



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	16MHz
Connectivity	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	4
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08qd2j1csc

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



4.5.2	Program and Erase Times	37
4.5.3	Program and Erase Command Execution	38
4.5.4	Burst Program Execution	39
4.5.5	Access Errors	41
4.5.6	Flash Block Protection	42
4.5.7	Vector Redirection	43
Security	/	43
Flash R	egisters and Control Bits	44
4.7.1	Flash Clock Divider Register (FCDIV)	44
4.7.2	Flash Options Register (FOPT and NVOPT)	46
4.7.3	Flash Configuration Register (FCNFG)	47
4.7.4	Flash Protection Register (FPROT and NVPROT)	47
4.7.5	Flash Status Register (FSTAT)	48
4.7.6	Flash Command Register (FCMD)	49
	4.5.2 4.5.3 4.5.4 4.5.5 4.5.6 4.5.7 Security Flash R 4.7.1 4.7.2 4.7.3 4.7.4 4.7.5 4.7.6	 4.5.2 Program and Erase Times 4.5.3 Program and Erase Command Execution 4.5.4 Burst Program Execution 4.5.5 Access Errors 4.5.6 Flash Block Protection 4.5.7 Vector Redirection Security Flash Registers and Control Bits 4.7.1 Flash Clock Divider Register (FCDIV) 4.7.2 Flash Options Register (FOPT and NVOPT) 4.7.3 Flash Configuration Register (FCNFG) 4.7.4 Flash Protection Register (FSTAT) 4.7.6 Flash Command Register (FCMD)

Chapter 5 Resets, Interrupts, and General System Control

5.1	Introdu	ction	51
5.2	Feature	S	51
5.3	MCU R	eset	51
5.4	Comput	ter Operating Properly (COP) Watchdog	
5.5	Interrup	ts	53
	5.5.1	Interrupt Stack Frame	54
	5.5.2	External Interrupt Request (IRQ) Pin	54
	5.5.3	Interrupt Vectors, Sources, and Local Masks	
5.6	Low-Vo	bltage Detect (LVD) System	56
	5.6.1	Power-On Reset Operation	
	5.6.2	LVD Reset Operation	
	5.6.3	LVD Interrupt Operation	
	5.6.4	Low-Voltage Warning (LVW)	
5.7	Real-Ti	me Interrupt (RTI)	
5.8	Reset, I	nterrupt, and System Control Registers and Control Bits	
	5.8.1	Interrupt Pin Request Status and Control Register (IRQSC)	
	5.8.2	System Reset Status Register (SRS)	59
	5.8.3	System Background Debug Force Reset Register (SBDFR)	60
	5.8.4	System Options Register 1 (SOPT1)	61
	5.8.5	System Options Register 2 (SOPT2)	62
	5.8.6	System Device Identification Register (SDIDH, SDIDL)	62
	5.8.7	System Real-Time Interrupt Status and Control Register (SRTISC)	63
	5.8.8	System Power Management Status and Control 1 Register (SPMSC1)	64
	5.8.9	System Power Management Status and Control 2 Register (SPMSC2)	65

Chapter 6 Parallel Input/Output Control

6.1	Port Data and Data Direction	



Chapter 12 Development Support

12.1	Introduction	159
	12.1.1 Forcing Active Background	159
	12.1.2 Module Configuration	159
	12.1.3 Features	160
12.2	Background Debug Controller (BDC)	160
	12.2.1 BKGD Pin Description	161
	12.2.2 Communication Details	161
	12.2.3 BDC Commands	164
	12.2.4 BDC Hardware Breakpoint	167
12.3	Register Definition	167
	12.3.1 BDC Registers and Control Bits	168
	12.3.2 System Background Debug Force Reset Register (SBDFR)	170

