128 Series of MCUs:

- MC9S08DZ128
- MC9S08DZ96
- MC9S08DV128
- MC9S08DV96

Table 1-1 summarizes the feature set available in the MC9S08DZ128 Series.

Table 4-12 shows program and erase times. The bus clock frequency and FCDIV determine the frequency of FCLK (f_{FCLK}). The time for one cycle of FCLK is $t_{FCLK} = 1/f_{FCLK}$. The times are shown as a number of cycles of FCLK and as an absolute time for the case where $t_{FCLK} = 5\ \mu s$. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

Table 4-12. Program and Erase Times

Parameter	Cycles of FCLK	Time if FCLK = 200 kHz
Byte program	9	45 µs
Burst program	4	20 µs[1]
Sector erase	4000	20 ms
Mass erase	20,000	100 ms
Sector erase abort	4	20 µs[1]

[1] Excluding start/end overhead

4.6.3 Program and Erase Command Execution

The FCDIV register must be initialized after any reset and any error flag is cleared before beginning command execution. The command execution steps are:

1. Write a data value to an address in the FLASH or EEPROM array. The address and data information from this write is latched into the FLASH and EEPROM interface. This write is a required first step in any command sequence. For erase and blank check commands, the value of the data is not important. For sector erase commands, the address can be any address in the sector of FLASH or EEPROM to be erased. For mass erase and blank check commands, the address can be any address in the FLASH or EEPROM memory. FLASH and EEPROM erase independently of each other.

   NOTE

   Before programming a particular byte in the FLASH or EEPROM, the sector in which that particular byte resides must be erased by a mass or sector erase operation. Reprogramming bits in an already programmed byte without first performing an erase operation may disturb data stored in the FLASH or EEPROM memory.

2. Write the command code for the desired command to FCMD. The six valid commands are blank check (0x05), byte program (0x20), burst program (0x25), sector erase (0x40), mass erase (0x41), and sector erase abort (0x47). The command code is latched into the command buffer.
3. Write a 1 to the FCBEF bit in FSTAT to clear FCBEF and launch the command (including its address and data information).

   A partial command sequence can be aborted manually by writing a 0 to FCBEF any time after the write to the memory array and before writing the 1 that clears FCBEF and launches the complete command. Aborting a command in this way sets the FACCERR access error flag which must be cleared before starting a new command.

A strictly monitored procedure must be obeyed or the command will not be accepted. This minimizes the possibility of any unintended changes to the memory contents. The command complete flag (FCCF) indicates when a command is complete. The command sequence must be completed by clearing FCBEF to launch the command. Figure 4-11 is a flowchart for executing all of the commands except for burst programming and sector erase abort.

