



Welcome to [E-XFL.COM](https://www.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 <sup>2</sup> C, 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	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	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz128f2mlh">https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz128f2mlh</a>



Section Number	Title	Page
7.4.2	Interrupt Sequence .....	149
7.4.3	Wait Mode Operation .....	150
7.4.4	Stop Mode Operation .....	150
7.4.5	BGND Instruction .....	151
7.5	CALL and RTC Instructions .....	151
7.6	HCS08 Instruction Set Summary .....	153

## Chapter 8 Multi-Purpose Clock Generator (S08MCGV2)

8.1	Introduction .....	165
8.1.1	Features .....	167
8.1.2	Modes of Operation .....	169
8.2	External Signal Description .....	169
8.3	Register Definition .....	170
8.3.1	MCG Control Register 1 (MCGC1) .....	170
8.3.2	MCG Control Register 2 (MCGC2) .....	172
8.3.3	MCG Trim Register (MCGTRM) .....	173
8.3.4	MCG Status and Control Register (MCGSC) .....	174
8.3.5	MCG Control Register 3 (MCGC3) .....	175
8.3.6	MCG Test and Control Register (MCGT) .....	177
8.4	Functional Description .....	178
8.4.1	Operational Modes .....	178
8.4.2	Mode Switching .....	182
8.4.3	Bus Frequency Divider .....	183
8.4.4	Low Power Bit Usage .....	183
8.4.5	Internal Reference Clock .....	183
8.4.6	External Reference Clock .....	183
8.4.7	Fixed Frequency Clock .....	184
8.5	Initialization / Application Information .....	185
8.5.1	MCG Module Initialization Sequence .....	185
8.5.2	Using a 32.768 kHz Reference .....	186
8.5.3	MCG Mode Switching .....	187
8.5.4	Calibrating the Internal Reference Clock (IRC) .....	195

## Chapter 9 5-V Analog Comparator (S08ACMPV3)

9.1	Introduction .....	199
9.1.1	ACMP Configuration Information .....	199
9.1.2	Features .....	201
9.1.3	Modes of Operation .....	201
9.1.4	Block Diagram .....	201
9.2	External Signal Description .....	203

## 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 <sup>1</sup>	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 <sup>2</sup>
1	0	Both bits must be 1		x <sup>3</sup>	Stop3 with voltage regulator active
1	0	Either bit a 0		0	Stop3
1	0	Either bit a 0		1	Stop2

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

<sup>2</sup> When in Stop3 mode with BDM enabled, The S<sub>IDD</sub> will be near R<sub>IDD</sub> levels because internal clocks are enabled.

<sup>3</sup> 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.

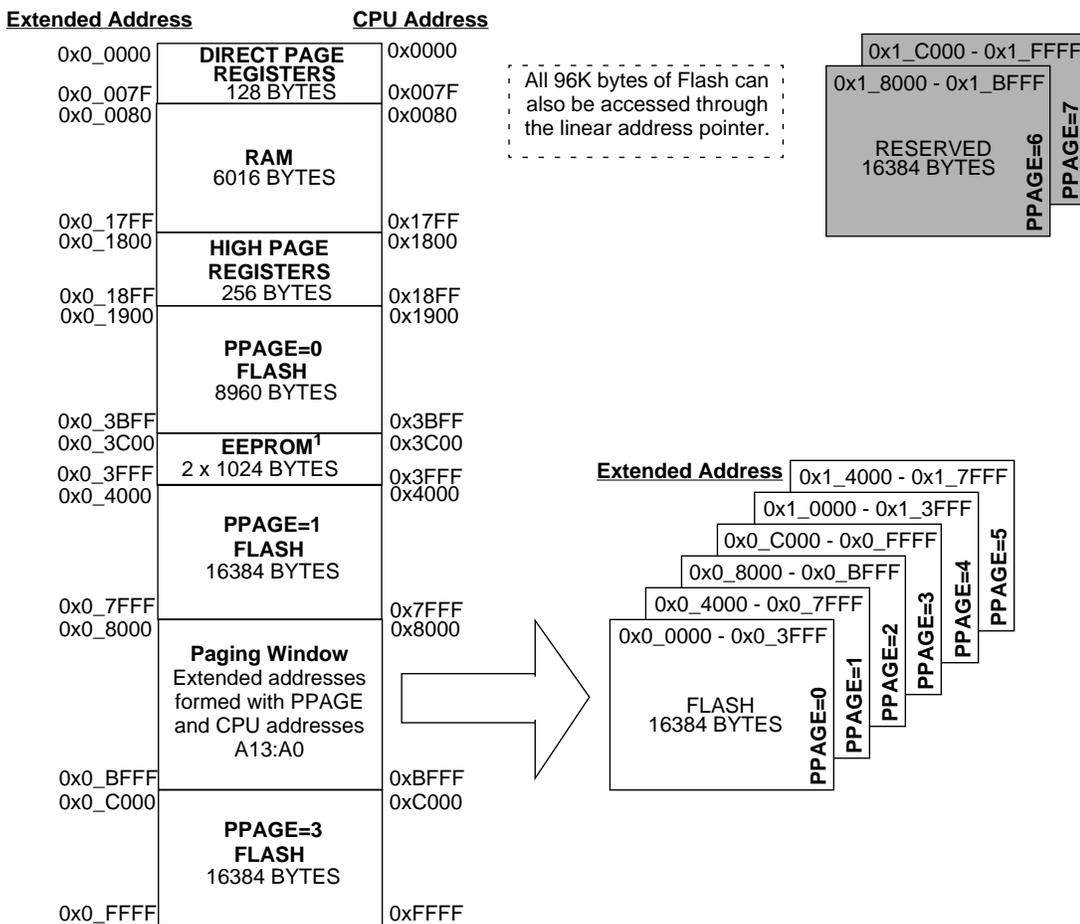
Exit from stop3 is done by asserting  $\overline{\text{RESET}}$  or an asynchronous interrupt pin. The asynchronous interrupt pins are IRQ, PIA0–PIA7, PIB0–PIB7, PID0–PID7, and PIJ0–PIJ7. Exit from stop3 can also be done by the low voltage detection (LVD) reset, the low voltage warning (LVW) interrupt, 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.