Appendix A Electrical Characteristics

A.1	Introduction	173
A.2	Absolute Maximum Ratings	173
A.3	Thermal Characteristics	174
A.4	ESD Protection and Latch-Up Immunity	175
A.5	DC Characteristics	175
A.6	Supply Current Characteristics	
A.7	Internal Clock Source Characteristics	
A.8	AC Characteristics	
	A.8.1 Control Timing	
	A.8.2 Timer/PWM (TPM) Module Timing	
A.9	ADC Characteristics	
A.10	Flash Specifications	

Appendix B Ordering Information and Mechanical Drawings

B .1	Ordering Information	191
	B.1.1 Device Numbering Scheme	
B.2	Mechanical Drawings	



Chapter 3 Modes of Operation

3.1 Introduction

The operating modes of the MC9S08QD4 series are described in this chapter. Entry into each mode, exit from each mode, and functionality while in each of the modes are described.

3.2 Features

- Active background mode for code development
- Wait mode:
 - CPU shuts down to conserve power
 - System clocks running
 - Full voltage regulation maintained
- Stop modes:
 - CPU and bus clocks stopped
 - Stop2 Partial power down of internal circuits, RAM contents retained
 - Stop3 All internal circuits powered for fast recovery

3.3 Run Mode

This is the normal operating mode for the MC9S08QD4 series. This mode is selected when the BKGD/MS pin is high at the rising edge of reset. In this mode, the CPU executes code from internal memory with execution beginning at the address fetched from memory at 0xFFFE:0xFFFF after reset.

3.4 Active Background Mode

The active background mode functions are managed through the background debug controller (BDC) in the HCS08 core. The BDC provides the means for analyzing MCU operation during software development.

Active background mode is entered in any of five ways:

- When the BKGD/MS pin is low at the rising edge of reset
- When a BACKGROUND command is received through the BKGD pin
- When a BGND instruction is executed
- When encountering a BDC breakpoint



Chapter 3 Modes of Operation

After waking up from stop2, the PPDF bit in SPMSC2 is set. This flag may be 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 logic 1 is written to PPDACK in SPMSC2.

To maintain I/O state for pins that were configured as general-purpose I/O, the user must restore the contents of the I/O port registers, which have been saved in RAM, to the port registers before writing to the PPDACK bit. If the port registers are not restored from RAM before writing to PPDACK, then the register bits will assume their reset states when the I/O pin latches are opened and the I/O pins will switch to their reset states.

For pins that were configured as peripheral I/O, the user must reconfigure the peripheral module that interfaces to the pin before writing to the PPDACK bit. If the peripheral module is not enabled before writing to PPDACK, the pins will be controlled by their associated port control registers when the I/O latches are opened.

3.6.2 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.

Stop3 can be exited by asserting $\overline{\text{RESET}}$, or by an interrupt from one of the following sources: the real-time interrupt (RTI), LVD, ADC, IRQ, or the KBI.

If stop3 is exited by means of the RESET pin, then the MCU is reset and operation will resume after taking the reset vector. Exit by means of one of the internal interrupt sources results in the MCU taking the appropriate interrupt vector.

3.6.3 Active BDM Enabled in Stop Mode

Entry into the active background mode from run mode is enabled if the ENBDM bit in BDCSCR is set. This register is described in Chapter 12, "Development Support," of this data sheet. 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 so background debug communication is still possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation. If the user attempts to enter stop2 with ENBDM set, the MCU will instead enter stop3.

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. Table 3-2 summarizes the behavior of the MCU in stop when entry into the background debug mode is enabled.



Chapter 4 Memory Map and Register Definition

program command is issued, the charge pump is enabled and then remains enabled after completion of the burst program operation if these two conditions are met:

- The next burst program command has been queued before the current program operation has completed.
- The next sequential address selects a byte on the same physical row as the current byte being programmed. A row of flash memory consists of 64 bytes. A byte within a row is selected by addresses A5 through A0. A new row begins when addresses A5 through A0 are all 0s.

The first byte of a series of sequential bytes being programmed in burst mode will take the same amount of time to program as a byte programmed in standard mode. Subsequent bytes will program in the burst program time provided that the conditions above are met. In the case the next sequential address is the beginning of a new row, the program time for that byte will be the standard time instead of the burst time. This is because the high voltage to the array must be disabled and then enabled again. If a new burst command has not been queued before the current command completes, then the charge pump will be disabled and high voltage removed from the array.