4. Wait until the FCCF bit in FSTAT is set. As soon as FCCF= 1, the operation has completed successfully.

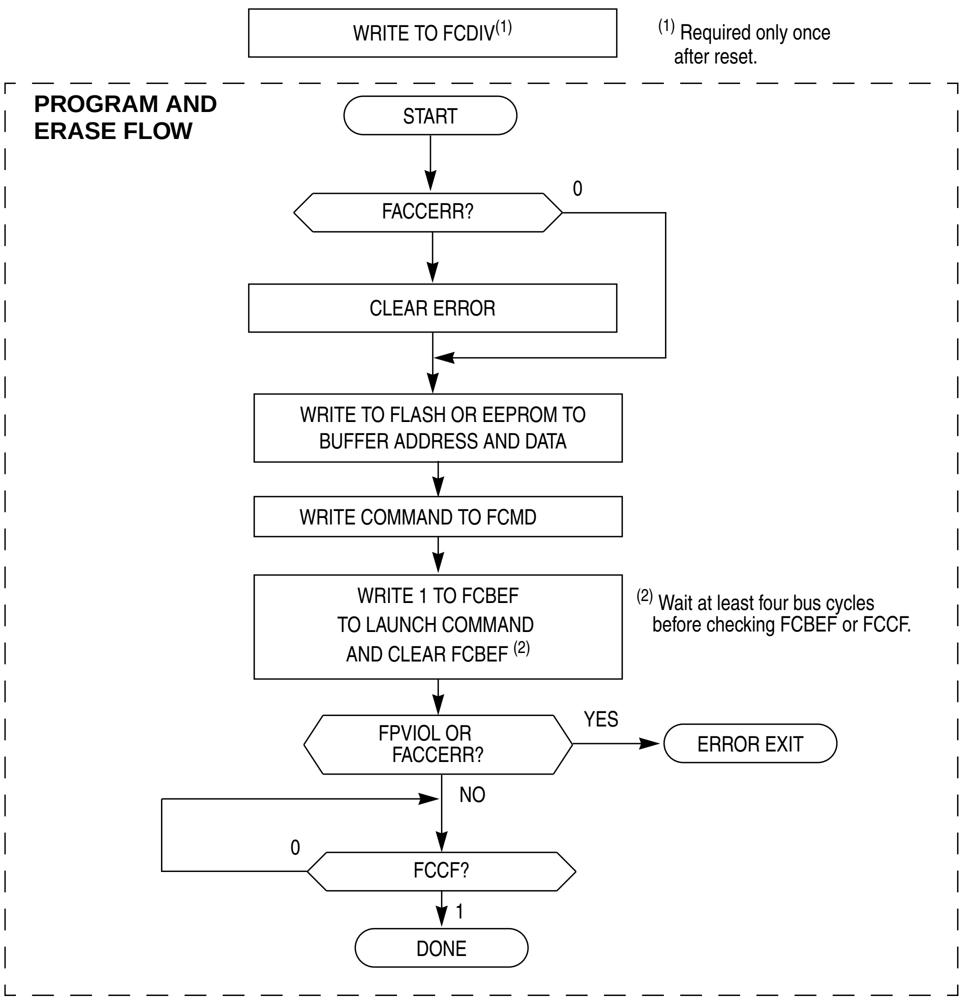

Figure 4-11. Program and Erase Flowchart

4.6.4 Burst Program Execution

The burst program command is used to program sequential bytes of data in less time than would be required using the standard program command. This is possible because the high voltage to the FLASH array does not need to be disabled between program operations. Ordinarily, when a program or erase command is issued, an internal charge pump associated with the FLASH memory must be enabled to supply high voltage to the array. Upon completion of the command, the charge pump is turned off. When

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.

	7	6	5	4	3	2	1	0
R	DIVLD	PRDIV8	DIV					
W	(unimplemented)							
Reset	0	0	0	0	0	0	0	0

(shaded) = Unimplemented or Reserved

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

6.5.4.7 Port D Interrupt Pin Select Register (PTDPS)

	7	6	5	4	3	2	1	0
R/W	PTDPS7	PTDPS6	PTDPS5	PTDPS4	PTDPS3	PTDPS2	PTDPS1	PTDPS0
Reset:	0	0	0	0	0	0	0	0

Figure 6-30. Port D Interrupt Pin Select Register (PTDPS)

Table 6-28. PTDPS Register Field Descriptions

Field	Description
7:0 PTDPS[7:0]	**Port D Interrupt Pin Selects** — Each of the PTDPSn bits enable the corresponding port D interrupt pin. 0 Pin not enabled as interrupt. 1 Pin enabled as interrupt.

6.5.4.8 Port D Interrupt Edge Select Register (PTDES)

	7	6	5	4	3	2	1	0
R/W	PTDES7	PTDES6	PTDES5	PTDES4	PTDES3	PTDES2	PTDES1	PTDES0
Reset:	0	0	0	0	0	0	0	0

Figure 6-31. Port D Edge Select Register (PTDES)

Table 6-29. PTDES Register Field Descriptions

Field	Description
7:0 PTDES[7:0]	**Port D Edge Selects** — Each of the PTDESn bits serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 A pull-up device is connected to the associated pin and detects falling edge/low level for interrupt generation. 1 A pull-down device is connected to the associated pin and detects rising edge/high level for interrupt generation.

6.5.8 Port H Registers

Port H is controlled by the registers listed below.

