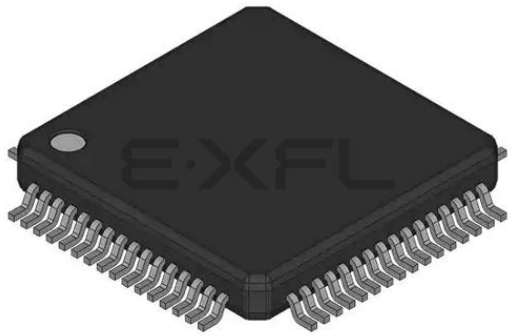

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
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/pro/item?MUrl=&PartUrl=mc9s08dv48aclh

Chapter 7
Central Processor Unit (S08CPUV3)

Chapter 8
Multi-Purpose Clock Generator (S08MCGV1)

Table 2-1. Pin Availability by Package Pin-Count

Pin Number			<-- Lowest Priority --> Highest			
64	48	32	Port Pin/Interrupt		Alt 1	Alt 2
1	1	—	PTB6	PIB6	ADP14	
2	—	—	PTC5			
3	2	1	PTA7	PIA7	ADP7	IRQ
4	—	—	PTC6			
5	3	—	PTB7	PIB7	ADP15	
6	—	—	PTC7			
7	4	2				V_{DD}
8	5	3				V_{SS}
9	6	4	PTG0		EXTAL	
10	7	5	PTG1		XTAL	
11	8	6				$\overline{\text{RESET}}$
12	9	—	PTF4			ACMP2+
13	10	—	PTF5			ACMP2-
14	—	—	PTF6			ACMP2O
15	11	7	PTE0		TxD1	
16	12	8	PTE1[2]		RxD1[2]	
17	13	9	PTE2			$\overline{\text{SS}}$
18	14	10	PTE3			SPSCK
19	15	11	PTE4		SCL[3]	MOSI
20	16	12	PTE5		SDA[3]	MISO
21	—	—	PTG2			
22	—	—	PTG3			
23	17	—	PTF0			TxD2[4]
24	18	—	PTF1			RxD2[4]
25	19	—	PTF2		TPM1CLK	SCL[3]
26	20	—	PTF3		TPM2CLK	SDA[3]
27	—	—	PTG4			
28	—	—	PTG5			
29	21	13	PTE6		TxD2[4]	TXCAN
30	22	14	PTE7		RxD2[4]	RxCAN
31	23	15	PTD0	PID0		TPM2CH0
32	24	16	PTD1	PID1		TPM2CH1
33	25	17	PTD2	PID2		TPM1CH0
34	26	18	PTD3	PID3		TPM1CH1
35	27	19	PTD4	PID4		TPM1CH2
36	28	20	PTD5	PID5		TPM1CH3
37	—	—	PTF7			
38	29	—				V_{SS}
39	30	—				V_{DD}
40	31	—	PTD6	PID6		TPM1CH4
41	32	—	PTD7	PID7		TPM1CH5
42	33	21			BKGD	MS
43	—	—	PTC0			
44	34	22	PTB0	PIB0	ADP8	
45	—	—	PTC1			
46	35	23	PTA0	PIA0	ADP0	MCLK
47	—	—	PTC2			
48	36	24	PTB1	PIB1	ADP9	
49	37	25	PTA1	PIA1	ADP1[1]	ACMP1+[1]
50	38	—	PTB2	PIB2	ADP10	
51	39	26	PTA2	PIA2	ADP2[1]	ACMP1-[1]
52	—	—	PTC3			
53	40	—	PTB3	PIB3	ADP11	
54	41	27	PTA3	PIA3	ADP3	ACMP1O
55	42	28				V_{SSA}
56						V_{REFL}
57	43	29				V_{REFH}
58						V_{DDA}
59	44	30	PTA4	PIA4	ADP4	
60	45	—	PTB4	PIB4	ADP12	
61	—	—	PTC4			
62	46	31	PTA5	PIA5	ADP5	
63	47	—	PTB5	PIB5	ADP13	
64	48	32	PTA6	PIA6	ADP6	

1. If both of these analog modules are enabled, they both will have access to the pin.
2. Pin does not contain a clamp diode to V_{DD} and should not be driven above V_{DD}. The voltage measured on this pin when internal pull-up is enabled may be as low as V_{DD} – 0.7 V. The internal gates connected to this pin are pulled to V_{DD}.
3. The IIC module pins can be repositioned using IICPS bit in the SOPT1 register. The default reset locations are on PTF2 and PTF3.
4. The SCI2 module pins can be repositioned using SCI2PS bit in the SOPT1 register. The default reset locations are on PTF0 and PTF1.