Field	Description
5 COP	 Computer Operating Properly (COP) Watchdog — Reset was caused by the COP watchdog timer timing out. This reset source can be blocked by COPE = 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
1 LVD	 Low Voltage Detect — If the LVDRE bit is set and the supply drops below the LVD trip voltage, an LVD reset will occur. This bit is also set by POR. 0 Reset not caused by LVD trip or POR. 1 Reset caused by LVD trip or POR.

Table 5-4. SRS Register Field Descriptions (continued)

5.8.3 System Background Debug Force Reset Register (SBDFR)

This high-page register contains a single write-only control bit. A serial background command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return 0x00.



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

Figure 5-4. System Background Debug Force Reset Register (SBDFR)

Table 5-5. SBDFR Register Field Descriptions

Field	Description
0 BDFR	Background Debug Force Reset — A serial background command such as WRITE_BYTE can be used to allow 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. To enter user mode, PTA4/TPM2CH0O/BKGD/MS must be high immediately after issuing WRITE_BYTE command. To enter BDM, PTA4/TPM2CH0O/BKGD/MS must be low immediately after issuing WRITE_BYTE command. See A.8.1, "Control Timing," for more information.



Chapter 6 Parallel Input/Output Control

This section explains software controls related to parallel input/output (I/O) and pin control. The MC9S08QD4 series has one parallel I/O port which include a total of 4 I/O pins, one output-only pin, and one input-only pin. See Section Chapter 2, "External Signal Description," for more information about pin assignments and external hardware considerations of these pins.

All of these I/O pins are shared with on-chip peripheral functions as shown in Table 2-1. The peripheral modules have priority over the I/Os so that when a peripheral is enabled, the I/O functions associated with the shared pins are disabled. After reset, the shared peripheral functions are disabled so that the pins are controlled by the I/O. All of the I/Os are configured as inputs (PTxDDn = 0) with pullup devices disabled (PTxPEn = 0), except for output-only pin PTA4 which defaults to BKGD/MS pin.

NOTE

Not all general-purpose I/O pins are available on all packages. To avoid extra current drain from floating input pins, the user's reset initialization routine in the application program must either enable on-chip pullup devices or change the direction of unconnected pins to outputs so the pins do not float.

6.1 Port Data and Data Direction

Reading and writing of parallel I/Os is performed through the port data registers. The direction, either input or output, is controlled through the port data direction registers. The parallel I/O port function for an individual pin is illustrated in the block diagram shown in Figure 6-1.