6.5.8.1 Port H Data Register (PTHD)

	7	6	5	4	3	2	1	0
R W	PTHD7	PTHD6	PTHD5	PTHD4	PTHD3	PTHD2	PTHD1	PTHD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-47. Port H Data Register (PTHD)

Table 6-45. PTHD Register Field Descriptions

Field	Description
7:0 PTHD[7:0]	**Port H Data Register Bits** — For port H pins that are inputs, reads return the logic level on the pin. For port H 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 H pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTHD 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.

6.5.8.2 Port H Data Direction Register (PTHDD)

	7	6	5	4	3	2	1	0
R W	PTHDD7	PTHDD6	PTHDD5	PTHDD4	PTHDD3	PTHDD2	PTHDD1	PTHDD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-48. Port H Data Direction Register (PTHDD)

Table 6-46. PTHDD Register Field Descriptions

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

6.5.8.5 Port H Drive Strength Selection Register (PTHDS)

	7	6	5	4	3	2	1	0
R W	PTHDS7	PTHDS6	PTHDS5	PTHDS4	PTHDS3	PTHDS2	PTHDS1	PTHDS0
Reset:	0	0	0	0	0	0	0	0

Figure 6-51. Drive Strength Selection for Port H Register (PTHDS)

Table 6-49. PTHDS Register Field Descriptions

Field	Description
7:0 PTHDS[7:0]	**Output Drive Strength Selection for Port H Bits** — Each of these control bits selects between low and high output drive for the associated PTH pin. For port H pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port H bit n. 1 High output drive strength selected for port H bit n.

6.5.10.5 Port K Drive Strength Selection Register (PTKDS)

	7	6	5	4	3	2	1	0
R W	PTKDS7	PTKDS6	PTKDS5	PTKDS4	PTKDS3	PTKDS2	PTKDS1	PTKDS0
Reset:	0	0	0	0	0	0	0	0

Figure 6-64. Drive Strength Selection for Port K Register (PTKDS)

Table 6-62. PTKDS Register Field Descriptions

Field	Description
7:0 PTKDS[7:0]	**Output Drive Strength Selection for Port K Bits** — Each of these control bits selects between low and high output drive for the associated PTK pin. For port K pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port K bit n. 1 High output drive strength selected for port K bit n.

8.4.1.6 PLL Bypassed External (PBE)

In PLL bypassed external (PBE) mode, the MCGOUT clock is derived from the external reference clock and the PLL is operational but its output clock is not used. This mode is useful to allow the PLL to acquire its target frequency while the MCGOUT clock is driven from the external reference clock.

The PLL bypassed external mode is entered when all the following conditions occur:

- CLKS bits are written to 10
- IREFS bit is written to 0
- PLLS bit is written to 1
- RDIV bits are written to divide reference clock to be within the range of 1 MHz to 2 MHz
- LP bit is written to 0

In PLL bypassed external mode, the MCGOUT clock is derived from the external reference clock. The external reference clock which is enabled can be an external crystal/resonator or it can be another external clock source. The PLL clock frequency locks to a multiplication factor, as selected by the VDIV bits, times the external reference frequency, as selected by the RDIV, RANGE and DIV32 bits. If BDM is enabled then the MCGLCLK is derived from the DCO (open-loop mode) divided by two. If BDM is not enabled then the FLL is disabled in a low power state.

In this mode, the DRST bit reads 0 regardless of whether the DRS bit is set to 1 or 0.

8.4.1.7 Bypassed Low Power Internal (BLPI)

The bypassed low power internal (BLPI) mode is entered when all the following conditions occur:

- CLKS bits are written to 01
- IREFS bit is written to 1
- PLLS bit is written to 0
- LP bit is written to 1
- BDM mode is not active

In bypassed low power internal mode, the MCGOUT clock is derived from the internal reference clock.

The PLL and the FLL are disabled at all times in BLPI mode and the MCGLCLK will not be available for BDC communications If the BDM becomes active the mode will switch to FLL bypassed internal (FBI) mode.

In this mode, the DRST bit reads 0 regardless of whether the DRS bit is set to 1 or 0.