<sup>1</sup> EEPROM address range shows half the total EEPROM. See Section 4.6.10, "EEPROM Mapping" for more details.

Figure 4-2. MC9S08DZ96 Memory Map

Table 4-2. Direct-Page Register Summary (Sheet 1 of 4)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0000	PTAD	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
0x0001	PTADD	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
0x0002	PTBD	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0x0003	PTBDD	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0004	PTCD	PTCD7	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
0x0005	PTCDD	PTCDD7	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x0006	PTDD	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
0x0007	PTDDD	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
0x0008	PTED	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
0x0009	PTEDD	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
0x000A	PTFD	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
0x000B	PTFDD	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
0x000C	PTGD	PTGD7	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
0x000D	PTGDD	PTGDD7	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
0x000E	ACMP1SC	ACME	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD1	ACMOD0
0x000F	ACMP2SC	ACME	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD1	ACMOD0
0x0010	ADCSC1	COCO	AIEN	ADCO	ADCH				
0x0011	ADCSC2	ADACT	ADTRG	ACFE	ACFGT	0	0	Reserved	Reserved
0x0012	ADCRH	0	0	0	0	ADR11	ADR10	ADR9	ADR8
0x0013	ADCRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0x0014	ADCCVH	0	0	0	0	ADC11	ADC10	ADCV9	ADCV8
0x0015	ADCCVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
0x0016	ADCCFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0x0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0x0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
0x0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
0x001A– 0x001B	Reserved	—	—	—	—	—	—	—	—
0x001C	IRQSC	0	IRQPDD	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
0x001D– 0x001F	Reserved	—	—	—	—	—	—	—	—
0x0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x0022	TPM1CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0

**Table 4-3. High-Page Register Summary (Sheet 5 of 5)**

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x18C7	TPM3C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x18C8	TPM3C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x18C9	TPM3C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x18CA	TPM3C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x18CB	TPM3C2SC	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
0x18CC	TPM3C2VH	Bit 15	14	13	12	11	10	9	Bit 8
0x18CD	TPM3C2VL	Bit 7	6	5	4	3	2	1	Bit 0
0x18CE	TPM3C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
0x18CF	TPM3C3VH	Bit 15	14	13	12	11	10	9	Bit 8
0x18D0	TPM3C3VL	Bit 7	6	5	4	3	2	1	Bit 0
0x18D1– 0x18D7	Reserved	—	—	—	—	—	—	—	—
0x18D8	IIC2A	AD7	AD6	AD5	AD4	AD3	AD2	AD1	0
0x18D9	IIC2F	MULT			ICR				
0x18DA	IIC2C1	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
0x18DB	IIC2S	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
0x18DC	IIC2D	DATA							
0x18DD	IIC2C2	GCAEN	ADEXT	0	0	0	AD10	AD9	AD8
0x18DE– 0x18FF	Reserved	—	—	—	—	—	—	—	—

Figure 4-4 shows the structure of receive and transmit buffers for extended identifier mapping. These registers vary depending on whether standard or extended mapping is selected. See [Chapter 12](#), “Freescale’s Controller Area Network (S08MSCANV1),” for details on extended and standard identifier mapping.