Source Form	Operation	ress de	Object Code	les	Cyc-by-Cyc Details	Affect on CCR	
		Add Moc		Cyc		VH	INZC
RSP	Reset Stack Pointer (Low Byte) SPL ← \$FF (High Byte Not Affected)	INH	9C	1	q		
RTI	Return from Interrupt SP ← (SP) + \$0001; Pull (CCR) SP ← (SP) + \$0001; Pull (A) SP ← (SP) + \$0001; Pull (X) SP ← (SP) + \$0001; Pull (PCH) SP ← (SP) + \$0001; Pull (PCL)	INH	80	9	սսսսեքքք	11	\$\$\$
RTS	Return from Subroutine SP \leftarrow SP + \$0001; Pull (PCH) SP \leftarrow SP + \$0001; Pull (PCL)	INH	81	5	ufppp		
SBC #opr8i SBC opr8a SBC opr16a SBC oprx16,X SBC oprx8,X SBC ,X SBC oprx16,SP SBC oprx8,SP	Subtract with Carry A \leftarrow (A) – (M) – (C)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 ii B2 dd C2 hh 11 D2 ee ff E2 ff F2 9E D2 ee ff 9E E2 ff	2 3 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↓ -	-\$\$\$
SEC	Set Carry Bit $(C \leftarrow 1)$	INH	99	1	q		1
SEI	Set Interrupt Mask Bit $(I \leftarrow 1)$	INH	9B	1	q		1 – – –
STA opr8a STA opr16a STA oprx16,X STA oprx8,X STA ,X STA oprx16,SP STA oprx8,SP	Store Accumulator in Memory M ← (A)	DIR EXT IX2 IX1 IX SP2 SP1	B7 dd C7 hh ll D7 ee ff E7 ff F7 9E D7 ee ff 9E E7 ff	3 4 3 2 5 4	wpp pwpp wp powpp pwpp	0 –	- ‡ ‡ -
STHX opr8a STHX opr16a STHX oprx8,SP	Store H:X (Index Reg.) (M:M + \$0001) ← (H:X)	DIR EXT SP1	35 dd 96 hh ll 9E FF ff	4 5 5	wwpp pwwpp	0 –	-\$\$-
STOP	Enable Interrupts: Stop Processing Refer to MCU Documentation I bit \leftarrow 0; Stop Processing	INH	8E	2	fp		0
STX opr8a STX opr16a STX oprx16,X STX oprx8,X STX ,X STX oprx16,SP STX oprx8,SP	Store X (Low 8 Bits of Index Register) in Memory $M \leftarrow (X)$	DIR EXT IX2 IX1 IX SP2 SP1	BF dd CF hh ll DF ee ff EF ff FF 9E DF ee ff 9E EF ff	3 4 3 2 5 4	wpp pwpp pwpp pwpp	0 –	- ‡ ‡ -



Chapter 7 Central Processor Unit (S08CPUV2)

Bit-Mani	pulation	ulation Branch Read-Modify-Write Control Register/Memory									
				9E60 6 NEG 3 SP1				2	9ED0 5 SUB 4 SP2	9EE0 4 SUB 3 SP1	
				9E61 6 CBEQ 4 SP1				2	DED1 5 CMP 4 SP2	9EE1 4 CMP 3 SP1	
								2	9ED2 5 SBC 4 SP2	9EE2 4 SBC 3 SP1	
				9E63 6 COM 3 SP1				2	9ED3 5 CPX 4 SP2	9EE3 4 CPX 3 SP1	9EF3 6 CPHX 3 SP1
				9E64 6 LSR 3 SP1				2	9ED4 5 AND 4 SP2	9EE4 4 AND 3 SP1	
								2	9ED5 5 BIT 4 SP2	9EE5 4 BIT 3 SP1	
				9E66 6 ROR 3 SP1				2	DED6 5 LDA 4 SP2	9EE6 4 LDA 3 SP1	
				9E67 6 ASR 3 SP1				2	9ED7 5 STA 4 SP2	9EE7 4 STA 3 SP1	
				9E68 6 LSL 3 SP1				2	ED8 5 EOR SP2	9EE8 4 EOR 3 SP1	
				9E69 6 ROL 3 SP1				2	ADC ADC SP2	9EE9 4 ADC 3 SP1	
				9E6A 6 DEC 3 SP1				2	OEDA 5 ORA 4 SP2	9EEA 4 ORA 3 SP1	
				9E6B 8 DBNZ 4 SP1				2	PEDB 5 ADD 4 SP2	9EEB 4 ADD 3 SP1	
				9E6C 6 INC 3 SP1							
				9E6D 5 TST 3 SP1							
						9EAE 5 LDHX 2 I>	9EBE 6 LDHX 4 IX2	9ECE 5 9 LDHX 3 IX1 4	DEDE 5 LDX 4 SP2	9EEE 4 LDX 3 SP1	9EFE 5 LDHX 3 SP1
				9E6F 6 CLR 3 SP1				9	9EDF 5 STX 4 SP2	9EEF 4 STX 3 SP1	9EFF 5 STHX 3 SP1

Table 7-3. Opcode Map (Sheet 2 of 2)

Inherent Immediate Direct Extended DIR to DIR IX+ to DIR REL IX IX1 IX2 IMD DIX+ INH IMM DIR EXT DD IX+D

