

Welcome to [E-XFL.COM](http://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	16MHz
Connectivity	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	4
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08qd4j1mscr">https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08qd4j1mscr</a>

9.5	Module Initialization .....	132
9.5.1	ICS Module Initialization Sequence .....	132

## Chapter 10 Keyboard Interrupt (S08KBIV2)

10.1	Introduction .....	135
10.1.1	Features .....	137
10.1.2	Modes of Operation .....	137
10.1.3	Block Diagram .....	137
10.2	External Signal Description .....	138
10.3	Register Definition .....	138
10.3.1	KBI Status and Control Register (KBISC) .....	138
10.3.2	KBI Pin Enable Register (KBIPE) .....	139
10.3.3	KBI Edge Select Register (KBIES) .....	139
10.4	Functional Description .....	140
10.4.1	Edge Only Sensitivity .....	140
10.4.2	Edge and Level Sensitivity .....	140
10.4.3	KBI Pullup/Pulldown Resistors .....	141
10.4.4	KBI Initialization .....	141

## Chapter 11 Timer/Pulse-Width Modulator (S08TPMV2)

11.1	Introduction .....	143
11.1.1	TPM2 Configuration Information .....	143
11.1.2	TCLK1 and TCLK2 Configuration Information .....	143
11.1.3	Features .....	145
11.1.4	Block Diagram .....	145
11.2	External Signal Description .....	147
11.2.1	External TPM Clock Sources .....	147
11.2.2	TPMxCHn — TPMx Channel n I/O Pins .....	147
11.3	Register Definition .....	147
11.3.1	Timer Status and Control Register (TPMxSC) .....	148
11.3.2	Timer Counter Registers (TPMxCNTH:TPMxCNTL) .....	149
11.3.3	Timer Counter Modulo Registers (TPMxMODH:TPMxMODL) .....	150
11.3.4	Timer Channel n Status and Control Register (TPMxCnSC) .....	151
11.3.5	Timer Channel Value Registers (TPMxCnVH:TPMxCnVL) .....	152
11.4	Functional Description .....	153
11.4.1	Counter .....	153
11.4.2	Channel Mode Selection .....	154
11.4.3	Center-Aligned PWM Mode .....	156
11.5	TPM Interrupts .....	157
11.5.1	Clearing Timer Interrupt Flags .....	157
11.5.2	Timer Overflow Interrupt Description .....	157
11.5.3	Channel Event Interrupt Description .....	158
11.5.4	PWM End-of-Duty-Cycle Events .....	158

**Table 4-4. Nonvolatile Register Summary**

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0xFFAA – 0xFFAC	Reserved	—	—	—	—	—	—	—	—
0xFFAD	Reserved for ADCRL of AD26 value during ICS trim	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0xFFAE	Reserved for ADCRH of AD26 value during ICS trim and ICS Trim value “FTRIM”	ADR9	ADR8	—	—	—	—	—	FTRIM
0xFFAF	Reserved for ICS Trim value “TRIM”	TRIM							
0xFFB0 – 0xFFB7	<b>NVBACKKEY</b>	8-Byte Comparison Key							
0xFFB8 – 0xFFBC	Reserved	—	—	—	—	—	—	—	—
0xFFBD	<b>NVPROT</b>	FPS							
0xFFBE	Unused	—	—	—	—	—	—	—	—
0xFFBF	<b>NVOPT</b>	KEYEN	FNORED	0	0	0	0	SEC01	SEC00

Provided the key enable (KEYEN) bit is 1, the 8-byte comparison key can be used to temporarily disengage memory security. This key mechanism can be accessed only through user code running in secure memory. (A security key cannot be entered directly through background debug commands.) This security key can be disabled completely by programming the KEYEN bit to 0. If the security key is disabled, the only way to disengage security is by mass erasing the flash if needed (normally through the background debug interface) and verifying that flash is blank. To avoid returning to secure mode after the next reset, program the security bits (SEC01:SEC00) to the unsecured state (1:0).