**Table 4-4. MSCAN Foreground Receive and Transmit Buffer Layouts — Extended Mapping Shown (Sheet 1 of 2)**

0x18A0	CANRIDR0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
0x18A1	CANRIDR1	ID20	ID19	ID18	SRR <sup>(1)</sup>	IDE <sup>(1)</sup>	ID17	ID16	ID15
0x18A2	CANRIDR2	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
0x18A3	CANRIDR3	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR <sup>2</sup>
0x18A4 – 0x18AB	CANRDSR0 – CANRDSR7	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0x18AC	CANRDLR	—	—	—	—	DLC3	DLC2	DLC1	DLC0
0x18AD	Reserved	—	—	—	—	—	—	—	—
0x18AE	CANRTSRH	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
0x18AF	CANRTSRL	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
0x18B0	CANTIDR0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
0x18B1	CANTIDR1	ID20	ID19	ID18	SRR <sup>(3)</sup>	IDE <sup>(1)</sup>	ID17	ID16	ID15
0x18B2	CANTIDR2	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	D7	D6	D5	D4	D3	D2	D1	D0
Reset:	0	0	0	0	0	0	0	0

**Figure 4-10. Linear Address Pointer Add Byte Register (LAPAB)**

**Table 4-11. Linear Address Pointer Add Byte Register Field Descriptions**

Field	Description
7:0 D7:D0	The 2s complement value written to LAPAB will be added to contents of the linear address pointer register, LAP2:LAP0. Writing a value of 0x7f to LAPAB will increase LAP by 127, a value of 0x80 will decrease LAP by 128, and a value of 0xff will decrease LAP by 1.

## 4.5 RAM

The MC9S08DZ128 Series includes static RAM. The locations in RAM below 0x0100 can be accessed using the more efficient direct addressing mode, and any single bit in this area can be accessed with the bit manipulation instructions (BCLR, BSET, BRCLR, and BRSET). Locating the most frequently accessed program variables in this area of RAM is preferred.

The RAM retains data while the MCU is in low-power wait, stop2, or stop3 mode. At power-on the contents of RAM are uninitialized. RAM data is unaffected by any reset if the supply voltage does not drop below the minimum value for RAM retention ( $V_{RAM}$ ).

For compatibility with M68HC05 MCUs, the HCS08 resets the stack pointer to 0x00FF. In the MC9S08DZ128 Series, it is usually best to reinitialize the stack pointer to the top of the RAM so the direct page RAM can be used for frequently accessed RAM variables and bit-addressable program variables. Include the following 2-instruction sequence in your reset initialization routine (where RamLast is equated to the highest address of the RAM in the Freescale Semiconductor equate file).

```
LDHX    #RamLast+1    ;point one past RAM
TXS                    ;SP<-(H:X-1)
```

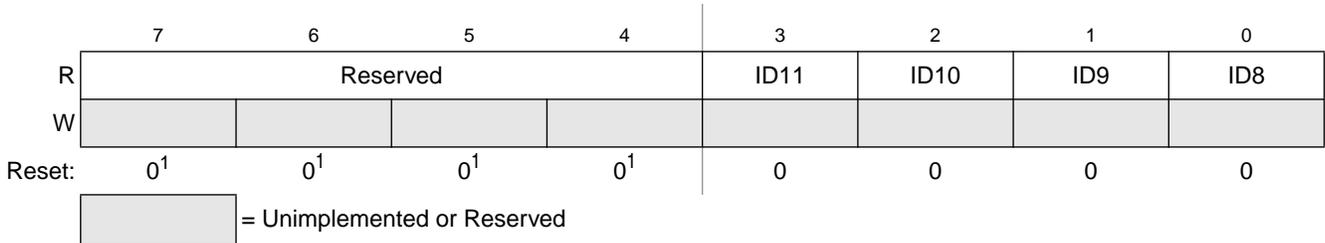
### NOTE

On most devices in the MC9S08DZ128 Series, more than 4K of RAM is present in two separate address blocks.

When security is enabled, the RAM is considered a secure memory resource and is not accessible through BDM or code executing from non-secure memory. See [Section 4.6.9, “Security”](#), for a detailed description of the security feature.

## 5.8.6 System Device Identification Register (SDIDH, SDIDL)

These high page read-only registers are included so host development systems can identify the HCS08 derivative and revision number. This allows the development software to recognize where specific memory blocks, registers, and control bits are located in a target MCU.

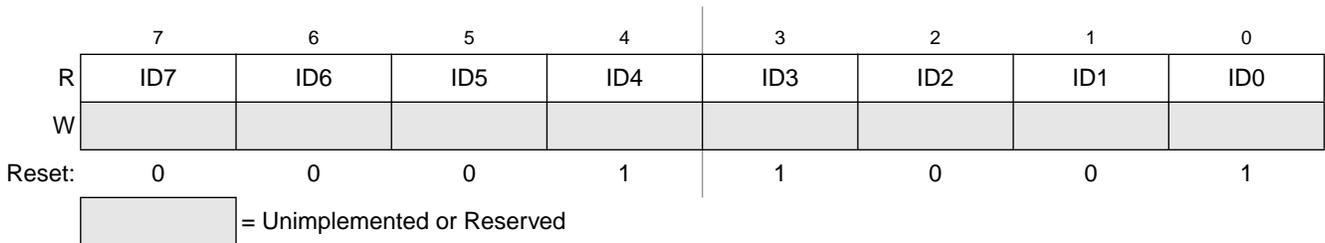


<sup>1</sup> The revision number that is hard coded into these bits reflects the current silicon revision level.

