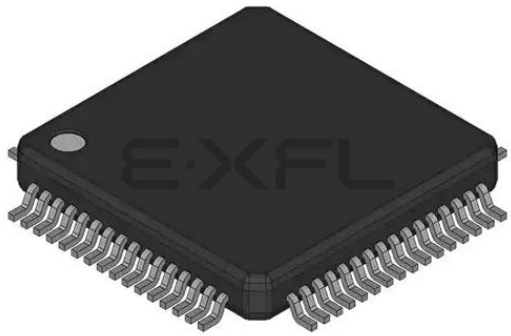


Welcome to E-XFL.COM

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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08dv128mlh

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.

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.

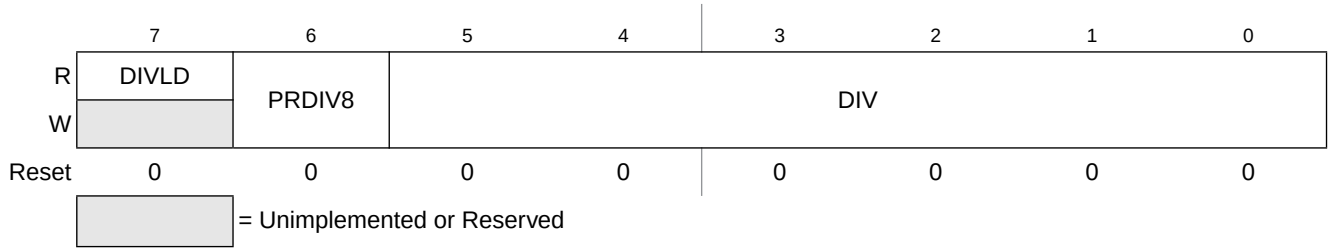


Figure 4-14. FLASH and EEPROM Clock Divider Register (FCDIV)

Table 4-15. FOPT Register Field Descriptions

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

Table 4-16. Security States[1]

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

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

4.6.11.3 FLASH and EEPROM Configuration Register (FCNFG)

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

= Unimplemented or Reserved

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

Table 4-17. FCNFG Register Field Descriptions

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

5.5.1 Interrupt Stack Frame

Figure 5-1 shows the contents and organization of a stack frame. Before the interrupt, the stack pointer (SP) points at the next available byte location on the stack. The current values of CPU registers are stored on the stack starting with the low-order byte of the program counter (PCL) and ending with the CCR. After stacking, the SP points at the next available location on the stack which is the address that is one less than the address where the CCR was saved. The PC value that is stacked is the address of the instruction in the main program that would have executed next if the interrupt had not occurred.

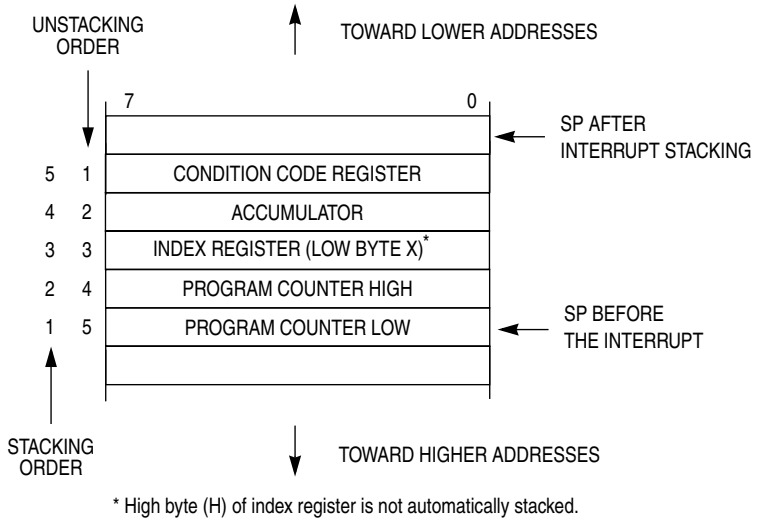


Figure 5-1. Interrupt Stack Frame

When an RTI instruction is executed, these values are recovered from the stack in reverse order. As part of the RTI sequence, the CPU fills the instruction pipeline by reading three bytes of program information, starting from the PC address recovered from the stack.