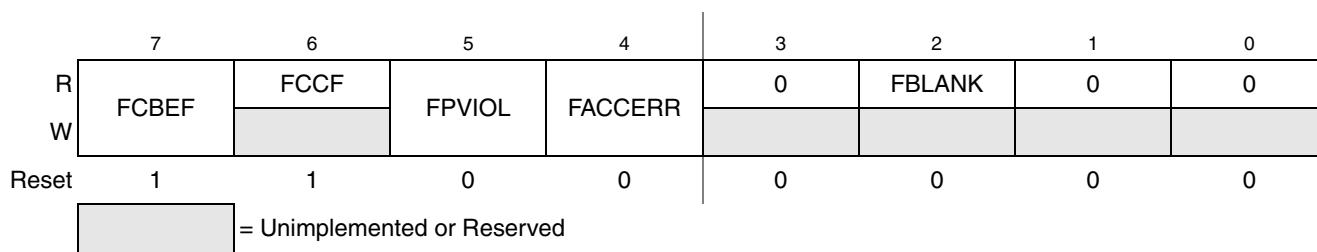
The ICS factory-trimmed value will be stored in 0xFFAE (bit-0) and 0xFFAF. Development tools, such as programmers can trim the ICS and the internal temperature sensor (via the ADC) and store the values in 0xFFAD–0xFFAF.

## 4.4 RAM

The MC9S08QD4 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 when the MCU is in low-power wait, stop2, or stop3 mode. At power-on or after wakeup from stop1, the contents of RAM are uninitialized. RAM data is unaffected by any reset provided that the supply voltage does not drop below the minimum value for RAM retention ( $V_{RAM}$ ).

### 4.7.5 Flash Status Register (FSTAT)



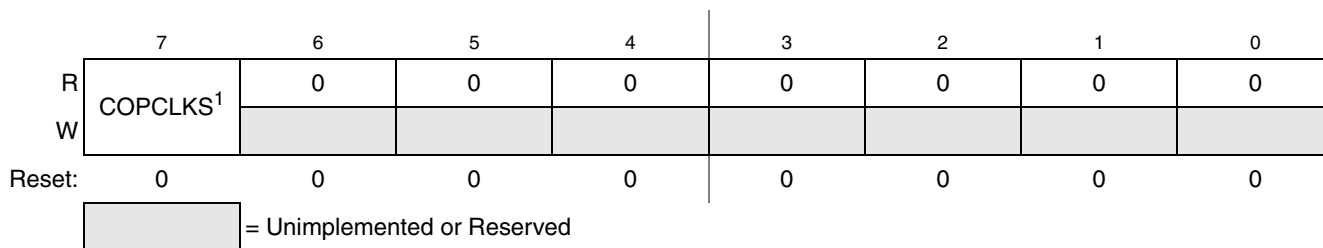
**Figure 4-9. Flash Status Register (FSTAT)**