**Figure 5-7. System Device Identification Register — High (SDIDH)**

**Table 5-8. SDIDH Register Field Descriptions**

Field	Description
3:0 ID[11:8]	<b>Part Identification Number</b> — MC9S08DZ128 Series MCUs are hard-coded to the value 0x0019. See also ID bits in <a href="#">Table 5-9</a> .



**Figure 5-8. System Device Identification Register — Low (SDIDL)**

**Table 5-9. SDIDL Register Field Descriptions**

Field	Description
7:0 ID[7:0]	<b>Part Identification Number</b> — MC9S08DZ128 Series MCUs are hard-coded to the value 0x0019. See also ID bits in <a href="#">Table 5-8</a> .

### 6.5.2.5 Port B Drive Strength Selection Register (PTBDS)

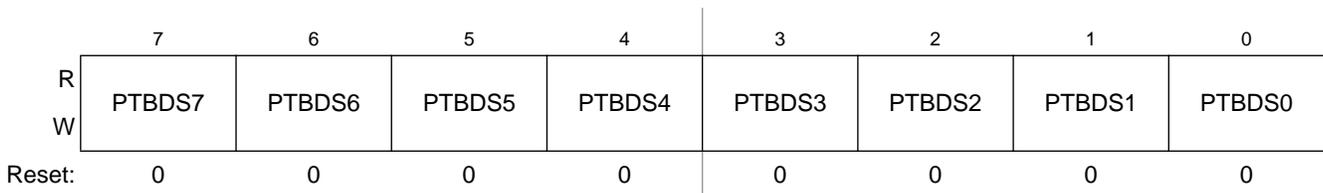


Figure 6-15. Drive Strength Selection for Port B Register (PTBDS)

Table 6-13. PTBDS Register Field Descriptions

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

### 6.5.2.6 Port B Interrupt Status and Control Register (PTBSC)

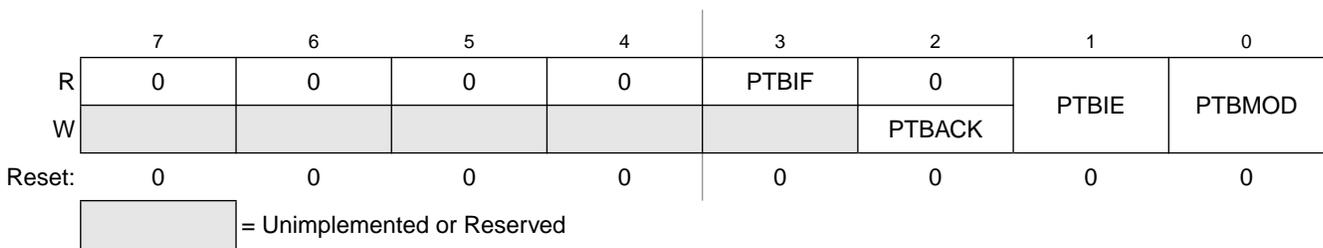


Figure 6-16. Port B Interrupt Status and Control Register (PTBSC)

Table 6-14. PTBSC Register Field Descriptions

Field	Description
3 PTBIF	<b>Port B Interrupt Flag</b> — PTBIF indicates when a Port B interrupt is detected. Writes have no effect on PTBIF. 0 No Port B interrupt detected. 1 Port B interrupt detected.
2 PTBACK	<b>Port B Interrupt Acknowledge</b> — Writing a 1 to PTBACK is part of the flag clearing mechanism. PTBACK always reads as 0.
1 PTBIE	<b>Port B Interrupt Enable</b> — PTBIE determines whether a port B interrupt is requested. 0 Port B interrupt request not enabled. 1 Port B interrupt request enabled.
0 PTBMOD	<b>Port B Detection Mode</b> — PTBMOD (along with the PTBES bits) controls the detection mode of the port B interrupt pins. 0 Port B pins detect edges only. 1 Port B pins detect both edges and levels.

### 6.5.3.3 Port C Pull Enable Register (PTCPE)

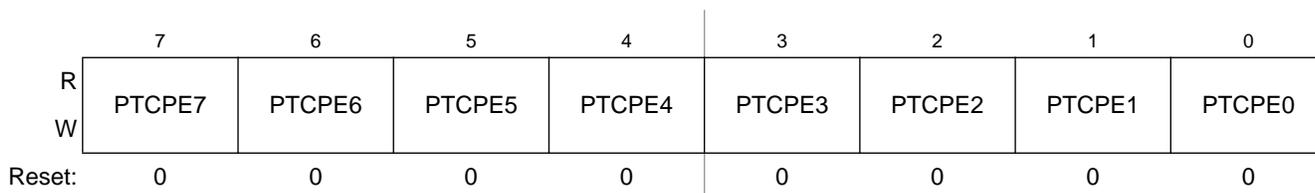


Figure 6-21. Internal Pull Enable for Port C Register (PTCPE)