Relative Indexed, No Offset Indexed, 8-Bit Offset Indexed, 16-Bit Offset IMM to DIR DIR to IX+

Stack Pointer, 8-Bit Offset Stack Pointer, 16-Bit Offset Indexed, No Offset with Post Increment Indexed, 1-Byte Offset with Post Increment

SP1 SP2 IX+

IX1+

Note: All Sheet 2 Opcodes are Preceded by the Page 2 Prebyte (9E)

Prebyte (9E) and Opcode in Hexadecimal 9E60 6 NEG Number of Bytes 3 SP1 Addressing Mode



Chapter 8 Analog-to-Digital Converter (ADC10V1)

8.1 Introduction

The 10-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

The ADC module design supports up to 28 separate analog inputs (AD0–AD27). Only four (ADC1P0–ADC1P3) of the possible inputs are implemented on the MC9S08QD4 series MCU. These inputs are selected by the ADCH bits.



Field	Description
1 ADPC9	 ADC Pin Control 9 — ADPC9 is used to control the pin associated with channel AD9. 0 AD9 pin I/O control enabled 1 AD9 pin I/O control disabled
0 ADPC8	 ADC Pin Control 8 — ADPC8 is used to control the pin associated with channel AD8. 0 AD8 pin I/O control enabled 1 AD8 pin I/O control disabled

Table 8-10. APCTL2 Register Field Descriptions (continued)

8.3.10 Pin Control 3 Register (APCTL3)

APCTL3 is used to control channels 16–23 of the ADC module.



Figure 8-13. Pin Control 3 Register (APCTL3)

Table 8-11. APCTL3 Register Field Descriptions

Field	Description
7 ADPC23	 ADC Pin Control 23 — ADPC23 is used to control the pin associated with channel AD23. 0 AD23 pin I/O control enabled 1 AD23 pin I/O control disabled
6 ADPC22	 ADC Pin Control 22 — ADPC22 is used to control the pin associated with channel AD22. AD22 pin I/O control enabled AD22 pin I/O control disabled
5 ADPC21	 ADC Pin Control 21 — ADPC21 is used to control the pin associated with channel AD21. 0 AD21 pin I/O control enabled 1 AD21 pin I/O control disabled
4 ADPC20	 ADC Pin Control 20 — ADPC20 is used to control the pin associated with channel AD20. 0 AD20 pin I/O control enabled 1 AD20 pin I/O control disabled
3 ADPC19	 ADC Pin Control 19 — ADPC19 is used to control the pin associated with channel AD19. 0 AD19 pin I/O control enabled 1 AD19 pin I/O control disabled
2 ADPC18	 ADC Pin Control 18 — ADPC18 is used to control the pin associated with channel AD18. 0 AD18 pin I/O control enabled 1 AD18 pin I/O control disabled



Analog-to-Digital Converter (S08ADC10V1)

8.4.4.2 Completing Conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, ADCRH and ADCRL. This is indicated by the setting of COCO. An interrupt is generated if AIEN is high at the time that COCO is set.

A blocking mechanism prevents a new result from overwriting previous data in ADCRH and ADCRL if the previous data is in the process of being read while in 10-bit MODE (the ADCRH register has been read but the ADCRL register has not). When blocking is active, the data transfer is blocked, COCO is not set, and the new result is lost. In the case of single conversions with the compare function enabled and the compare condition false, blocking has no effect and ADC operation is terminated. In all other cases of operation, when a data transfer is blocked, another conversion is initiated regardless of the state of ADCO (single or continuous conversions enabled).

If single conversions are enabled, the blocking mechanism could result in several discarded conversions and excess power consumption. To avoid this issue, the data registers must not be read after initiating a single conversion until the conversion completes.

8.4.4.3 Aborting Conversions

Any conversion in progress will be aborted when:

- A write to ADCSC1 occurs (the current conversion will be aborted and a new conversion will be initiated, if ADCH are not all 1s).
- A write to ADCSC2, ADCCFG, ADCCVH, or ADCCVL occurs. This indicates a mode of operation change has occurred and the current conversion is therefore invalid.
- The MCU is reset.
- The MCU enters stop mode with ADACK not enabled.