8.4.1.8 Bypassed Low Power External (BLPE)

The bypassed low power external (BLPE) mode is entered when all the following conditions occur:

- CLKS bits are written to 10
- IREFS bit is written to 0
- PLLS bit is written to 0 or 1
- LP bit is written to 1

[1] R is the reference divider selected by the RDIV bits, B is the bus frequency divider selected by the BDIV bits, F is the FLL factor selected by the DRS and DMX32 bits, and M is the multiplier selected by the VDIV bits.

This section will include 3 mode switching examples using an 8 MHz external crystal. If using an external clock source less than 1 MHz, the MCG should not be configured for any of the PLL modes (PEE and PBE).

8.5.3.1 Example # 1: Moving from FEI to PEE Mode: External Crystal = 8 MHz, Bus Frequency = 16 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE mode until the 8 MHz crystal reference frequency is set to achieve a bus frequency of 16 MHz. Because the MCG is in FEI mode out of reset, this example also shows how to initialize the MCG for PEE mode out of reset. First, the code sequence will be described. Then a flowchart will be included which illustrates the sequence.

1. First, FEI must transition to FBE mode:

   a) MCGC2 = 0x36 (%00110110)
      – BDIV (bits 7 and 6) set to %00, or divide-by-1
      – RANGE (bit 5) set to 1 because the frequency of 8 MHz is within the high frequency range
      – HGO (bit 4) set to 1 to configure external oscillator for high gain operation
      – EREFS (bit 2) set to 1, because a crystal is being used
      – ERCLKEN (bit 1) set to 1 to ensure the external reference clock is active

   b) Loop until OSCINIT (bit 1) in MCGSC is 1, indicating the crystal selected by the EREFS bit has been initialized.

   c) Because RANGE = 1, set DIV32 (bit 4) in MCGC3 to allow access to the proper RDIV bits while in an FLL external mode.

   d) MCGC1 = 0x98 (%10011000)
      – CLKS (bits 7 and 6) set to %10 in order to select external reference clock as system clock source
      – RDIV (bits 5-3) set to %011, or divide-by-256 because 8MHz / 256 = 31.25 kHz which is in the 31.25 kHz to 39.0625 kHz range required by the FLL
      – IREFS (bit 2) cleared to 0, selecting the external reference clock

   e) Loop until IREFST (bit 4) in MCGSC is 0, indicating the external reference is the current source for the reference clock

   f) Loop until CLKST (bits 3 and 2) in MCGSC is %10, indicating that the external reference clock is selected to feed MCGOUT

2. Then, FBE must transition either directly to PBE mode or first through BLPE mode and then to PBE mode:

10.6.1.3 Analog Input Pins

The external analog inputs are typically shared with digital I/O pins on MCU devices. The pin I/O control is disabled by setting the appropriate control bit in one of the pin control registers. Conversions can be performed on inputs without the associated pin control register bit set. It is recommended that the pin control register bit always be set when using a pin as an analog input. This avoids problems with contention because the output buffer is in its high impedance state and the pullup is disabled. Also, the input buffer draws DC current when its input is not at V_{DD} or V_{SS}. Setting the pin control register bits for all pins used as analog inputs should be done to achieve lowest operating current.

Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 μF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used they must be placed as near as possible to the package pins and be referenced to V_{SSA}.

For proper conversion, the input voltage must fall between V_{REFH} and V_{REFL}. If the input is equal to or exceeds V_{REFH}, the converter circuit converts the signal to 0xFFF (full scale 12-bit representation), 0x3FF (full scale 10-bit representation) or 0xFF (full scale 8-bit representation). If the input is equal to or less than V_{REFL}, the converter circuit converts it to 0x000. Input voltages between V_{REFH} and V_{REFL} are straight-line linear conversions. There is a brief current associated with V_{REFL} when the sampling capacitor is charging. The input is sampled for 3.5 cycles of the ADCK source when ADLSMP is low, or 23.5 cycles when ADLSMP is high.

For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins should not be transitioning during conversions.

10.6.2 Sources of Error

Several sources of error exist for A/D conversions. These are discussed in the following sections.