**Table 4-12. FSTAT Register Field Descriptions**

Field	Description
7 FCBEF	<p><b>Flash Command Buffer Empty Flag</b> — The FCBEF bit is used to launch commands. It also indicates that the command buffer is empty so that a new command sequence can be executed when performing burst programming. The FCBEF bit is cleared by writing a 1 to it or when a burst program command is transferred to the array for programming. Only burst program commands can be buffered.</p> <p>0 Command buffer is full (not ready for additional commands). 1 A new burst program command can be written to the command buffer.</p>
6 FCCF	<p><b>Flash Command Complete Flag</b> — FCCF is set automatically when the command buffer is empty and no command is being processed. FCCF is cleared automatically when a new command is started (by writing 1 to FCBEF to register a command). Writing to FCCF has no meaning or effect.</p> <p>0 Command in progress 1 All commands complete</p>
5 FPVIOL	<p><b>Protection Violation Flag</b> — FPVIOL is set automatically when FCBEF is cleared to register a command that attempts to erase or program a location in a protected block (the erroneous command is ignored). FPVIOL is cleared by writing a 1 to FPVIOL.</p> <p>0 No protection violation. 1 An attempt was made to erase or program a protected location.</p>
4 FACCERR	<p><b>Access Error Flag</b> — FACCERR is set automatically when the proper command sequence is not obeyed exactly (the erroneous command is ignored), if a program or erase operation is attempted before the FCDIV register has been initialized, or if the MCU enters stop while a command was in progress. For a more detailed discussion of the exact actions that are considered access errors, see <a href="#">Section 4.5.5, “Access Errors.”</a> FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect.</p> <p>0 No access error. 1 An access error has occurred.</p>
2 FBLANK	<p><b>Flash Verified as All Blank (erased) Flag</b> — FBLANK is set automatically at the conclusion of a blank check command if the entire flash array was verified to be erased. FBLANK is cleared by clearing FCBEF to write a new valid command. Writing to FBLANK has no meaning or effect.</p> <p>0 After a blank check command is completed and FCCF = 1, FBLANK = 0 indicates the flash array is not completely erased. 1 After a blank check command is completed and FCCF = 1, FBLANK = 1 indicates the flash array is completely erased (all 0xFF).</p>

### 5.8.5 System Options Register 2 (SOPT2)

This high-page register contains bits to configure MCU-specific features on MC9S08QD4 series devices.



<sup>1</sup> This bit can be written only one time after reset. Additional writes are ignored.

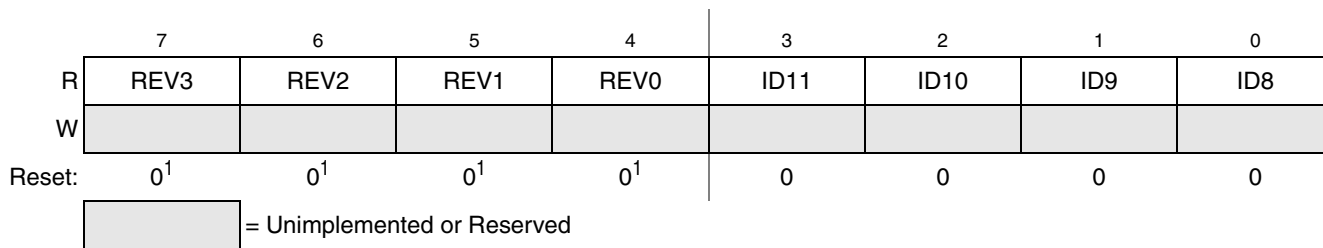
**Figure 5-6. System Options Register 2 (SOPT2)**

**Table 5-7. SOPT2 Register Field Descriptions**

Field	Description
7 COPCLKS	<b>COP Watchdog Clock Select</b> — This write-once bit selects the clock source of the COP watchdog. 0 Internal 32 kHz clock is source to COP. 1 Bus clock is source to COP.

### 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.

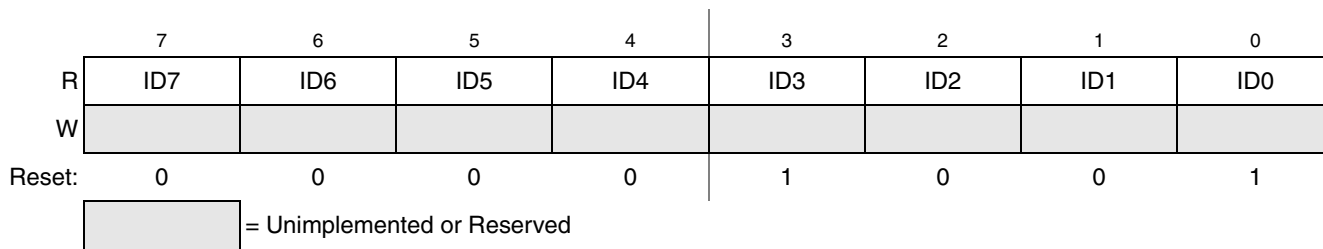


<sup>1</sup> The revision number that is hard coded into these bits reflects the current silicon revision level.