The status flag corresponding to the interrupt source must be acknowledged (cleared) before returning from the ISR. Typically, the flag is cleared at the beginning of the ISR so that if another interrupt is generated by this same source, it will be registered so it can be serviced after completion of the current ISR.

5.5.2 External Interrupt Request (IRQ) Pin

External interrupts are managed by the IRQ status and control register, IRQSC. When the IRQ function is enabled, synchronous logic monitors the pin for edge-only or edge-and-level events. When the MCU is in stop mode and system clocks are shut down, a separate asynchronous path is used so the IRQ (if enabled) can wake the MCU.

5.5.2.1 Pin Configuration Options

The IRQ pin enable (IRQPE) control bit in IRQSC must be 1 in order for the IRQ pin to act as the interrupt request (IRQ) input. As an IRQ input, the user can choose the polarity of edges or levels detected (IRQEDG), whether the pin detects edges-only or edges and levels (IRQMOD), and whether an event causes an interrupt or only sets the IRQF flag which can be polled by software.

5.8.2 System Reset Status Register (SRS)

This high page register includes read-only status flags to indicate the source of the most recent reset. When a debug host forces reset by writing 1 to BDFR in the SBDFR register, none of the status bits in SRS will be set. Writing any value to this register address causes a COP reset when the COP is enabled except the values 0x55 and 0xAA. Writing a 0x55-0xAA sequence to this address clears the COP watchdog timer without affecting the contents of this register. The reset state of these bits depends on what caused the MCU to reset.

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	ILAD	LOC	LVD	0
W	Writing 0x55, 0xAA to SRS address clears COP watchdog timer.							
POR:	1	0	0	0	0	0	1	0
LVD:	0	0	0	0	0	0	1	0
Any other reset:	0	Note(1)	Note(1)	Note(1)	Note(1)	0	0	0

1 Any of these reset sources that are active at the time of reset entry will cause the corresponding bit(s) to be set; bits corresponding to sources that are not active at the time of reset entry will be cleared.

Figure 5-3. System Reset Status (SRS)

Table 5-3. SRS Register Field Descriptions

Field	Description
7 POR	**Power-On Reset** — Reset was caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR. 1 POR caused reset.
6 PIN	**External Reset Pin** — Reset was caused by an active-low level on the external reset pin. 0 Reset not caused by external reset pin. 1 Reset came from external reset pin.
5 COP	**Computer Operating Properly (COP) Watchdog** — Reset was caused by the COP watchdog timer timing out. This reset source can be blocked by COPT bits = 0:0. 0 Reset not caused by COP timeout. 1 Reset caused by COP timeout.
4 ILOP	**Illegal Opcode** — Reset was caused by an attempt to execute an unimplemented or illegal opcode. The STOP instruction is considered illegal if stop is disabled by STOPE = 0 in the SOPT register. The BGND instruction is considered illegal if active background mode is disabled by ENBDM = 0 in the BDCSC register. 0 Reset not caused by an illegal opcode. 1 Reset caused by an illegal opcode.
3 ILAD	**Illegal Address** — Reset was caused by an attempt to access either data or an instruction at an unimplemented memory address. 0 Reset not caused by an illegal address. 1 Reset caused by an illegal address.

Semiconductor-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

6.5.1 Port A Registers

Port A is controlled by the registers listed below.

6.5.1.1 Port A Data Register (PTAD)

	7	6	5	4	3	2	1	0
R W	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-3. Port A Data Register (PTAD)

Table 6-1. PTAD Register Field Descriptions

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

6.5.1.2 Port A Data Direction Register (PTADD)

	7	6	5	4	3	2	1	0
R W	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-4. Port A Data Direction Register (PTADD)

Table 6-2. PTADD Register Field Descriptions

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

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.

7.3 Addressing Modes

Addressing modes define the way the CPU accesses operands and data. In the HCS08, memory, status and control registers, and input/output (I/O) ports share a single 64-Kbyte CPU address space. This arrangement means that the same instructions that access variables in RAM can also be used to access I/O and control registers or nonvolatile program space.

NOTE

For more information about extended addressing modes, see the Memory Management Unit section in the Memory chapter.