When a conversion is aborted, the contents of the data registers, ADCRH and ADCRL, are not altered but continue to be the values transferred after the completion of the last successful conversion. In the case that the conversion was aborted by a reset, ADCRH and ADCRL return to their reset states.

8.4.4.4 Power Control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, the ADACK clock generator is also enabled.

Power consumption when active can be reduced by setting ADLPC. This results in a lower maximum value for f_{ADCK} (see the electrical specifications).

8.4.4.5 Total Conversion Time

The total conversion time depends on the sample time (as determined by ADLSMP), the MCU bus frequency, the conversion mode (8-bit or 10-bit), and the frequency of the conversion clock (f_{ADCK}). After the module becomes active, sampling of the input begins. ADLSMP is used to select between short and long sample times. When sampling is complete, the converter is isolated from the input channel and a successive approximation algorithm is performed to determine the digital value of the analog signal. The



result of the conversion is transferred to ADCRH and ADCRL upon completion of the conversion algorithm.

If the bus frequency is less than the f_{ADCK} frequency, precise sample time for continuous conversions cannot be guaranteed when short sample is enabled (ADLSMP=0). If the bus frequency is less than 1/11th of the f_{ADCK} frequency, precise sample time for continuous conversions cannot be guaranteed when long sample is enabled (ADLSMP=1).

The maximum total conversion time for different conditions is summarized in Table 8-12.

T		1	
Conversion Type	ADICLK	ADLSMP	Max Total Conversion Time
Single or first continuous 8-bit	0x, 10	0	20 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit	0x, 10	0	23 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	0x, 10	1	40 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit	0x, 10	1	43 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	11	0	5 μ s + 20 ADCK + 5 bus clock cycles
Single or first continuous 10-bit	11	0	5 μ s + 23 ADCK + 5 bus clock cycles
Single or first continuous 8-bit	11	1	5 μ s + 40 ADCK + 5 bus clock cycles
Single or first continuous 10-bit	11	1	5 μ s + 43 ADCK + 5 bus clock cycles
Subsequent continuous 8-bit;	xx	0	17 ADCK cycles
$f_{BUS} \ge f_{ADCK}$			
Subsequent continuous 10-bit;	ХХ	0	20 ADCK cycles
$f_{BUS} \ge f_{ADCK}$			
Subsequent continuous 8-bit;	xx	1	37 ADCK cycles
$f_{BUS} \ge f_{ADCK}/11$			
Subsequent continuous 10-bit;	xx	1	40 ADCK cycles
$f_{BUS} \ge f_{ADCK}/11$			

Table 8-12. Total Conversion Time vs. Control Conditions

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by the ADICLK bits, and the divide ratio is specified by the ADIV bits. For example, in 10-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, and a bus frequency of 8 MHz, then the conversion time for a single conversion is:

Conversion time = $\frac{23 \text{ ADCK cyc}}{8 \text{ MHz/1}} + \frac{5 \text{ bus cyc}}{8 \text{ MHz}} = 3.5 \text{ }\mu\text{s}$

Number of bus cycles = $3.5 \ \mu s \ x \ 8 \ MHz = 28 \ cycles$

NOTE

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.



8.6.2 Sources of Error

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

8.6.2.1 Sampling Error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy. Given the maximum input resistance of approximately $7k\Omega$ and input capacitance of approximately 5.5 pF, sampling to within 1/4LSB (at 10-bit resolution) can be achieved within the minimum sample window (3.5 cycles @ 8 MHz maximum ADCK frequency) provided the resistance of the external analog source (R_{AS}) is kept below 5 kΩ.