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

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

Field	Description
7:4 REV[3:0]	<b>Revision Number</b> — The high-order 4 bits of address SDIDH are hard coded to reflect the current mask set revision number (0–F).
3:0 ID[11:8]	<b>Part Identification Number</b> — Each derivative in the HCS08 family has a unique identification number. The MC9S08QD4 series is hard coded to the value 0x011. See also ID bits in <a href="#">Table 5-9</a> .



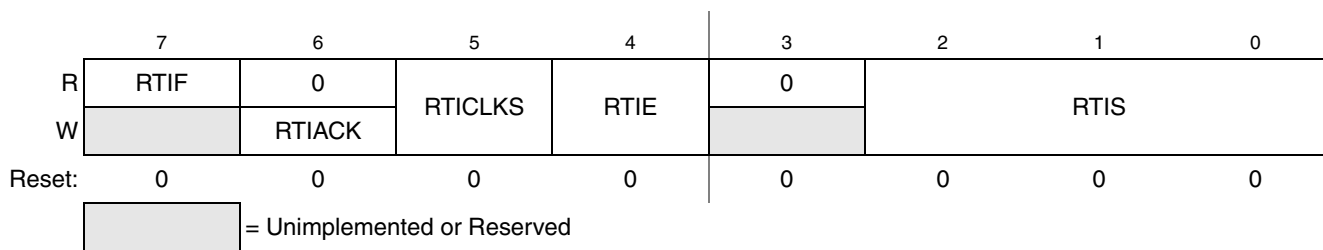
**Figure 5-8. System Device Identification Register — Low (SDIDL)**

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

Field	Description
7:0 ID[7:0]	<b>Part Identification Number</b> — Each derivative in the HCS08 family has a unique identification number. The MC9S08QD4 series is hard coded to the value 0x011. See also ID bits in <a href="#">Table 5-8</a> .

### 5.8.7 System Real-Time Interrupt Status and Control Register (SRTISC)

This high-page register contains status and control bits for the RTI.



**Figure 5-9. System RTI Status and Control Register (SRTISC)**

**Table 5-10. SRTISC Register Field Descriptions**

Field	Description
7 RTIF	<b>Real-Time Interrupt Flag</b> — This read-only status bit indicates the periodic wakeup timer has timed out. 0 Periodic wakeup timer not timed out. 1 Periodic wakeup timer timed out.
6 RTIACK	<b>Real-Time Interrupt Acknowledge</b> — This write-only bit is used to acknowledge real-time interrupt request (write 1 to clear RTIF). Writing 0 has no meaning or effect. Reads always return 0.
5 RTICLKs	<b>Real-Time Interrupt Clock Select</b> — This read/write bit selects the clock source for the real-time interrupt. 0 Real-time interrupt request clock source is internal 1 kHz oscillator. 1 Real-time interrupt request clock source is 32 kHz ICS clock.
4 RTIE	<b>Real-Time Interrupt Enable</b> — This read-write bit enables real-time interrupts. 0 Real-time interrupts disabled. 1 Real-time interrupts enabled.
2:0 RTIS	<b>Real-Time Interrupt Delay Selects</b> — These read/write bits select the period for the RTI. See <a href="#">Table 5-11</a>