MCU derivatives with more than 64-Kbytes of memory also include a memory management unit (MMU) to support extended memory space. A PPAGE register is used to manage 16-Kbyte pages of memory which can be accessed by the CPU through a 16-Kbyte window from 0x8000 through 0xBFFF. The CPU includes two special instructions (CALL and RTC). CALL operates like the JSR instruction except that CALL saves the current PPAGE value on the stack and provides a new PPAGE value for the destination. RTC works like the RTS instruction except RTC restores the old PPAGE value in addition to the PC during the return from the called routine. The MMU also includes a linear address pointer register and data access registers so that the extended memory space operates as if it was a single linear block of memory. For additional information about the MMU, refer to the Memory chapter of this data sheet.

Some instructions use more than one addressing mode. For instance, move instructions use one addressing mode to specify the source operand and a second addressing mode to specify the destination address. Instructions such as BRCLR, BRSET, CBEQ, and DBNZ use one addressing mode to specify the location of an operand for a test and then use relative addressing mode to specify the branch destination address when the tested condition is true. For BRCLR, BRSET, CBEQ, and DBNZ, the addressing mode listed in the instruction set tables is the addressing mode needed to access the operand to be tested, and relative addressing mode is implied for the branch destination.

7.3.1 Inherent Addressing Mode (INH)

In this addressing mode, operands needed to complete the instruction (if any) are located within CPU registers so the CPU does not need to access memory to get any operands.

7.3.2 Relative Addressing Mode (REL)

Relative addressing mode is used to specify the destination location for branch instructions. A signed 8-bit offset value is located in the memory location immediately following the opcode. During execution, if the branch condition is true, the signed offset is sign-extended to a 16-bit value and is added to the current contents of the program counter, which causes program execution to continue at the branch destination address.

7.3.3 Immediate Addressing Mode (IMM)

In immediate addressing mode, the operand needed to complete the instruction is included in the object code immediately following the instruction opcode in memory. In the case of a 16-bit immediate operand,

reference can achieve a high-range maximum DCO output of 39.85 MHz with a multiplier of 1216. When the DRS bit is clear, the 32.768 kHz reference can achieve a mid-range maximum DCO output of 19.92 MHz with a multiplier of 608.

In FBI and FEI modes, setting the DMX32 bit is not recommended. If the internal reference is trimmed to a frequency above 32.768 kHz, the greater FLL multiplication factor could potentially push the microcontroller system clock out of specification and damage the part.

8.5.3 MCG Mode Switching

When switching between operational modes of the MCG, certain configuration bits must be changed in order to properly move from one mode to another. Each time any of these bits are changed (PLLS, IREFS, CLKS, or EREFS), the corresponding bits in the MCGSC register (PLLST, IREFST, CLKST, or OSCINIT) must be checked before moving on in the application software.

Additionally, care must be taken to ensure that the reference clock divider (RDIV) is set properly for the mode being switched to. For instance, in PEE mode, if using a 4 MHz crystal, RDIV must be set to %001 (divide-by-2) or %010 (divide -by-4) in order to divide the external reference down to the required frequency between 1 and 2 MHz.

If switching to FBE or FEE mode, first setting the DIV32 bit will ensure a proper reference frequency is sent to the FLL clock at all times.

In FBE, FEE, FBI, and FEI modes, at any time, the application can switch the FLL multiplication factor between 1024 and 512 with the DRS bit in MCGT. Writes to DRS will be ignored if LP=1 or PLLS=1.

The RDIV and IREFS bits should always be set properly before changing the PLLS bit so that the FLL or PLL clock has an appropriate reference clock frequency to switch to. The table below shows MCGOUT frequency calculations using RDIV, BDIV, and VDIV settings for each clock mode. The bus frequency is equal to MCGOUT divided by 2.