Table 6-19. PTCPE Register Field Descriptions

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

**NOTE**

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

### 6.5.3.4 Port C Slew Rate Enable Register (PTCSE)

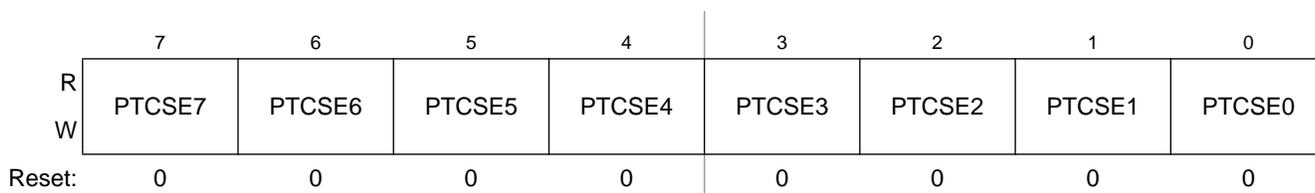


Figure 6-22. Slew Rate Enable for Port C Register (PTCSE)

Table 6-20. PTCSE Register Field Descriptions

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

## 6.5.11 Port L Registers

Port L is controlled by the registers listed below.

### 6.5.11.1 Port L Data Register (PTLD)

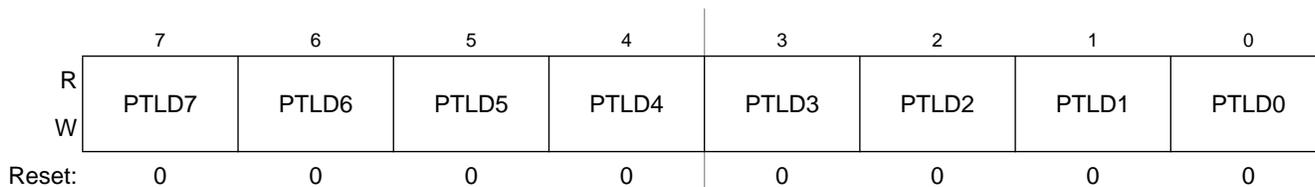


Figure 6-65. Port L Data Register (PTLD)

Table 6-63. PTLD Register Field Descriptions

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

### 6.5.11.2 Port L Data Direction Register (PTLDD)

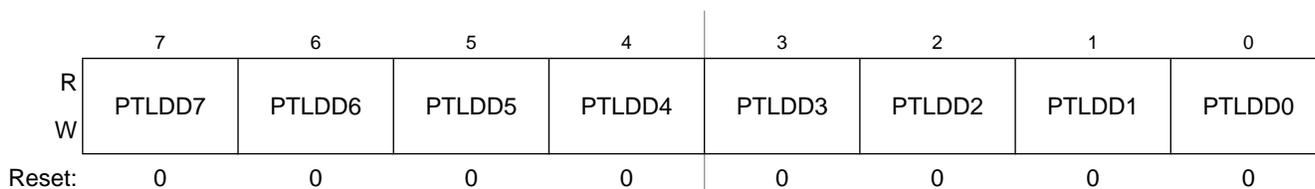


Figure 6-66. Port L Data Direction Register (PTLDD)

Table 6-64. PTLDD Register Field Descriptions

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

Table 7-2. Instruction Set Summary (Sheet 3 of 9)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V 1 1 H	I N Z C
BPL <i>rel</i>	Branch if Plus (if N = 0)	REL	2A rr	3	ppp	- 1 1 -	- - - -
BRA <i>rel</i>	Branch Always (if I = 1)	REL	20 rr	3	ppp	- 1 1 -	- - - -
BRCLR <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Clear (if (Mn) = 0)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 dd rr 03 dd rr 05 dd rr 07 dd rr 09 dd rr 0B dd rr 0D dd rr 0F dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	- 1 1 -	- - - - ↑
BRN <i>rel</i>	Branch Never (if I = 0)	REL	21 rr	3	ppp	- 1 1 -	- - - -
BRSET <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Set (if (Mn) = 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 dd rr 02 dd rr 04 dd rr 06 dd rr 08 dd rr 0A dd rr 0C dd rr 0E dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	- 1 1 -	- - - - ↑
BSET <i>n,opr8a</i>	Set Bit <i>n</i> in Memory (Mn ← 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 dd 12 dd 14 dd 16 dd 18 dd 1A dd 1C dd 1E dd	5 5 5 5 5 5 5 5	rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp	- 1 1 -	- - - -
BSR <i>rel</i>	Branch to Subroutine PC ← (PC) + \$0002 push (PCL); SP ← (SP) - \$0001 push (PCH); SP ← (SP) - \$0001 PC ← (PC) + <i>rel</i>	REL	AD rr	5	ssppp	- 1 1 -	- - - -
CALL <i>page, opr16a</i>	Call Subroutine	EXT	AC pg hhll	8	ppsssppp	- 1 1 -	- - - -
CBEQ <i>opr8a,rel</i> CBEQA # <i>opr8i,rel</i> CBEQX # <i>opr8i,rel</i> CBEQ <i>opr8,X+,rel</i> CBEQ <i>,X+,rel</i> CBEQ <i>opr8,SP,rel</i>	Compare and... Branch if (A) = (M) Branch if (A) = (M) Branch if (X) = (M) Branch if (A) = (M) Branch if (A) = (M) Branch if (A) = (M)	DIR IMM IMM IX1+ IX+ SP1	31 dd rr 41 ii rr 51 ii rr 61 ff rr 71 rr 9E 61 ff rr	5 4 4 5 5 6	rpppp pppp pppp rpppp rfppp prpppp	- 1 1 -	- - - -
CLC	Clear Carry Bit (C ← 0)	INH	98	1	p	- 1 1 -	- - - 0
CLI	Clear Interrupt Mask Bit (I ← 0)	INH	9A	1	p	- 1 1 -	0 - - -
CLR <i>opr8a</i> CLRA CLR X CLR H CLR <i>opr8,X</i> CLR <i>,X</i> CLR <i>opr8,SP</i>	Clear M ← \$00 A ← \$00 X ← \$00 H ← \$00 M ← \$00 M ← \$00 M ← \$00	DIR INH INH INH IX1 IX SP1	3F dd 4F 5F 8C 6F ff 7F 9E 6F ff	5 1 1 1 5 4 6	rfwpp p p p rfwpp rfwp prfwpp	0 1 1 -	- 0 1 -