10.6.2.1 Sampling Error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy. Given the maximum input resistance of approximately 7kΩ and input capacitance of approximately 5.5 pF, sampling to within 1/4LSB (at 12-bit resolution) can be achieved within the minimum sample window (3.5 cycles @ 8 MHz maximum ADCK frequency) provided the resistance of the external analog source (R_{AS}) is kept below 2 kΩ.

Higher source resistances or higher-accuracy sampling is possible by setting ADLSMP (to increase the sample window to 23.5 cycles) or decreasing ADCK frequency to increase sample time.

10.6.2.2 Pin Leakage Error

Leakage on the I/O pins can cause conversion error if the external analog source resistance (R_{AS}) is high. If this error cannot be tolerated by the application, keep R_{AS} lower than $V_{DDAD} / (2^N * I_{LEAK})$ for less than 1/4LSB leakage error (N = 8 in 8-bit, 10 in 10-bit or 12 in 12-bit mode).

12.5.2.1 Message Transmit Background

Modern application layer software is built upon two fundamental assumptions:

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactiveness requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the "local priority" concept described in Section 12.5.2.2, "Transmit Structures."

12.5.2.2 Transmit Structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in Figure 12-38.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see Section 12.4, "Programmer's Model of Message Storage"). An additional Section 12.4.5, "Transmit Buffer Priority Register (TBPR) contains an 8-bit local priority field (PRIO) (see Section 12.4.5, "Transmit Buffer Priority Register (TBPR)"). The remaining two bytes are used for time stamping of a message, if required (see Section 12.4.6, "Time Stamp Register (TSRH–TSRL)").

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see Section 12.3.6, "MSCAN Transmitter Flag Register (CANTFLG)"). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see Section 12.3.10, "MSCAN Transmit Buffer Selection Register (CANTBSEL)"). This makes the respective buffer accessible within the CANTXFG address space (see Section 12.4, "Programmer's Model of Message Storage"). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler software simpler because only one address area is applicable for the transmit process, and the required address space is minimized.

The CPU then stores the identifier, the control bits, and the data content into one of the transmit buffers. Finally, the buffer is flagged as ready for transmission by clearing the associated TXE flag.

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 messages[1]. 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

Chapter 13
Serial Peripheral Interface (S08SPIV3)

13.1 Introduction

The serial peripheral interface (SPI) module provides for full-duplex, synchronous, serial communication between the MCU and peripheral devices. These peripheral devices can include other microcontrollers, analog-to-digital converters, shift registers, sensors, memories, etc.

The SPI runs at a baud rate up to the bus clock divided by two in master mode and up to the bus clock divided by four in slave mode. Software can poll the status flags, or SPI operation can be interrupt driven.

All MC9S08DZ128 Series MCUs in the 100-pin package have two SPIs; devices in the 64-pin and 48-pin packages have one SPI.

Table 13-3. SPIxC2 Register Field Descriptions

Field	Description
4 MODFEN	**Master Mode-Fault Function Enable** — When the SPI is configured for slave mode, this bit has no meaning or effect. (The $\overline{SS}$ pin is the slave select input.) In master mode, this bit determines how the $\overline{SS}$ pin is used (refer to Table 13-2 for more details). 0 Mode fault function disabled, master $\overline{SS}$ pin reverts to general-purpose I/O not controlled by SPI 1 Mode fault function enabled, master $\overline{SS}$ pin acts as the mode fault input or the slave select output
3 BIDIROE	**Bidirectional Mode Output Enable** — When bidirectional mode is enabled by SPI pin control 0 (SPC0) = 1, BIDIROE determines whether the SPI data output driver is enabled to the single bidirectional SPI I/O pin. Depending on whether the SPI is configured as a master or a slave, it uses either the MOSI (MOMI) or MISO (SISO) pin, respectively, as the single SPI data I/O pin. When SPC0 = 0, BIDIROE has no meaning or effect. 0 Output driver disabled so SPI data I/O pin acts as an input 1 SPI I/O pin enabled as an output
1 SPISWAI	**SPI Stop in Wait Mode** 0 SPI clocks continue to operate in wait mode 1 SPI clocks stop when the MCU enters wait mode
0 SPC0	**SPI Pin Control 0** — The SPC0 bit chooses single-wire bidirectional mode. If MSTR = 0 (slave mode), the SPI uses the MISO (SISO) pin for bidirectional SPI data transfers. If MSTR = 1 (master mode), the SPI uses the MOSI (MOMI) pin for bidirectional SPI data transfers. When SPC0 = 1, BIDIROE is used to enable or disable the output driver for the single bidirectional SPI I/O pin. 0 SPI uses separate pins for data input and data output 1 SPI configured for single-wire bidirectional operation