Table 8-10. MCGOUT Frequency Calculation Options

Clock Mode	f_{MCGOUT}[1]	Note
FEI (FLL engaged internal)	(f_{int} * F) / B	Typical f_{MCGOUT} = 16 MHz immediately after reset.
FEE (FLL engaged external)	(f_{ext} / R *F) / B	f_{ext} / R must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	f_{ext} / B	f_{ext} / R must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	f_{int} / B	Typical f_{int} = 32 kHz
PEE (PLL engaged external)	[(f_{ext} / R) * M] / B	f_{ext} / R must be in the range of 1 MHz to 2 MHz
PBE (PLL bypassed external)	f_{ext} / B	f_{ext} / R must be in the range of 1 MHz to 2 MHz
BLPI (Bypassed low power internal)	f_{int} / B	
BLPE (Bypassed low power external)	f_{ext} / B	

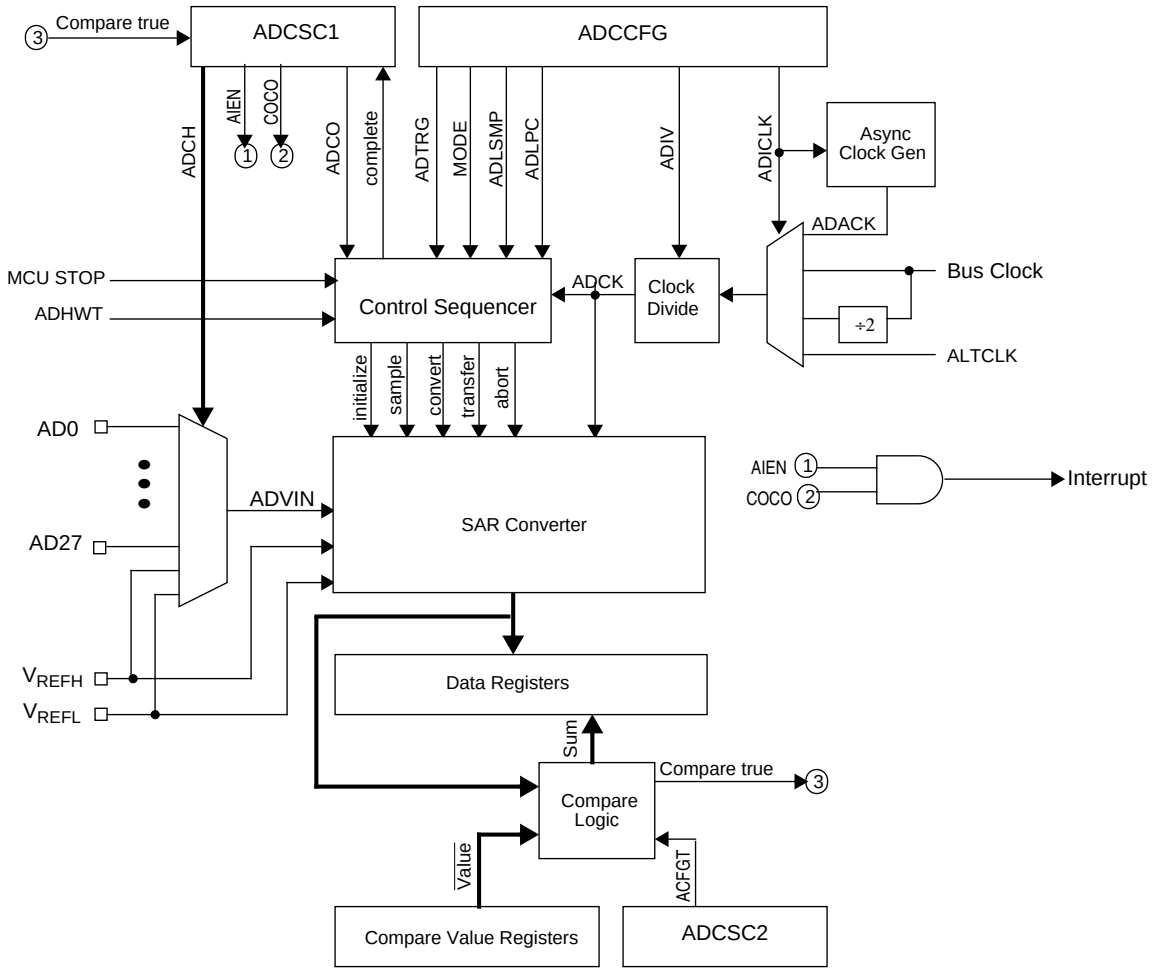


Figure 10-2. ADC Block Diagram

10.2 External Signal Description

The ADC module supports up to 28 separate analog inputs. It also requires four supply/reference/ground connections.