3.6.1.2 Active BDM Enabled in Stop3 Mode

Entry into the active background mode from run mode is enabled if ENBDM in BDCSCR is set. This register is described in Chapter 17, "Development Support." If ENBDM is set when the CPU executes a STOP instruction, the system clocks to the background debug logic remain active when the MCU enters stop mode. Because of this, background debug communication remains possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation.

Most background commands are not available in stop mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in either stop or wait mode. The BACKGROUND command can be used to wake the MCU from stop and enter active background mode if the ENBDM bit is set. After entering background debug mode, all background commands are available.

3.6.2 Stop2 Mode

Stop2 mode is entered by executing a STOP instruction under the conditions as shown in Table 3-1. Most of the internal circuitry of the MCU is powered off in stop2 with the exception of the RAM. Upon entering stop2, all I/O pin control signals are latched so that the pins retain their states during stop2.

Exit from stop2 is performed by asserting $\overline{\text{RESET}}$. On 3M05C or older masksets only, exit from stop2 can also be performed by asserting PTA7/ADP7/IRQ.

NOTE

On 3M05C or older masksets only, PTA7/ADP7/IRQ is an active low wake-up and must be configured as an input prior to executing a STOP instruction to avoid an immediate exit from stop2. PTA7/ADP7/IRQ can be disabled as a wake-up if it is configured as a high driven output. For lowest power consumption in stop2, this pin should not be left open when configured as input (enable the internal pullup; or tie an external pullup/down device; or set pin as output).

In addition, the real-time counter (RTC) can wake the MCU from stop2, if enabled.

Upon wake-up from stop2 mode, the MCU starts up as from a power-on reset (POR):

- All module control and status registers are reset
- The LVD reset function is enabled and the MCU remains in the reset state if V_{DD} is below the LVD trip point (low trip point selected due to POR)
- The CPU takes the reset vector

In addition to the above, upon waking up from stop2, the PPDF bit in SPMSC2 is set. This flag is used to direct user code to go to a stop2 recovery routine. PPDF remains set and the I/O pin states remain latched until a 1 is written to PPDACK in SPMSC2.