13.4.3 SPI Baud Rate Register (SPIxBR)

This register is used to set the prescaler and bit rate divisor for an SPI master. This register may be read or written at any time.

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 13-7. SPI Baud Rate Register (SPIxBR)

Table 13-4. SPIxBR Register Field Descriptions

Field	Description
6:4 SPPR[2:0]	**SPI Baud Rate Prescale Divisor** — This 3-bit field selects one of eight divisors for the SPI baud rate prescaler as shown in Table 13-5. The input to this prescaler is the bus rate clock (BUSCLK). The output of this prescaler drives the input of the SPI baud rate divider (see Figure 13-4).
2:0 SPR[2:0]	**SPI Baud Rate Divisor** — This 3-bit field selects one of eight divisors for the SPI baud rate divider as shown in Table 13-6. The input to this divider comes from the SPI baud rate prescaler (see Figure 13-4). The output of this divider is the SPI bit rate clock for master mode.

Input capture, output compare, and edge-aligned PWM functions do not make sense when the counter is operating in up/down counting mode so this implies that all active channels within a TPM must be used in CPWM mode when CPWMS=1.

The TPM may be used in an 8-bit MCU. The settings in the timer channel registers are buffered to ensure coherent 16-bit updates and to avoid unexpected PWM pulse widths. Writes to any of the registers TPMxMODH, TPMxMODL, TPMxCnVH, and TPMxCnVL, actually write to buffer registers.

In center-aligned PWM mode, the TPMxCnVH:L registers are updated with the value of their write buffer according to the value of CLKSB:CLKSA bits, so:

- If (CLKSB:CLKSA = 0:0), the registers are updated when the second byte is written
- If (CLKSB:CLKSA not = 0:0), the registers are updated after the both bytes were written, and the TPM counter changes from (TPMxMODH:TPMxMODL - 1) to (TPMxMODH:TPMxMODL). If the TPM counter is a free-running counter, the update is made when the TPM counter changes from 0xFFFE to 0xFFFF.

When TPMxCNTH:TPMxCNTL=TPMxMODH:TPMxMODL, the TPM can optionally generate a TOF interrupt (at the end of this count).

Writing to TPMxSC cancels any values written to TPMxMODH and/or TPMxMODL and resets the coherency mechanism for the modulo registers. Writing to TPMxCnSC cancels any values written to the channel value registers and resets the coherency mechanism for TPMxCnVH:TPMxCnVL.

16.5 Reset Overview

16.5.1 General

The TPM is reset whenever any MCU reset occurs.

16.5.2 Description of Reset Operation

Reset clears the TPMxSC register which disables clocks to the TPM and disables timer overflow interrupts (TOIE=0). CPWMS, MSnB, MSnA, ELSnB, and ELSnA are all cleared which configures all TPM channels for input-capture operation with the associated pins disconnected from I/O pin logic (so all MCU pins related to the TPM revert to general purpose I/O pins).

16.6 Interrupts

16.6.1 General

The TPM generates an optional interrupt for the main counter overflow and an interrupt for each channel. The meaning of channel interrupts depends on each channel's mode of operation. If the channel is configured for input capture, the interrupt flag is set each time the selected input capture edge is recognized. If the channel is configured for output compare or PWM modes, the interrupt flag is set each time the main timer counter matches the value in the 16-bit channel value register.

18.3.3.16 Debug Count Status Register (DBGCNT)

Module Base + 0x000F

	7	6	5	4	3	2	1	0
R	0	0	0	0	CNT			
W								
POR or non-end-run	0	0	0	0	0	0	0	0
Reset end-run[1]	0	0	0	0	U	U	U	U

= Unimplemented or Reserved

Figure 18-17. Debug Count Status Register (DBGCNT)

[1] In the case of an end-trace to reset where DBGEN=1 and BEGIN=0, the CNT[3:0] bits do not change after reset.