Table 10-2. Signal Properties

Name	Function
AD27–AD0	Analog Channel inputs
V_{REFH}	High reference voltage
V_{REFL}	Low reference voltage
V_{DDAD}	Analog power supply
V_{SSAD}	Analog ground

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

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

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

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

= Unused, always read 'x'

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

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

2 IDE is 0.

12.4.1 Identifier Registers (IDR0–IDR3)

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

12.4.1.1 IDR0–IDR3 for Extended Identifier Mapping

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

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

Table 12-36. CPU vs. MSCAN Operating Modes

CPU Mode	MSCAN Mode			
	Normal	Reduced Power Consumption		
		Sleep	Power Down	Disabled (CANE=0)
Run	CSWAI = X[1] SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = 1 SLPAK = 1		CSWAI = X SLPRQ = X SLPAK = X
Wait	CSWAI = 0 SLPRQ = 0 SLPAK = 0	CSWAI = 0 SLPRQ = 1 SLPAK = 1	CSWAI = 1 SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X
Stop3		CSWAI = X[2] SLPRQ = 1 SLPAK = 1	CSWAI = X SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = X SLPAK = X
Stop1 or 2			CSWAI = X SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X

1 'X' means don't care.

2 For a safe wake up from Sleep mode, SLPRQ and SLPAK must be set to 1 before going into Stop3 mode.

12.5.5.1 Operation in Run Mode

As shown in Table 12-36, only MSCAN sleep mode is available as low power option when the CPU is in run mode.

12.5.5.2 Operation in Wait Mode

The WAIT instruction puts the MCU in a low power consumption stand-by mode. If the CSWAI bit is set, additional power can be saved in power down mode because the CPU clocks are stopped. After leaving this power down mode, the MSCAN restarts its internal controllers and enters normal mode again.

While the CPU is in wait mode, the MSCAN can be operated in normal mode and generate interrupts (registers can be accessed via background debug mode). The MSCAN can also operate in any of the low-power modes depending on the values of the SLPRQ/SLPAK and CSWAI bits as seen in Table 12-36.

12.5.5.3 Operation in Stop Mode

The STOP instruction puts the MCU in a low power consumption stand-by mode. In stop1 or stop2 modes, the MSCAN is set in power down mode regardless of the value of the SLPRQ/SLPAK. In stop3 mode, power down or sleep modes are determined by the SLPRQ/SLPAK values set prior to entering stop3. CSWAI bit has no function in any of the stop modes.Table 12-36.

12.5.5.6 MSCAN Power Down Mode

The MSCAN is in power down mode (Table 12-36) when

- CPU is in stop mode

  or
- CPU is in wait mode and the CSWAI bit is set

When entering the power down mode, the MSCAN immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations to the above rule, the MSCAN immediately drives the TXCAN pin into a recessive state.

NOTE

The user is responsible for ensuring that the MSCAN is not active when power down mode is entered. The recommended procedure is to bring the MSCAN into Sleep mode before the STOP or WAIT instruction (if CSWAI is set) is executed. Otherwise, the abort of an ongoing message can cause an error condition and impact other CAN bus devices.

In power down mode, all clocks are stopped and no registers can be accessed. If the MSCAN was not in sleep mode before power down mode became active, the module performs an internal recovery cycle after powering up. This causes some fixed delay before the module enters normal mode again.

12.5.5.7 Programmable Wake-Up Function

The MSCAN can be programmed to wake up the MSCAN as soon as CAN bus activity is detected (see control bit WUPE in Section 12.3.1, "MSCAN Control Register 0 (CANCTL0)"). The sensitivity to existing CAN bus action can be modified by applying a low-pass filter function to the RXCAN input line while in sleep mode (see control bit WUPM in Section 12.3.2, "MSCAN Control Register 1 (CANCTL1)").