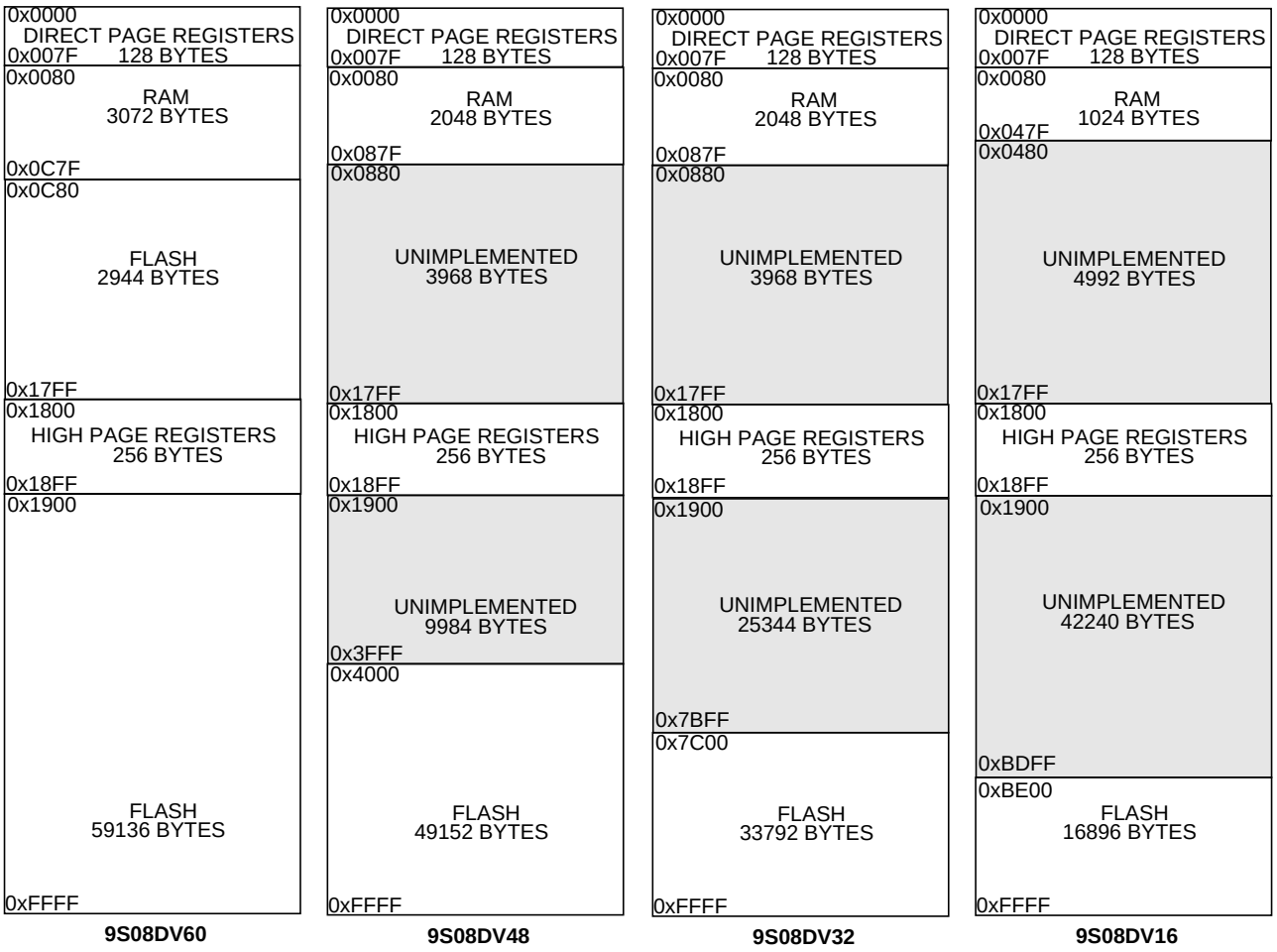

Figure 4-1. MC9S08DV60 Series Memory Map

4.2 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the MC9S08DV60 Series equate file provided by Freescale Semiconductor.

Table 4-1. Reset and Interrupt Vectors

Address (High/Low)	Vector	Vector Name
0xFFC0:0xFFC1	ACMP2	Vacmp2
0xFFC2:0xFFC3	ACMP1	Vacmp1
0xFFC4:0xFFC5	MSCAN Transmit	Vcantx
0xFFC6:0xFFC7	MSCAN Receive	Vcanrx
0xFFC8:0xFFC9	MSCAN errors	Vcanerr
0xFFCA:0xFFCB	MSCAN wake up	Vcanwu
0xFFCC:0xFFCD	RTC	Vrtc

An output pin can be selected to have high output drive strength by setting the corresponding bit in the drive strength select register (PTxDSn). When high drive is selected, a pin is capable of sourcing and sinking greater current. Even though every I/O pin can be selected as high drive, the user must ensure that the total current source and sink limits for the MCU are not exceeded. Drive strength selection is intended to affect the DC behavior of I/O pins. However, the AC behavior is also affected. High drive allows a pin to drive a greater load with the same switching speed as a low drive enabled pin into a smaller load. Because of this, the EMC emissions may be affected by enabling pins as high drive.

6.3 Pin Interrupts

Port A, port B, and port D pins can be configured as external interrupt inputs and as an external means of waking the MCU from stop or wait low-power modes.

The block diagram for each port interrupt logic is shown Figure 6-2.