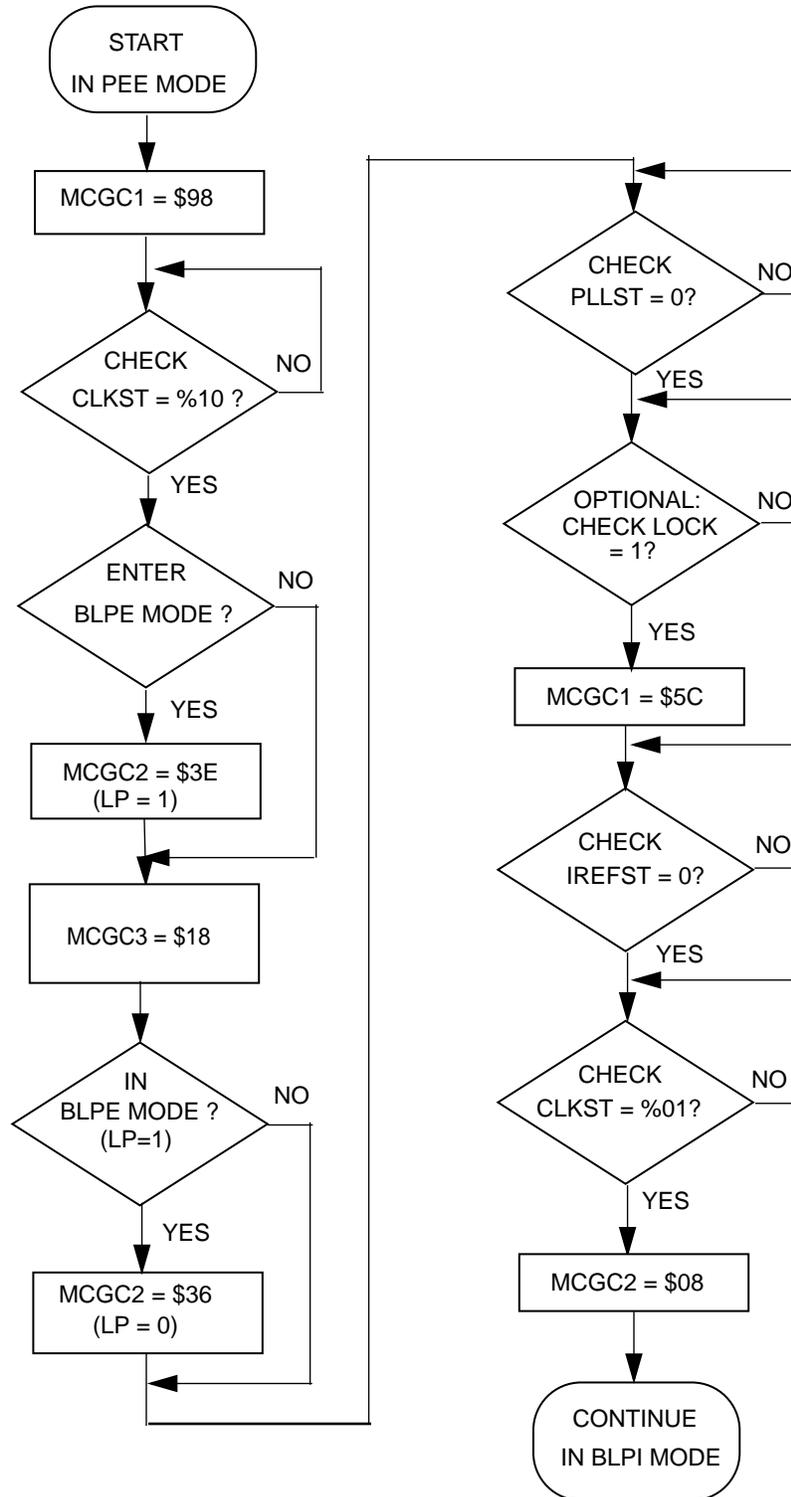


Figure 8-11. Flowchart of PEE to BLPI Mode Transition using an 8 MHz crystal

**Table 10-1. ADC Channel Assignment**

ADCH	Channel	Input	ADCH	Channel	Input
00000	AD0	PTA0/ADP0/MCLK	10000	AD16	PTC0/ADP16
00001	AD1	PTA1/ADP1/ACMP1+	10001	AD17	PTC1/ADP17
00010	AD2	PTA2/ADP2/ACMP1P-	10010	AD18	PTC2/ADP18
00011	AD3	PTA3/ADP3/ACMP1O	10011	AD19	PTC3/ADP19
00100	AD4	PTA4/ADP4	10100	AD20	PTC4/ADP20
00101	AD5	PTA5/ADP5	10101	AD21	PTC5/ADP21
00110	AD6	PTA6/ADP6	10110	AD22	PTC6/ADP22
00111	AD7	PTA7/ADP7	10111	AD23	PTC7/ADP23
01000	AD8	PTB0/ADP8	11000– 11001	AD24 through AD25	Reserved
01001	AD9	PTB1/ADP9	11010	AD26	Temperature Sensor <sup>1</sup>
01010	AD10	PTB2/ADP10	11011	AD27	Internal Bandgap <sup>2</sup>
01011	AD11	PTB3/ADP11	11100	V <sub>REFH</sub>	V <sub>REFH</sub>
01100	AD12	PTB4/ADP12	11101	V <sub>REFH</sub>	V <sub>REFH</sub>
01101	AD13	PTB5/ADP13	11110	V <sub>REFL</sub>	V <sub>REFL</sub>
01110	AD14	PTB6/ADP14	11111	Module Disabled	None
01111	AD15	PTB7/ADP15			

**Notes:**

- 1 For information, see [Section 10.1.5, “Temperature Sensor”](#).
- 2 Requires BGBE =1 in SPMSC1 see [Section 5.8.7, “System Power Management Status and Control 1 Register \(SPMSC1\)”](#). For value of bandgap voltage reference see [Section A.6, “DC Characteristics”](#).