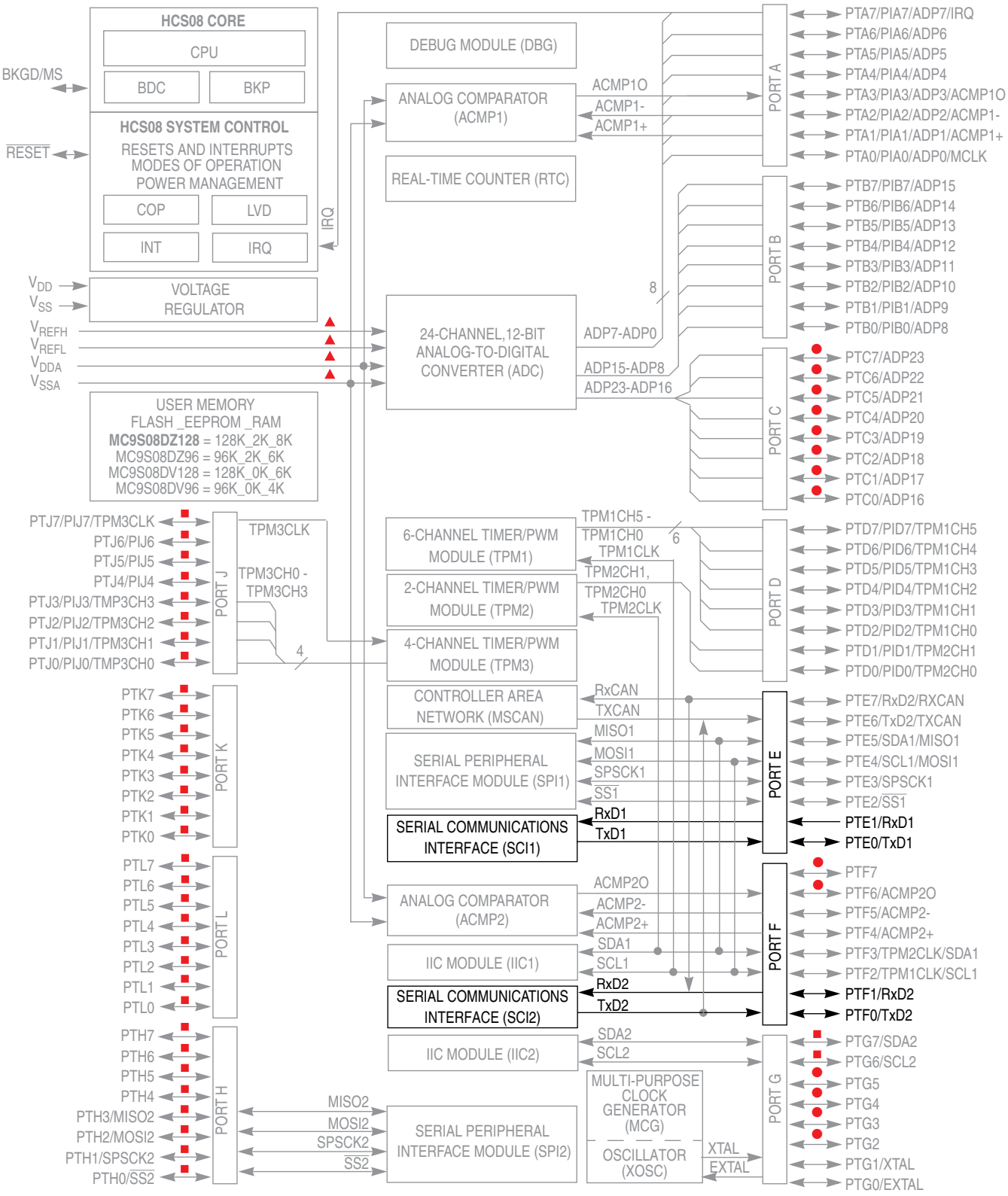
This feature can be used to protect the MSCAN from wake-up due to short glitches on the CAN bus lines. Such glitches can result from—for example—electromagnetic interference within noisy environments.

12.5.6 Reset Initialization

The reset state of each individual bit is listed in Section 12.3, "Register Definition," which details all the registers and their bit-fields.

12.5.7 Interrupts

This section describes all interrupts originated by the MSCAN. It documents the enable bits and generated flags. Each interrupt is listed and described separately.



■ - Pin not connected in 64-pin and 48-pin packages ● - Pin not available in the 48-pin package

▲ - In 48-pin package, VDDA and VREFH are internally connected to each other and VSSA and VREFL are internally connected to each other.