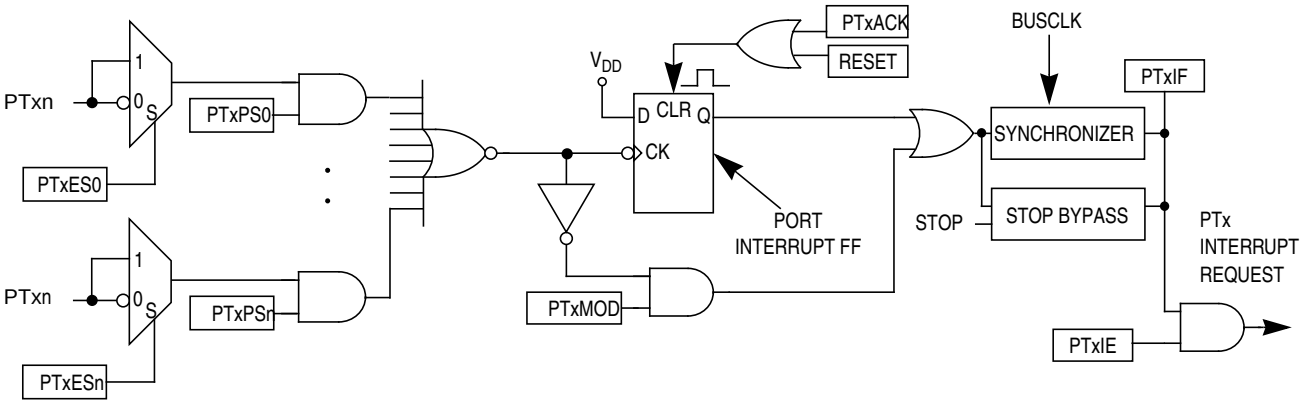

Figure 6-2. Port Interrupt Block Diagram

Writing to the PTxPSn bits in the port interrupt pin select register (PTxPS) independently enables or disables each port pin. Each port can be configured as edge sensitive or edge and level sensitive based on the PTxMOD bit in the port interrupt status and control register (PTxSC). Edge sensitivity can be software programmed to be either falling or rising; the level can be either low or high. The polarity of the edge or edge and level sensitivity is selected using the PTxESn bits in the port interrupt edge select register (PTxES).

Synchronous logic is used to detect edges. Prior to detecting an edge, enabled port inputs must be at the deasserted logic level. A falling edge is detected when an enabled port input signal is seen as a logic 1 (the deasserted level) during one bus cycle and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input signal is seen as a logic 0 during one bus cycle and then a logic 1 during the next cycle.

6.3.1 Edge Only Sensitivity

A valid edge on an enabled port pin will set PTxIF in PTxSC. If PTxIE in PTxSC is set, an interrupt request will be presented to the CPU. Clearing of PTxIF is accomplished by writing a 1 to PTxACK in PTxSC.

7.3.6.7 SP-Relative, 16-Bit Offset (SP2)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.4 Special Operations

The CPU performs a few special operations that are similar to instructions but do not have opcodes like other CPU instructions. In addition, a few instructions such as STOP and WAIT directly affect other MCU circuitry. This section provides additional information about these operations.

7.4.1 Reset Sequence

Reset can be caused by a power-on-reset (POR) event, internal conditions such as the COP (computer operating properly) watchdog, or by assertion of an external active-low reset pin. When a reset event occurs, the CPU immediately stops whatever it is doing (the MCU does not wait for an instruction boundary before responding to a reset event). For a more detailed discussion about how the MCU recognizes resets and determines the source, refer to the Resets, Interrupts, and System Configuration chapter.

The reset event is considered concluded when the sequence to determine whether the reset came from an internal source is done and when the reset pin is no longer asserted. At the conclusion of a reset event, the CPU performs a 6-cycle sequence to fetch the reset vector from 0xFFFE and 0xFFFF and to fill the instruction queue in preparation for execution of the first program instruction.

7.4.2 Interrupt Sequence

When an interrupt is requested, the CPU completes the current instruction before responding to the interrupt. At this point, the program counter is pointing at the start of the next instruction, which is where the CPU should return after servicing the interrupt. The CPU responds to an interrupt by performing the same sequence of operations as for a software interrupt (SWI) instruction, except the address used for the vector fetch is determined by the highest priority interrupt that is pending when the interrupt sequence started.

The CPU sequence for an interrupt is:

1. Store the contents of PCL, PCH, X, A, and CCR on the stack, in that order.
2. Set the I bit in the CCR.
3. Fetch the high-order half of the interrupt vector.
4. Fetch the low-order half of the interrupt vector.
5. Delay for one free bus cycle.
6. Fetch three bytes of program information starting at the address indicated by the interrupt vector to fill the instruction queue in preparation for execution of the first instruction in the interrupt service routine.

After the CCR contents are pushed onto the stack, the I bit in the CCR is set to prevent other interrupts while in the interrupt service routine. Although it is possible to clear the I bit with an instruction in the

c) MCGC1 = 0xB8 (%10111000)
   - 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 %111, or divide-by-128 because 4 MHz / 128 = 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

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

e) Loop until CLKST (bits 3 and 2) in MCGSC are %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:

a) BLPE: If a transition through BLPE mode is desired, first set LP (bit 3) in MCGC2 to 1.