### 10.1.3 Alternate Clock

The ADC module is capable of performing conversions using the MCU bus clock, the bus clock divided by two, the local asynchronous clock (ADACK) within the module, or the alternate clock, ALTCLK. The alternate clock for the MC9S08DZ128 Series MCU devices is the external reference clock (MCGERCLK).

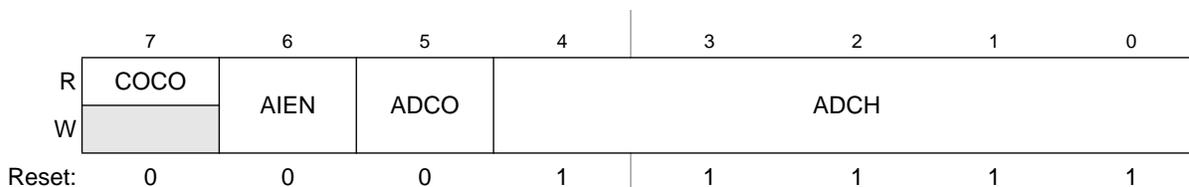
The selected clock source must run at a frequency such that the ADC conversion clock (ADCK) runs at a frequency within its specified range ( $f_{ADCK}$ ) after being divided down from the ALTCLK input as determined by the ADIV bits.

ALTCLK is active while the MCU is in wait mode provided the conditions described above are met. This allows ALTCLK to be used as the conversion clock source for the ADC while the MCU is in wait mode.

ALTCLK cannot be used as the ADC conversion clock source while the MCU is in either stop2 or stop3.

### 10.1.4 Hardware Trigger

The ADC hardware trigger, ADHWT, is the output from the real time counter (RTC) overflow or the external interrupt request (IRQ) pin. The source is selected by the ADC hardware trigger select bit,