Figure 14-1. MC9S08DZ128 Block Diagram with SCI Highlighted

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								BDFR[1]
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

1 BDFR is writable only through serial background mode debug commands, not from user programs.

Figure 17-6. System Background Debug Force Reset Register (SBDFR)

Table 17-3. SBDFR Register Field Description

Field	Description
0 BDFR	**Background Debug Force Reset** — A serial active background mode command such as WRITE_BYTE allows an external debug host to force a target system reset. Writing 1 to this bit forces an MCU reset. This bit cannot be written from a user program.

18.3.3.10 Debug Comparator B Extension Register (DBGCBX)

Module Base + 0x0009

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

= Unimplemented or Reserved

Figure 18-11. Debug Comparator B Extension Register (DBGCBX)

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

Table 18-12. DBGCBX Field Descriptions

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

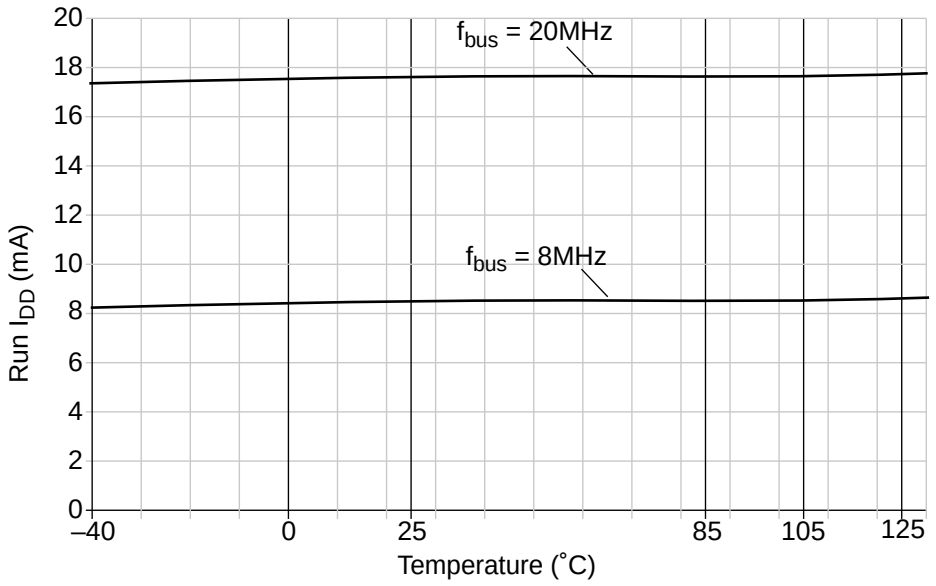


Figure A-6. Typical Run IDD vs. Temperature (VDD = 5V)

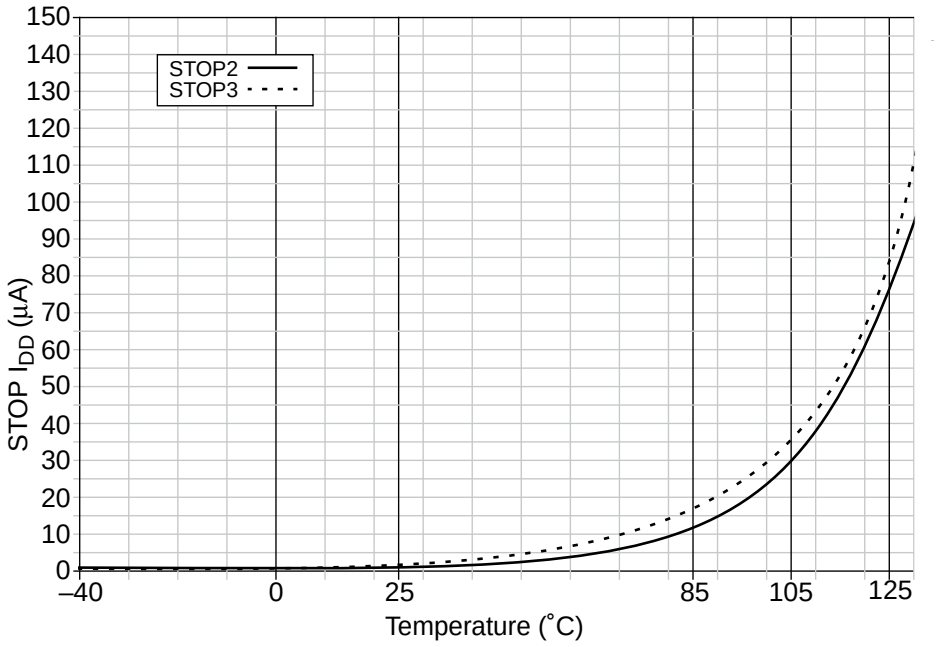


Figure A-7. Typical Stop I_{DD} vs. Temperature (V_{DD} = 5V)