b) BLPE/PBE: MCGC1 = 0x90 (%10010000)
   - RDIV (bits 5-3) set to %010, or divide-by-4 because 4 MHz / 4 = 1 MHz which is in the 1 MHz to 2 MHz range required by the PLL. In BLPE mode, the configuration of the RDIV does not matter because both the FLL and PLL are disabled. Changing them only sets up the the dividers for PLL usage in PBE mode

c) BLPE/PBE: MCGC3 = 0x44 (%01000100)
   - PLLS (bit 6) set to 1, selects the PLL. In BLPE mode, changing this bit only prepares the MCG for PLL usage in PBE mode
   - VDIV (bits 3-0) set to %0100, or multiply-by-16 because 1 MHz reference * 16 = 16 MHz. In BLPE mode, the configuration of the VDIV bits does not matter because the PLL is disabled. Changing them only sets up the multiply value for PLL usage in PBE mode

d) BLPE: If transitioning through BLPE mode, clear LP (bit 3) in MCGC2 to 0 here to switch to PBE mode

e) PBE: Loop until PLLST (bit 5) in MCGSC is set, indicating that the current source for the PLLS clock is the PLL

f) PBE: Then loop until LOCK (bit 6) in MCGSC is set, indicating that the PLL has acquired lock

3. Last, PBE mode transitions into PEE mode:

a) MCGC1 = 0x10 (%00010000)
   - CLKS (bits7 and 6) in MCGSC1 set to %00 in order to select the output of the PLL as the system clock source

b) Loop until CLKST (bits 3 and 2) in MCGSC are %11, indicating that the PLL output is selected to feed MCGOUT in the current clock mode
   - Now, With an RDIV of divide-by-4, a BDIV of divide-by-1, and a VDIV of multiply-by-16, MCGOUT = [(4 MHz / 4) * 16] / 1 = 16 MHz, and the bus frequency is MCGOUT / 2, or 8 MHz

c) MCGC1 = 0x98 (%10011000)
   – RDIV (bits 5-3) set to %011, or divide-by-8 because 8 MHz / 8= 1 MHz which is in the 1 MHz to 2 MHz range required by the PLL. In BLPE mode, the configuration of the RDIV does not matter because both the FLL and PLL are disabled. Changing them only sets up the the dividers for PLL usage in PBE mode

d) MCGC3 = 0x44 (%01000100)
   – PLLS (bit 6) set to 1, selects the PLL. In BLPE mode, changing this bit only prepares the MCG for PLL usage in PBE mode
   – VDIV (bits 3-0) set to %0100, or multiply-by-16 because 1 MHz reference * 16 = 16 MHz. In BLPE mode, the configuration of the VDIV bits does not matter because the PLL is disabled. Changing them only sets up the multiply value for PLL usage in PBE mode

e) Loop until PLLST (bit 5) in MCGSC is set, indicating that the current source for the PLLS clock is the PLL

3. Then, BLPE mode transitions into PBE mode:
   a) Clear LP (bit 3) in MCGC2 to 0 here to switch to PBE mode
   b) Then loop until LOCK (bit 6) in MCGSC is set, indicating that the PLL has acquired lock
4. Last, PBE mode transitions into PEE mode:
   a) MCGC1 = 0x18 (%00011000)
      – CLKS (bits7 and 6) in MCGSC1 set to %00 in order to select the output of the PLL as the system clock source
   b) Loop until CLKST (bits 3 and 2) in MCGSC are %11, indicating that the PLL output is selected to feed MCGOUT in the current clock mode
      – Now, With an RDIV of divide-by-8, a BDIV of divide-by-1, and a VDIV of multiply-by-16, MCGOUT = [(8 MHz / 8) * 16] / 1 = 16 MHz, and the bus frequency is MCGOUT / 2, or 8 MHz

Chapter 10
Analog-to-Digital Converter (S08ADC12V1)

10.1 Introduction

The 12-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

MC9S08DV60 Series devices operate at a higher voltage range (2.7 V to 5.5 V) and do not include stop1 mode. Please ignore references to stop1.

10.1.1 Analog Power and Ground Signal Names

References to V_{DDAD} and V_{SSAD} in this chapter correspond to signals V_{DDA} and V_{SSA}, respectively.

10.1.2 Channel Assignments

NOTE

The ADC channel assignments for the MC9S08DV60 Series devices are shown in Table 10-1. Reserved channels convert to an unknown value.

This chapter shows bits for all S08ADC12V1 channels. MC9S08DV60 Series MCUs do not use all of these channels. All bits corresponding to channels that are not available on a device are reserved.