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

8.6.2.2 Pin Leakage Error

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

8.6.2.3 Noise-Induced Errors

System noise which occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 μ F low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 μ F low-ESR capacitor from V_{DDAD} to V_{SSAD}.
- If inductive isolation is used from the primary supply, an additional 1 μ F capacitor is placed from V_{DDAD} to V_{SSAD}.
- V_{SSAD} (and V_{REFL} , if connected) is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in wait or stop3 mode before initiating (hardware triggered conversions) or immediately after initiating (hardware or software triggered conversions) the ADC conversion.
 - For software triggered conversions, immediately follow the write to the ADCSC1 with a WAIT instruction or STOP instruction.
 - For stop3 mode operation, select ADACK as the clock source. Operation in stop3 reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in wait or stop3 or I/O activity cannot be halted, these recommended actions may reduce the effect of noise on the accuracy:

• Place a 0.01 μ F capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSAD} (this will improve noise issues but will affect sample rate based on the external analog source resistance).



Analog-to-Digital Converter (S08ADC10V1)



9.3.3 ICS Trim Register (ICSTRM)



Figure 9-5. ICS Trim Register (ICSTRM)

Table 9-3. ICS Trim Register Field Descriptions

Field	Description
7:0 TRIM	ICS Trim Setting — The TRIM bits control the internal reference clock frequency by controlling the internal reference clock period. The bits' effect are binary weighted (i.e., bit 1 will adjust twice as much as bit 0). Increasing the binary value in TRIM will increase the period, and decreasing the value will decrease the period.
	An additional fine trim bit is available in ICSSC as the FTRIM bit.

9.3.4 ICS Status and Control (ICSSC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	CLK	ST	OSCINIT	ETDIM
w								
POR:	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	U

Figure 9-6. ICS Status and Control Register (ICSSC)

Table 9-4. ICS Status and Control Register Field Descriptions

Field	Description
7:4	Reserved, must be cleared.
3:2 CLKST	 Clock Mode Status — The CLKST bits indicate the current clock mode. The CLKST bits don't update immediately after a write to the CLKS bits due to internal synchronization between clock domains. Output of FLL is selected. FLL Bypassed, Internal reference clock is selected. FLL Bypassed, External reference clock is selected. Reserved.
1	OSC Initialization — If the external reference clock is selected by ERCLKEN or by the ICS being in FEE, FBE, or FBELP mode, and if EREFS is set, then this bit is set after the initialization cycles of the external oscillator clock have completed. This bit is cleared only when either ERCLKEN or EREFS are cleared.
0	ICS Fine Trim — The FTRIM bit controls the smallest adjustment of the internal reference clock frequency. Setting FTRIM will increase the period and clearing FTRIM will decrease the period by the smallest amount possible.



Chapter 11 Timer/Pulse-Width Modulator (S08TPMV2)



NOTES:

- ¹ Port pins are software configurable with pullup device if input port.
- ² Port pins are software configurable for output drive strength.
- ³ Port pins are software configurable for output slew rate control.
- ⁴ IRQ contains a software configurable (IRQPDD) pullup/pulldown device if PTA5 enabled as IRQ pin function (IRQPE = 1).
- ⁵ RESET contains integrated pullup device if PTA5 enabled as reset pin function (RSTPE = 1).
- ⁶ PTA5 does not contain a clamp diode to V_{DD} and must not be driven above V_{DD} . The voltage measured on this pin when internal pullup is enabled may be as low as $V_{DD} 0.7$ V. The internal gates connected to this pin are pulled to V_{DD} .
- 7 PTA4 contains integrated pullup device if BKGD enabled (BKGDPE = 1).
- ⁸ When pin functions as KBI (KBIPEn = 1) and associated pin is configured to enable the pullup device, KBEDGn can be used to reconfigure the pullup as a pulldown device.

Figure 11-1. MC9S08QD4 Series Block Diagram Highlighting TPM Block and Pins



Timer/Pulse-Width Modulator (S08TPMV2)

When center-aligned PWM operation is specified, the counter counts upward from 0x0000 through its terminal count and then counts downward to 0x0000 where it returns to up-counting. Both 0x0000 and the terminal count value (value in TPMxMODH:TPMxMODL) are normal length counts (one timer clock period long).