A.8 Analog Comparator (ACMP) Electricals

Table A-8. Analog Comparator Electrical Specifications

Num	C	Rating	Symbol	Min	Typical	Max	Unit
1	—	Supply voltage	V_{DD}	2.7	—	5.5	V

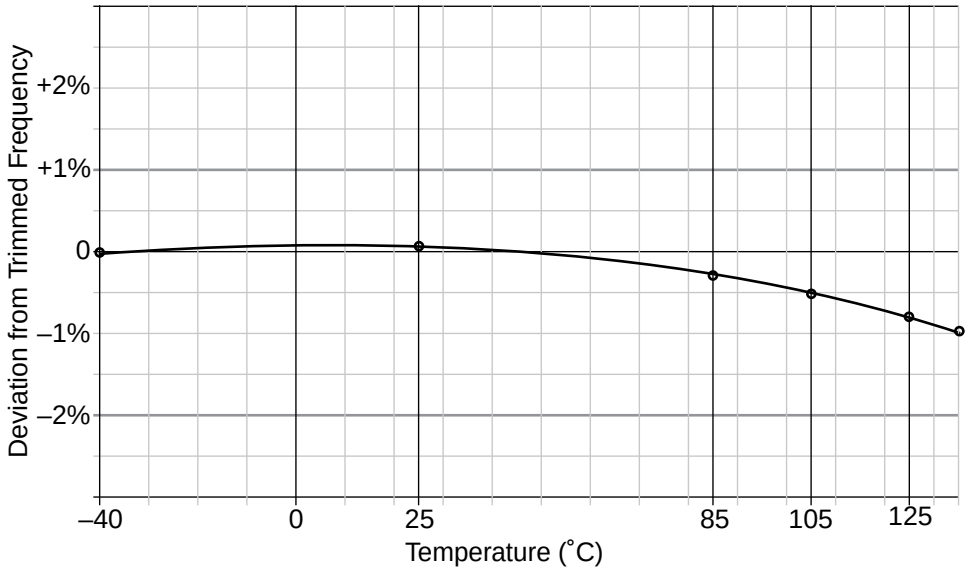


Figure A-9. Typical Frequency Deviation vs Temperature (ICS Trimmed to 16MHz bus@25˚C, 5V, FEI)[1]

A.12 AC Characteristics

This section describes ac timing characteristics for each peripheral system.

A.12.1 Control Timing

Table A-13. Control Timing

Num	C	Rating	Symbol	Min	Typical[1]	Max	Unit
1	D	Bus frequency (t_{cyc} = 1/f_{Bus})	f_{Bus}	dc	—	20	MHz
2	D	Internal low-power oscillator period	t_{LPO}	800		1500	µs
3	D	External reset pulse width[2]	t_{extrst}	100		—	ns
4	D	Reset low drive[3]	t_{rstdrv}	34 x t_{cyc}		—	ns
5	D	Active background debug mode latch setup time	t_{MSSU}	500		—	ns
6	D	Active background debug mode latch hold time	t_{MSH}	100		—	ns
7	D	IRQ/PIAx/ PIBx/PIDx/PIJx pulse width Asynchronous path[2] Synchronous path[3]	t_{ILIH}, t_{IHIL}	100 1.5 t_{cyc}	—	—	ns
8	C	Port rise and fall time (load = 50 pF)[4] Slew rate control disabled Slew rate control enabled	t_{Rise}, t_{Fall}	— —	3 30		ns

1 Typical data was characterized at 5.0 V, 25°C unless otherwise stated.

2 This is the shortest pulse that is guaranteed to be recognized as a $\overline{\text{RESET}}$ pin or pin interrupt request. Shorter pulses are not guaranteed to override reset requests from internal sources.

1. Based on the average of several hundred units from a typical characterization lot.