Table 10-1. ADC Channel Assignment

ADCH	Channel	Input
00000	AD0	PTA0/ADP0/MCLK
00001	AD1	PTA1/ADP1/ACMP1+
00010	AD2	PTA2/ADP2/ACMP1P-
00011	AD3	PTA3/ADP3/ACMP1O
00100	AD4	PTA4/ADP4
00101	AD5	PTA5/ADP5
00110	AD6	PTA6/ADP6
00111	AD7	PTA7/ADP7
01000	AD8	PTB0/ADP8
01001	AD9	PTB1/ADP9
01010	AD10	PTB2/ADP10
01011	AD11	PTB3/ADP11
01100	AD12	PTB4/ADP12
01101	AD13	PTB5/ADP13
01110	AD14	PTB6/ADP14
01111	AD15	PTB7/ADP15
10000–11001	AD16 through AD25	Reserved
11010	AD26	Temperature Sensor[1]
11011	AD27	Internal Bandgap[2]
11100	Reserved	Reserved
11101	V_{REFH}	V_{REFH}
11110	V_{REFL}	V_{REFL}
11111	Module Disabled	None

Notes:
1 For information, see Section 10.1.5, “Temperature Sensor”.

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 MC9S08DV60 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). The RTC counter can be clocked by either MCGERCLK or a nominal 1 kHz clock source.

The period of the RTC is determined by the input clock frequency, the RTCPS bits, and the RTCMOD register. When the ADC hardware trigger is enabled, a conversion is initiated upon an RTC counter overflow.

The RTC can be configured to cause a hardware trigger in MCU run, wait, and stop3.

10.6 Application Information

This section contains information for using the ADC module in applications. The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an A/D converter.

10.6.1 External Pins and Routing

The following sections discuss the external pins associated with the ADC module and how they should be used for best results.

10.6.1.1 Analog Supply Pins

The ADC module has analog power and ground supplies (V_{DDAD} and V_{SSAD}) available as separate pins on some devices. V_{SSAD} is shared on the same pin as the MCU digital V_{SS} on some devices. On other devices, V_{SSAD} and V_{DDAD} are shared with the MCU digital supply pins. In these cases, there are separate pads for the analog supplies bonded to the same pin as the corresponding digital supply so that some degree of isolation between the supplies is maintained.

When available on a separate pin, both V_{DDAD} and V_{SSAD} must be connected to the same voltage potential as their corresponding MCU digital supply (V_{DD} and V_{SS}) and must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

If separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the V_{SSAD} pin. This should be the only ground connection between these supplies if possible. The V_{SSAD} pin makes a good single point ground location.

10.6.1.2 Analog Reference Pins

In addition to the analog supplies, the ADC module has connections for two reference voltage inputs. The high reference is V_{REFH}, which may be shared on the same pin as V_{DDAD} on some devices. The low reference is V_{REFL}, which may be shared on the same pin as V_{SSAD} on some devices.

When available on a separate pin, V_{REFH} may be connected to the same potential as V_{DDAD}, or may be driven by an external source between the minimum V_{DDAD} spec and the V_{DDAD} potential (V_{REFH} must never exceed V_{DDAD}). When available on a separate pin, V_{REFL} must be connected to the same voltage potential as V_{SSAD}. V_{REFH} and V_{REFL} must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 µF capacitor with good high frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current causes a voltage drop that could result in conversion errors. Inductance in this path must be minimum (parasitic only).

Table 12-1. CANCTL0 Register Field Descriptions (continued)

Field	Description
1 SLPRQ[5]	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 12.5.5.4, "MSCAN Sleep Mode"). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 12.3.2, "MSCAN Control Register 1 (CANCTL1)"). SLPRQ cannot be set while the WUPIF flag is set (see Section 12.3.4.1, "MSCAN Receiver Flag Register (CANRFLG)"). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ[6,7]	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 12.5.5.5, "MSCAN Initialization Mode"). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 12.3.2, "MSCAN Control Register 1 (CANCTL1)"). The following registers enter their hard reset state and restore their default values: CANCTL0[8], CANRFLG[9], CANRIER[10], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

1 The MSCAN must be in normal mode for this bit to become set.

2 See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.

3 In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 12.5.5.2, "Operation in Wait Mode" and Section 12.5.5.3, "Operation in Stop Mode").

4 The CPU has to make sure that the WUPE bit and the WUPIE wake-up interrupt enable bit (see Section 12.3.5, "MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.