Table 18-19. DBGS Field Descriptions

Field	Description
3–0 CNT	**FIFO Valid Count Bits** — The CNT bits indicate the amount of valid data stored in the FIFO. Table 18-20 shows the correlation between the CNT bits and the amount of valid data in FIFO. The CNT will stop after a count to eight even if more data is being stored in the FIFO. The CNT bits are cleared when the DBG module is armed, and the count is incremented each time a new word is captured into the FIFO. The host development system is responsible for checking the value in CNT[3:0] and reading the correct number of words from the FIFO because the count does not decrement as data is read out of the FIFO at the end of a trace run.

Table 18-20. CNT Bits

CNT Value	Meaning
0000	No data valid
0001	1 word valid
0010	2 words valid
0011	3 words valid
0100	4 words valid
0101	5 words valid
0110	6 words valid
0111	7 words valid
1000	8 words valid

Table A-12. MCG Frequency Specifications (Temperature Range = –40 to 125°C Ambient) (continued)

Num	C	Rating	Symbol	Min	Typical	Max	Unit
15	D	PLL reference frequency range	f_{pll_ref}	1.0	—	2.0	MHz
16	T	RMS frequency variation of a single clock cycle measured 2 ms after reference edge.[6]	$f_{pll\_cycjit\_2ms}$	—	0.590[5]	—	$\%f_{pll}$
17	T	Maximum frequency variation averaged over 2 ms window.	$f_{pll\_maxjit\_2ms}$	—	0.001	—	$\%f_{pll}$
18	T	RMS frequency variation of a single clock cycle measured 625 ns after reference edge.[7]	$f_{pll\_cycjit\_625ns}$	—	0.566[5]	—	$\%f_{pll}$
19	T	Maximum frequency variation averaged over 625 ns window.	$f_{pll\_maxjit\_625ns}$	—	0.113	—	$\%f_{pll}$
20	D	Lock entry frequency tolerance [8]	D_{lock}	± 1.49	—	± 2.98	%
21	D	Lock exit frequency tolerance [9]	D_{unl}	± 4.47	—	± 5.97	%
22	D	Lock time - FLL	t_{fll_lock}	—	—	$t_{fll\_acquire}$+ 1075(1/f_{int_t})	s

1 This applies when TRIM register at value (0x80) and FTRIM control bit at value (0x0). These values load when in BDM modes.

2 The resulting bus clock frequency should not exceed the maximum specified bus clock frequency of the device.

3 This specification applies to any time the FLL reference source or reference divider is changed, trim value is changed, DMX32 bit is changed, DRS bit is changed, or changing from FLL disabled (BLPE, BLPI) to FLL enabled (FEI, FEE, FBE, FBI). If a crystal/resonator is being used as the reference, this specification assumes it is already running.

4 This specification applies to any time the PLL VCO divider or reference divider is changed, or changing from PLL disabled (BLPE, BLPI) to PLL enabled (PBE, PEE). If a crystal/resonator is being used as the reference, this specification assumes it is already running.

5 Jitter is the average deviation from the programmed frequency measured over the specified interval at maximum f_{BUS}. Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the FLL circuitry via V_{DD} and V_{SS} and variation in crystal oscillator frequency increase the C_{Jitter} percentage for a given interval. These jitter measurements are based upon a 40 MHz MCGOUT clock frequency.

6 In some specifications, this value is described as, “Long term accuracy of PLL output clock (averaged over 2 ms)” with symbol “$f_{pll\_jitter\_2ms}$.” The parameter is unchanged, but the description has been changed for clarification purposes.

7 In some specifications, this value is described as “Jitter of PLL output clock measured over 625 ns” with symbol “$f_{pll\_jitter\_625ns}$.” The parameter is unchanged, but the description has been changed for clarification purposes.

8 Below D_{lock} minimum, the MCG is guaranteed to enter lock. Above D_{lock} maximum, the MCG will not enter lock. But if the MCG is already in lock, then the MCG may stay in lock.