**Figure 10-3. Status and Control Register (ADCSC1)**
**Table 10-3. ADCSC1 Field Descriptions**

Field	Description
7 COCO	Conversion Complete Flag. The COCO flag is a read-only bit set each time a conversion is completed when the compare function is disabled (ACFE = 0). When the compare function is enabled (ACFE = 1), the COCO flag is set upon completion of a conversion only if the compare result is true. This bit is cleared when ADCSC1 is written or when ADCRL is read. 0 Conversion not completed 1 Conversion completed
6 AIEN	Interrupt Enable AIEN enables conversion complete interrupts. When COCO becomes set while AIEN is high, an interrupt is asserted. 0 Conversion complete interrupt disabled 1 Conversion complete interrupt enabled
5 ADCO	Continuous Conversion Enable. ADCO enables continuous conversions. 0 One conversion following a write to the ADCSC1 when software triggered operation is selected, or one conversion following assertion of ADHWT when hardware triggered operation is selected. 1 Continuous conversions initiated following a write to ADCSC1 when software triggered operation is selected. Continuous conversions are initiated by an ADHWT event when hardware triggered operation is selected.
4:0 ADCH	Input Channel Select. The ADCH bits form a 5-bit field that selects one of the input channels. The input channels are detailed in <a href="#">Table 10-4</a> . The successive approximation converter subsystem is turned off when the channel select bits are all set. This feature allows for explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional, single conversion from being performed. It is not necessary to set the channel select bits to all ones to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes.

**Table 10-4. Input Channel Select**

ADCH	Input Select
00000–01111	AD0–15
10000–11011	AD16–27
11100	Reserved
11101	$V_{REFH}$
11110	$V_{REFL}$
11111	Module disabled

### 10.4.1 Clock Select and Divide Control

One of four clock sources can be selected as the clock source for the ADC module. This clock source is then divided by a configurable value to generate the input clock to the converter (ADCK). The clock is selected from one of the following sources by means of the ADICLK bits.

- The bus clock, which is equal to the frequency at which software is executed. This is the default selection following reset.
- The bus clock divided by two. For higher bus clock rates, this allows a maximum divide by 16 of the bus clock.
- ALTCLK, as defined for this MCU (See module section introduction).
- The asynchronous clock (ADACK). This clock is generated from a clock source within the ADC module. When selected as the clock source, this clock remains active while the MCU is in wait or stop3 mode and allows conversions in these modes for lower noise operation.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC do not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by the ADIV bits and can be divide-by 1, 2, 4, or 8.

### 10.4.2 Input Select and Pin Control

The pin control registers (APCTL3, APCTL2, and APCTL1) disable the I/O port control of the pins used as analog inputs. When a pin control register bit is set, the following conditions are forced for the associated MCU pin:

- The output buffer is forced to its high impedance state.
- The input buffer is disabled. A read of the I/O port returns a zero for any pin with its input buffer disabled.
- The pullup is disabled.

### 10.4.3 Hardware Trigger

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when the ADTRG bit is set. This source is not available on all MCUs. Consult the module introduction for information on the ADHWT source specific to this MCU.

When ADHWT source is available and hardware trigger is enabled (ADTRG=1), a conversion is initiated on the rising edge of ADHWT. If a conversion is in progress when a rising edge occurs, the rising edge is ignored. In continuous convert configuration, only the initial rising edge to launch continuous conversions is observed. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

### 10.4.4 Conversion Control

Conversions can be performed in 12-bit mode, 10-bit mode, or 8-bit mode as determined by the MODE bits. Conversions can be initiated by a software or hardware trigger. In addition, the ADC module can be

Section 12.3.1, “MSCAN Control Register 0 (CANCTL0)”). In case of a transmission, the CPU can only read the time stamp after the respective transmit buffer has been flagged empty.

The timer value, which is used for stamping, is taken from a free running internal CAN bit clock. A timer overrun is not indicated by the MSCAN. The timer is reset (all bits set to 0) during initialization mode. The CPU can only read the time stamp registers.

	7	6	5	4	3	2	1	0
R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
W								
Reset:	x	x	x	x	x	x	x	x

Figure 12-36. Time Stamp Register — High Byte (TSRH)

	7	6	5	4	3	2	1	0
R	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
W								
Reset:	x	x	x	x	x	x	x	x

Figure 12-37. Time Stamp Register — Low Byte (TSRL)

Read: 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)”).

Write: Unimplemented

## 12.5 Functional Description

### 12.5.1 General

This section provides a complete functional description of the MSCAN. It describes each of the features and modes listed in the introduction.

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<sup>1</sup>. 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.

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

This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

### 17.3.1 BDC Registers and Control Bits

The BDC has two registers:

- The BDC status and control register (BDCSCR) is an 8-bit register containing control and status bits for the background debug controller.
- The BDC breakpoint match register (BDCBKPT) holds a 16-bit breakpoint match address.

These registers are accessed with dedicated serial BDC commands and are not located in the memory space of the target MCU (so they do not have addresses and cannot be accessed by user programs).

Some of the bits in the BDCSCR have write limitations; otherwise, these registers may be read or written at any time. For example, the ENBDM control bit may not be written while the MCU is in active background mode. (This prevents the ambiguous condition of the control bit forbidding active background mode while the MCU is already in active background mode.) Also, the four status bits (BDMACT, WS, WSF, and DVF) are read-only status indicators and can never be written by the WRITE\_CONTROL serial BDC command. The clock switch (CLKSW) control bit may be read or written at any time.