5 The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).

6 The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).

7 In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.

8 Not including WUPE, INITRQ, and SLPRQ.

9 TSTAT1 and TSTAT0 are not affected by initialization mode.

10 RSTAT1 and RSTAT0 are not affected by initialization mode.

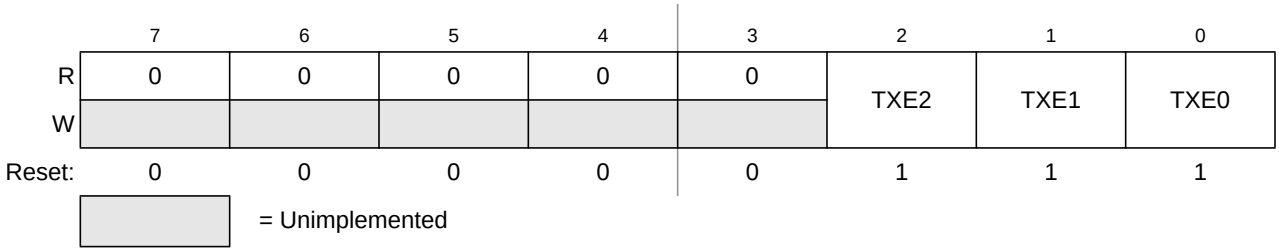

Figure 12-10. MSCAN Transmitter Flag Register (CANTFLG)

NOTE

The CANTFLG register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Read: Anytime
Write: Anytime for TXEx flags when not in initialization mode; write of 1 clears flag, write of 0 is ignored

Table 12-11. CANTFLG Register Field Descriptions

Field	Description
2:0 TXE[2:0]	**Transmitter Buffer Empty** — This flag indicates that the associated transmit message buffer is empty, and thus not scheduled for transmission. The CPU must clear the flag after a message is set up in the transmit buffer and is due for transmission. The MSCAN sets the flag after the message is sent successfully. The flag is also set by the MSCAN when the transmission request is successfully aborted due to a pending abort request (see Section 12.3.8, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). If not masked, a transmit interrupt is pending while this flag is set. Clearing a TXEx flag also clears the corresponding ABTAKx (see Section 12.3.9, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)"). When a TXEx flag is set, the corresponding ABTRQx bit is cleared (see Section 12.3.8, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). When listen-mode is active (see Section 12.3.2, "MSCAN Control Register 1 (CANCTL1)") the TXEx flags cannot be cleared and no transmission is started. Read and write accesses to the transmit buffer are blocked, if the corresponding TXEx bit is cleared (TXEx = 0) and the buffer is scheduled for transmission. 0 The associated message buffer is full (loaded with a message due for transmission) 1 The associated message buffer is empty (not scheduled)

12.3.7 MSCAN Transmitter Interrupt Enable Register (CANTIER)

This register contains the interrupt enable bits for the transmit buffer empty interrupt flags.

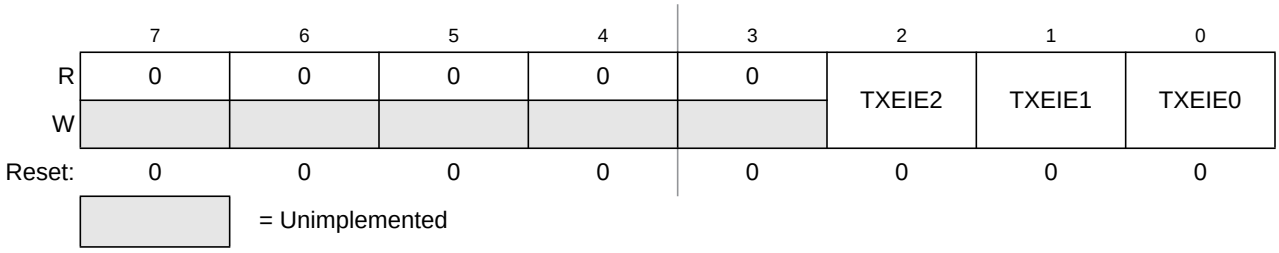

Figure 12-11. MSCAN Transmitter Interrupt Enable Register (CANTIER)

12.6.2 Bus-Off Recovery

The bus-off recovery is user configurable. The bus-off state can either be exited automatically or on user request.

For reasons of backwards compatibility, the MSCAN defaults to automatic recovery after reset. In this case, the MSCAN will become error active again after counting 128 occurrences of 11 consecutive recessive bits on the CAN bus (See the Bosch CAN specification for details).

If the MSCAN is configured for user request (BORM set in Section 12.3.2, "MSCAN Control Register 1 (CANCTL1)"), the recovery from bus-off starts after both independent events have become true:

- 128 occurrences of 11 consecutive recessive bits on the CAN bus have been monitored
- BOHOLD in Section 12.3.12, "MSCAN Miscellaneous Register (CANMISC) has been cleared by the user

These two events may occur in any order.