## 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.

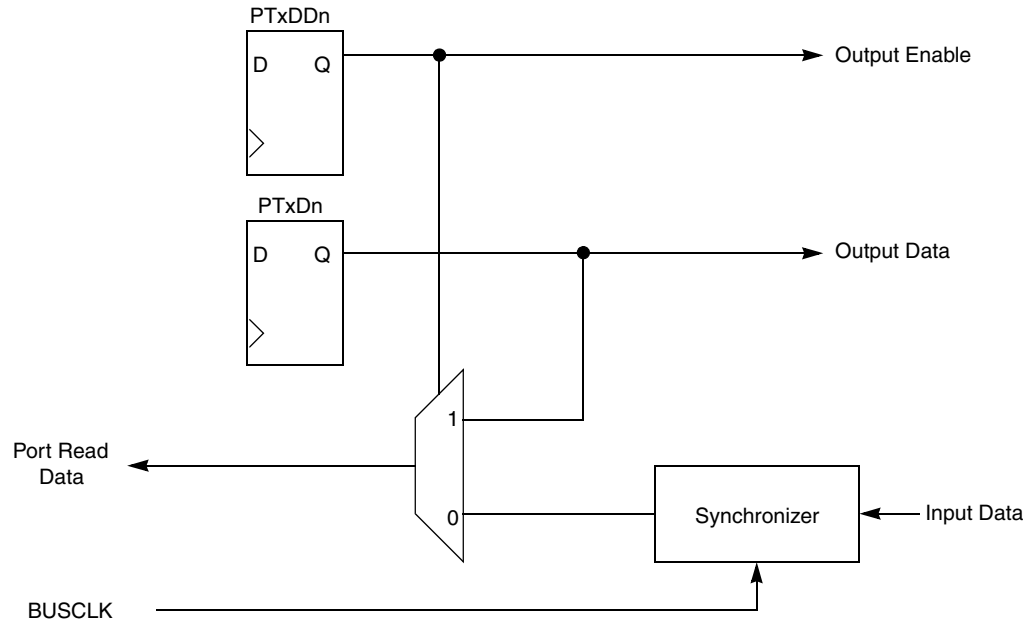
All of these I/O pins are shared with on-chip peripheral functions as shown in [Table 2-1](#). The peripheral modules have priority over the I/Os so that when a peripheral is enabled, the I/O functions associated with the shared pins are disabled. After reset, the shared peripheral functions are disabled 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](#).



**Figure 6-1. Parallel I/O Block Diagram**

The data direction control bit (PTxDDn) determines whether the output buffer for the associated pin is enabled, and also controls the source for port data register reads. The input buffer for the associated pin is always enabled unless the pin is enabled as an analog function or is an output-only pin.

When a shared digital function is enabled for a pin, the output buffer is controlled by the shared function. However, the data direction register bit will continue to control the source for reads of the port data register.

When a shared analog function is enabled for a pin, both the input and output buffers are disabled. A value of 0 is read for any port data bit where the bit is an input (PTxDDn = 0) and the input buffer is disabled. In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both the digital and analog functions are enabled, the analog function controls the pin.

It is a good programming practice to write to the port data register before changing the direction of a port pin to become an output. This ensures that the pin will not be driven momentarily with an old data value that happened to be in the port data register.

## 6.2 Pin Control — Pullup, Slew Rate and Drive Strength

Associated with the parallel I/O ports is a set of registers located in the high-page register space that operate independently of the parallel I/O registers. These registers are used to control pullups, slew rate and drive strength for the pins.

## 6.3 Pin Behavior in Stop Modes

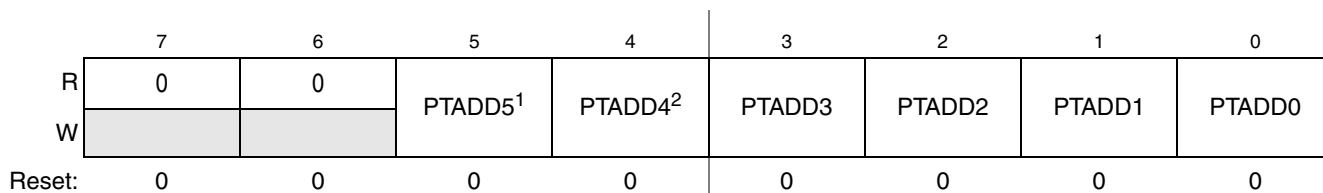
Pin behavior following execution of a STOP instruction depends on the stop mode that is entered. An explanation of pin behavior for the various stop modes follows:



**Table 6-1. PTAD Register Field Descriptions**

Field	Description
5:0 PTAD[5:0]	<b>Port A Data Register Bits</b> — For port A pins that are inputs, reads return the logic level on the pin. For port A pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port A pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTAD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled.