An interrupt flag and enable are associated with the main 16-bit counter. The timer overflow flag (TOF) is a software-accessible indication that the timer counter has overflowed. The enable signal selects between software polling (TOIE = 0) where no hardware interrupt is generated, or interrupt-driven operation (TOIE = 1) where a static hardware interrupt is automatically generated whenever the TOF flag is 1.

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the main 16-bit counter counts from 0x0000 through 0xFFFF and overflows to 0x0000 on the next counting clock. TOF becomes set at the transition from 0xFFFF to 0x0000. When a modulus limit is set, TOF becomes set at the transition from the value set in the modulus register to 0x0000. When the main 16-bit counter is operating in up-/down-counting mode, the TOF flag gets set as the counter changes direction at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The 0x0000 count value corresponds to the center of a period.)

Because the HCS08 MCU is an 8-bit architecture, a coherency mechanism is built into the timer counter for read operations. Whenever either byte of the counter is read (TPMxCNTH or TPMxCNTL), both bytes are captured into a buffer so when the other byte is read, the value will represent the other byte of the count at the time the first byte was read. The counter continues to count normally, but no new value can be read from either byte until both bytes of the old count have been read.

The main timer counter can be reset manually at any time by writing any value to either byte of the timer count TPMxCNTH or TPMxCNTL. Resetting the counter in this manner also resets the coherency mechanism in case only one byte of the counter was read before resetting the count.

11.4.2 Channel Mode Selection

Provided CPWMS = 0 (center-aligned PWM operation is not specified), the MSnB and MSnA control bits in the channel n status and control registers determine the basic mode of operation for the corresponding channel. Choices include input capture, output compare, and buffered edge-aligned PWM.

11.4.2.1 Input Capture Mode

With the input capture function, the TPM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TPM latches the contents of the TPM counter into the channel value registers (TPMxCnVH:TPMxCnVL). Rising edges, falling edges, or any edge may be chosen as the active edge that triggers an input capture.

When either byte of the 16-bit capture register is read, both bytes are latched into a buffer to support coherent 16-bit accesses regardless of order. The coherency sequence can be manually reset by writing to the channel status/control register (TPMxCnSC).

An input capture event sets a flag bit (CHnF) that can optionally generate a CPU interrupt request.

Field	Description
2 WS	 Wait or Stop Status — When the target CPU is in wait or stop mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of wait or stop and into active background mode where all BDC commands work. Whenever the host forces the target MCU into active background mode, the host must issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands. 0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active) 1 Target CPU is in wait or stop mode, or a BACKGROUND command was used to change from wait or stop to active background mode
1 WSF	 Wait or Stop Failure Status — This status bit is set if a memory access command failed due to the target CPU executing a wait or stop instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of wait or stop mode into active background mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the wait or stop instruction.) Memory access did not conflict with a wait or stop instruction Memory access command failed because the CPU entered wait or stop mode
0 DVF	Data Valid Failure Status — This status bit is not used in the MC9S08QD4 series because it does not have any slow access memory. 0 Memory access did not conflict with a slow memory access 1 Memory access command failed because CPU was not finished with a slow memory access

Table 12-2. BDCSCR Register Field Descriptions (continued)

12.3.1.2 BDC Breakpoint Match Register (BDCBKPT)

This 16-bit register holds the address for the hardware breakpoint in the BDC. The BKPTEN and FTS control bits in BDCSCR are used to enable and configure the breakpoint logic. Dedicated serial BDC commands (READ_BKPT and WRITE_BKPT) are used to read and write the BDCBKPT register but is not accessible to user programs because it is not located in the normal memory map of the MCU. Breakpoints are normally set while the target MCU is in active background mode before running the user application program. For additional information about setup and use of the hardware breakpoint logic in the BDC, refer to Section 12.2.4, "BDC Hardware Breakpoint."

12.3.2 System Background Debug Force Reset Register (SBDFR)

This register contains a single write-only control bit. A serial background mode command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return 0x00.



Development Support