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 bus clock divided by four in slave mode.

All devices in the MC9S08DV60 Series MCUs contain one SPI module, as shown in the following block diagram.

NOTE

Ensure that the SPI should not be disabled (SPE=0) at the same time as a bit change to the CPHA bit. These changes should be performed as separate operations or unexpected behavior may occur.

Table 14-6. SCIxS1 Field Descriptions

Field	Description
7 TDRE	**Transmit Data Register Empty Flag** — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIxS1 with TDRE = 1 and then write to the SCI data register (SCIxD). 0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	**Transmission Complete Flag** — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete). TC is cleared automatically by reading SCIxS1 with TC = 1 and then doing one of the following three things: • Write to the SCI data register (SCIxD) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to SBK in SCIxC2
5 RDRF	**Receive Data Register Full Flag** — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SCIxD). To clear RDRF, read SCIxS1 with RDRF = 1 and then read the SCI data register (SCIxD). 0 Receive data register empty. 1 Receive data register full.
4 IDLE	**Idle Line Flag** — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line. To clear IDLE, read SCIxS1 with IDLE = 1 and then read the SCI data register (SCIxD). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE will get set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
3 OR	**Receiver Overrun Flag** — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SCIxD yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SCIxD. To clear OR, read SCIxS1 with OR = 1 and then read the SCI data register (SCIxD). 0 No overrun. 1 Receive overrun (new SCI data lost).
2 NF	**Noise Flag** — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF will be set at the same time as the flag RDRF gets set for the character. To clear NF, read SCIxS1 and then read the SCI data register (SCIxD). 0 No noise detected. 1 Noise detected in the received character in SCIxD.

14.3.5.2 Stop Mode Operation

During all stop modes, clocks to the SCI module are halted.

In stop1 and stop2 modes, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes. No SCI module registers are affected in stop3 mode.

The receive input active edge detect circuit is still active in stop3 mode, but not in stop2.. An active edge on the receive input brings the CPU out of stop3 mode if the interrupt is not masked (RXEDGIE = 1).

Note, because the clocks are halted, the SCI module will resume operation upon exit from stop (only in stop3 mode). Software should ensure stop mode is not entered while there is a character being transmitted out of or received into the SCI module.

14.3.5.3 Loop Mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general-purpose port I/O pin.

14.3.5.4 Single-Wire Operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general-purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIxC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters that are sent out by the transmitter.

16.6.2.1.2 Center-Aligned PWM Case

When CPWMS=1, TOF gets set when the timer counter changes direction from up-counting to down-counting at the end of the terminal count (the value in the modulo register). In this case the TOF corresponds to the end of a PWM period.

16.6.2.2 Channel Event Interrupt Description

The meaning of channel interrupts depends on the channel's current mode (input-capture, output-compare, edge-aligned PWM, or center-aligned PWM).

16.6.2.2.1 Input Capture Events

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select no edge (off), rising edges, falling edges or any edge as the edge which triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the two-step sequence described in Section 16.6.2, "Description of Interrupt Operation."

16.6.2.2.2 Output Compare Events

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the two-step sequence described Section 16.6.2, "Description of Interrupt Operation."

16.6.2.2.3 PWM End-of-Duty-Cycle Events

For channels configured for PWM operation there are two possibilities. When the channel is configured for edge-aligned PWM, the channel flag gets set when the timer counter matches the channel value register which marks the end of the active duty cycle period. When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle period which are the times when the timer counter matches the channel value register. The flag is cleared by the two-step sequence described Section 16.6.2, "Description of Interrupt Operation."

16.7 The Differences from TPM v2 to TPM v3

1. Write to TPMxCNTH:L registers (Section 16.3.2, "TPM-Counter Registers (TPMxCNTH:TPMxCNTL)) [SE110-TPM case 7]

   Any write to TPMxCNTH or TPMxCNTL registers in TPM v3 clears the TPM counter (TPMxCNTH:L) and the prescaler counter. Instead, in the TPM v2 only the TPM counter is cleared in this case.
2. Read of TPMxCNTH:L registers (Section 16.3.2, "TPM-Counter Registers (TPMxCNTH:TPMxCNTL))
   - In TPM v3, any read of TPMxCNTH:L registers during BDM mode returns the value of the TPM counter that is frozen. In TPM v2, if only one byte of the TPMxCNTH:L registers was read before the BDM mode became active, then any read of TPMxCNTH:L registers during