### 6.4.1.2 Port A Data Direction (PTADD)



<sup>1</sup> PTADD5 has no effect on the input-only PTA5 pin. Read this bit is always equal to zero.

<sup>2</sup> PTADD4 has no effect on the output-only PTA4 pin.

**Figure 6-3. Port A Data Direction Register (PTADD)**

**Table 6-2. PTADD Register Field Descriptions**

Field	Description
5:0 PTADD[5:0]	<b>Data Direction for Port A Bits</b> — These read/write bits control the direction of port A pins and what is read for PTAD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port A bit n and PTAD reads return the contents of PTADn.

### 6.4.2 Port A Control Registers

The pins associated with port A are controlled by the registers in this section. These registers control the pin pullup, slew rate and drive strength of the Port A pins independent of the parallel I/O register.

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

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR				
						VH	I	N	Z	C
BCLR <i>n,opr8a</i>	Clear Bit <i>n</i> in Memory ( $M_n \leftarrow 0$ )	DIR (b0)	11 dd	5	rfwpp	--	-	-	-	-
		DIR (b1)	13 dd	5	rfwpp					
		DIR (b2)	15 dd	5	rfwpp					
		DIR (b3)	17 dd	5	rfwpp					
		DIR (b4)	19 dd	5	rfwpp					
		DIR (b5)	1B dd	5	rfwpp					
		DIR (b6)	1D dd	5	rfwpp					
		DIR (b7)	1F dd	5	rfwpp					
BCS <i>rel</i>	Branch if Carry Bit Set (if C = 1) (Same as BLO)	REL	25 rr	3	ppp	--	-	-	-	-
BEQ <i>rel</i>	Branch if Equal (if Z = 1)	REL	27 rr	3	ppp	--	-	-	-	-
BGE <i>rel</i>	Branch if Greater Than or Equal To (if $N \oplus V = 0$ ) (Signed)	REL	90 rr	3	ppp	--	-	-	-	-
BGND	Enter active background if ENBDM=1 Waits for and processes BDM commands until GO, TRACE1, or TAGGO	INH	82	5+	fp...ppp	--	-	-	-	-
BGT <i>rel</i>	Branch if Greater Than (if $Z \mid (N \oplus V) = 0$ ) (Signed)	REL	92 rr	3	ppp	--	-	-	-	-
BHCC <i>rel</i>	Branch if Half Carry Bit Clear (if H = 0)	REL	28 rr	3	ppp	--	-	-	-	-
BHCS <i>rel</i>	Branch if Half Carry Bit Set (if H = 1)	REL	29 rr	3	ppp	--	-	-	-	-
BHI <i>rel</i>	Branch if Higher (if $C \mid Z = 0$ )	REL	22 rr	3	ppp	--	-	-	-	-
BHS <i>rel</i>	Branch if Higher or Same (if C = 0) (Same as BCC)	REL	24 rr	3	ppp	--	-	-	-	-
BIH <i>rel</i>	Branch if IRQ Pin High (if IRQ pin = 1)	REL	2F rr	3	ppp	--	-	-	-	-
BIL <i>rel</i>	Branch if IRQ Pin Low (if IRQ pin = 0)	REL	2E rr	3	ppp	--	-	-	-	-
BIT # <i>opr8i</i> BIT <i>opr8a</i> BIT <i>opr16a</i> BIT <i>opr16,X</i> BIT <i>opr8,X</i> BIT <i>,X</i> BIT <i>opr16,SP</i> BIT <i>opr8,SP</i>	Bit Test (A) & (M) (CCR Updated but Operands Not Changed)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 ii B5 dd C5 hh ll D5 ee ff E5 ff F5 9E D5 ee ff 9E E5 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0-	-	↑	↓	-
BLE <i>rel</i>	Branch if Less Than or Equal To (if $Z \mid (N \oplus V) = 1$ ) (Signed)	REL	93 rr	3	ppp	--	-	-	-	-
BLO <i>rel</i>	Branch if Lower (if C = 1) (Same as BCS)	REL	25 rr	3	ppp	--	-	-	-	-
BLS <i>rel</i>	Branch if Lower or Same (if $C \mid Z = 1$ )	REL	23 rr	3	ppp	--	-	-	-	-
BLT <i>rel</i>	Branch if Less Than (if $N \oplus V = 1$ ) (Signed)	REL	91 rr	3	ppp	--	-	-	-	-
BMC <i>rel</i>	Branch if Interrupt Mask Clear (if I = 0)	REL	2C rr	3	ppp	--	-	-	-	-
BMI <i>rel</i>	Branch if Minus (if N = 1)	REL	2B rr	3	ppp	--	-	-	-	-
BMS <i>rel</i>	Branch if Interrupt Mask Set (if I = 1)	REL	2D rr	3	ppp	--	-	-	-	-
BNE <i>rel</i>	Branch if Not Equal (if Z = 0)	REL	26 rr	3	ppp	--	-	-	-	-
BPL <i>rel</i>	Branch if Plus (if N = 0)	REL	2A rr	3	ppp	--	-	-	-	-