9 Below D_{unl} minimum, the MCG will not exit lock if already in lock. Above D_{unl} maximum, the MCG is guaranteed to exit lock.

A.13 FLASH and EEPROM

This section provides details about program/erase times and program-erase endurance for the FLASH and EEPROM memory.

Program and erase operations do not require any special power sources other than the normal V_{DD} supply. For more detailed information about program/erase operations, see Chapter 4, “Memory.”

NOTE

All values shown in Table A-17 are preliminary and subject to further characterization.

Table A-17. FLASH and EEPROM Characteristics

Num	C	Rating	Symbol	Min	Typical	Max	Unit
1	—	Supply voltage for program/erase	$V_{prog/erase}$	2.7		5.5	V
2	—	Supply voltage for read operation	V_{Read}	2.7		5.5	V
3	—	Internal FCLK frequency[1]	f_{FCLK}	150		200	kHz
4	—	Internal FCLK period (1/FCLK)	t_{Fcyc}	5		6.67	μs
5	—	Byte program time (random location)[(2)]	t_{prog}		9		t_{Fcyc}
6	—	Byte program time (burst mode)[(2)]	t_{Burst}		4		t_{Fcyc}
7	—	Page erase time[2]	t_{Page}		4000		t_{Fcyc}
8	—	Mass erase time[(2)]	t_{Mass}		20,000		t_{Fcyc}
9	C	FLASH Program/erase endurance[3] T_L to T_H = –40°C to + 125°C T = 25°C	n_{FLPE}	10,000 	 100,000	— —	cycles
10	C	EEPROM Program/erase endurance[3] T_L to T_H = –40°C to + 0°C T_L to T_H = 0°C to + 125°C T = 25°C	n_{EEPE}	10,000 50,000 	 300,000	— — —	cycles
11	C	Data retention[4]	t_{D_ret}	15	100	—	years

1 The frequency of this clock is controlled by a software setting.

2 These values are hardware state machine controlled. User code does not need to count cycles. This information supplied for calculating approximate time to program and erase.

3 **Typical endurance** for FLASH and EEPROM is based on the intrinsic bitcell performance. For additional information on how Freescale Semiconductor defines typical endurance, please refer to Engineering Bulletin EB619, *Typical Endurance for Nonvolatile Memory.*

4 **Typical data retention** values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale Semiconductor defines typical data retention, please refer to Engineering Bulletin EB618, *Typical Data Retention for Nonvolatile Memory.*

A.14 EMC Performance

Electromagnetic compatibility (EMC) performance is highly dependant on the environment in which the MCU resides. Board design and layout, circuit topology choices, location and characteristics of external components as well as MCU software operation all play a significant role in EMC performance. The system designer should consult Freescale applications notes such as AN2321, AN1050, AN1263, AN2764, and AN1259 for advice and guidance specifically targeted at optimizing EMC performance.

A.14.1 Radiated Emissions

Microcontroller radiated RF emissions are measured from 150 kHz to 1 GHz using the TEM/GTEM Cell method in accordance with the IEC 61967-2 and SAE J1752/3 standards. The measurement is performed with the microcontroller installed on a custom EMC evaluation board while running specialized EMC test software. The radiated emissions from the microcontroller are measured in a TEM cell in two package orientations (North and East). For more detailed information concerning the evaluation results, conditions and setup, please refer to the EMC Evaluation Report for this device.

The maximum radiated RF emissions of the tested configuration in all orientations are less than or equal to the reported emissions levels.

Table A-18. Radiated Emissions

Parameter	Symbol	Conditions	Frequency	f_{osc}/f_{CPU}	Level[1] (Max)	Unit
Radiated emissions, electric field	V_{RE_TEM}	V_{DD} = 5V T_A = +25°C 100 LQFP	0.15 – 50 MHz	16 MHz Crystal 20 MHz Bus	14	dBμV
			50 – 150 MHz		21	
			150 – 500 MHz		6	
			500 – 1000 MHz		-5	
			IEC Level		L	—
			SAE Level		3	—

[1] Data based on qualification test results.