## 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.

are too fast, then 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.

## 8.4.2 Input Select and Pin Control

The pin control registers (APCTL3, APCTL2, and APCTL1) are used to 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.

## 8.4.3 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.

## 8.4.4 Conversion Control

Conversions can be performed in either 10-bit mode or 8-bit mode as determined by the MODE bits. Conversions can be initiated by either a software or hardware trigger. In addition, the ADC module can be configured for low power operation, long sample time, continuous conversion, and automatic compare of the conversion result to a software determined compare value.

### 8.4.4.1 Initiating Conversions

A conversion is initiated:

- Following a write to ADCSC1 (with ADCH bits not all 1s) if software triggered operation is selected.
- Following a hardware trigger (ADHWT) event if hardware triggered operation is selected.
- Following the transfer of the result to the data registers when continuous conversion is enabled.

If continuous conversions are enabled a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation, continuous conversions begin after ADCSC1 is written and continue until aborted. In hardware triggered operation, continuous conversions begin after a hardware trigger event and continue until aborted.

### 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

### 9.3.3 ICS Trim Register (ICSTRM)

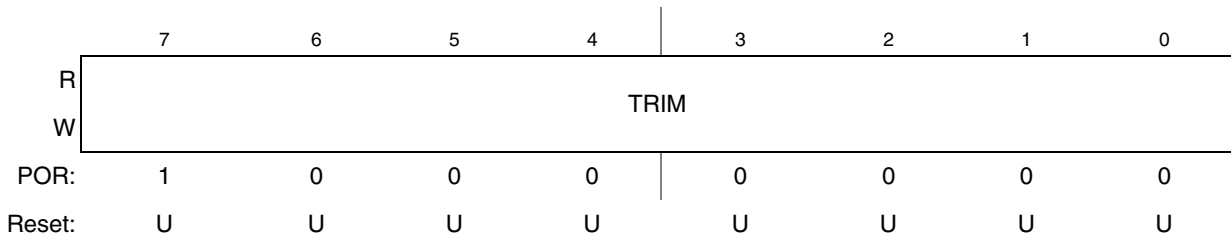


Figure 9-5. ICS Trim Register (ICSTRM)

Table 9-3. ICS Trim Register Field Descriptions

Field	Description
7:0 TRIM	<p><b>ICS Trim Setting</b> — 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.</p> <p>An additional fine trim bit is available in ICSSC as the FTRIM bit.</p>

### 9.3.4 ICS Status and Control (ICSSC)

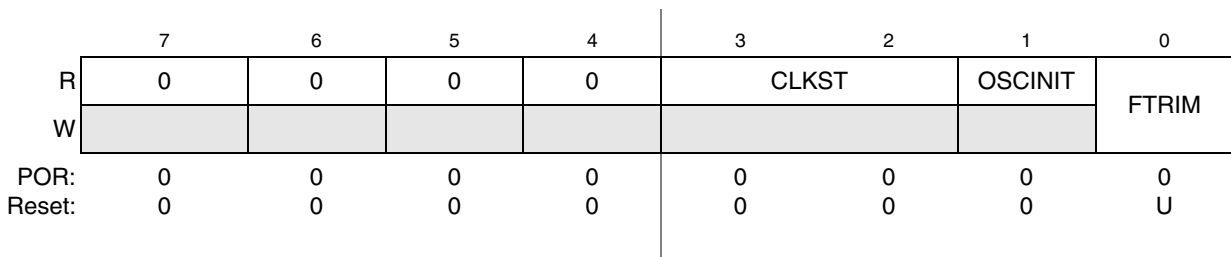


Figure 9-6. ICS Status and Control Register (ICSSC)

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

Field	Description
7:4	Reserved, must be cleared.
3:2 CLKST	<p><b>Clock Mode Status</b> — 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.</p> <p>00 Output of FLL is selected.                      01 FLL Bypassed, Internal reference clock is selected.                      10 FLL Bypassed, External reference clock is selected.                      11 Reserved.</p>
1	<p><b>OSC Initialization</b> — 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.</p>
0	<p><b>ICS Fine Trim</b> — 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.</p>

times the filter frequency, as selected by the RDIV bits, so that the ICSLCLK will be available for BDC communications, and the external reference clock is enabled.

#### 9.4.1.6 FLL Bypassed External Low Power (FBELP)

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

- CLKS bits are written to 10.
- IREFS bit is written to 0.
- BDM mode is not active and LP bit is written to 1.

In FLL bypassed external low power mode, the ICSOUT clock is derived from the external reference clock and the FLL is disabled. The ICSLCLK will not be available for BDC communications. The external reference clock is enabled.

#### 9.4.1.7 Stop

Stop mode is entered whenever the MCU enters a STOP state. In this mode, all ICS clock signals are static except in the following cases:

ICSIRCLK will be active in stop mode when all the following conditions occur:

- IRCLKEN bit is written to 1
- IREFSTEN bit is written to 1

ICSERCLK will be active in stop mode when all the following conditions occur:

- ERCLKEN bit is written to 1
- EREFSTEN bit is written to 1

### 9.4.2 Mode Switching

When switching between FLL engaged internal (FEI) and FLL engaged external (FEE) modes the IREFS bit can be changed at anytime, but the RDIV bits must be changed simultaneously so that the resulting frequency stays in the range of 31.25 kHz to 39.0625 kHz. After a change in the IREFS value the FLL will begin locking again after a few full cycles of the resulting divided reference frequency.

The CLKS bits can also be changed at anytime, but the RDIV bits must be changed simultaneously so that the resulting frequency stays in the range of 31.25 kHz to 39.0625 kHz. The actual switch to the newly selected clock will not occur until after a few full cycles of the new clock. If the newly selected clock is not available, the previous clock will remain selected.

### 9.4.3 Bus Frequency Divider

The BDIV bits can be changed at anytime and the actual switch to the new frequency will occur immediately.

## Chapter 10

# Keyboard Interrupt (S08KBIV2)

### 10.1 Introduction

The keyboard interrupt KBI module provides up to eight independently enabled external interrupt sources. Only four (KBI1P0–KBI1P3) of the possible interrupts are implemented on the MC9S08QD4 series MCU. These inputs are selected by the KBIPE bits.

[Figure 10-1](#) Shows the MC9S08QD4 series with the KBI module and pins highlighted.



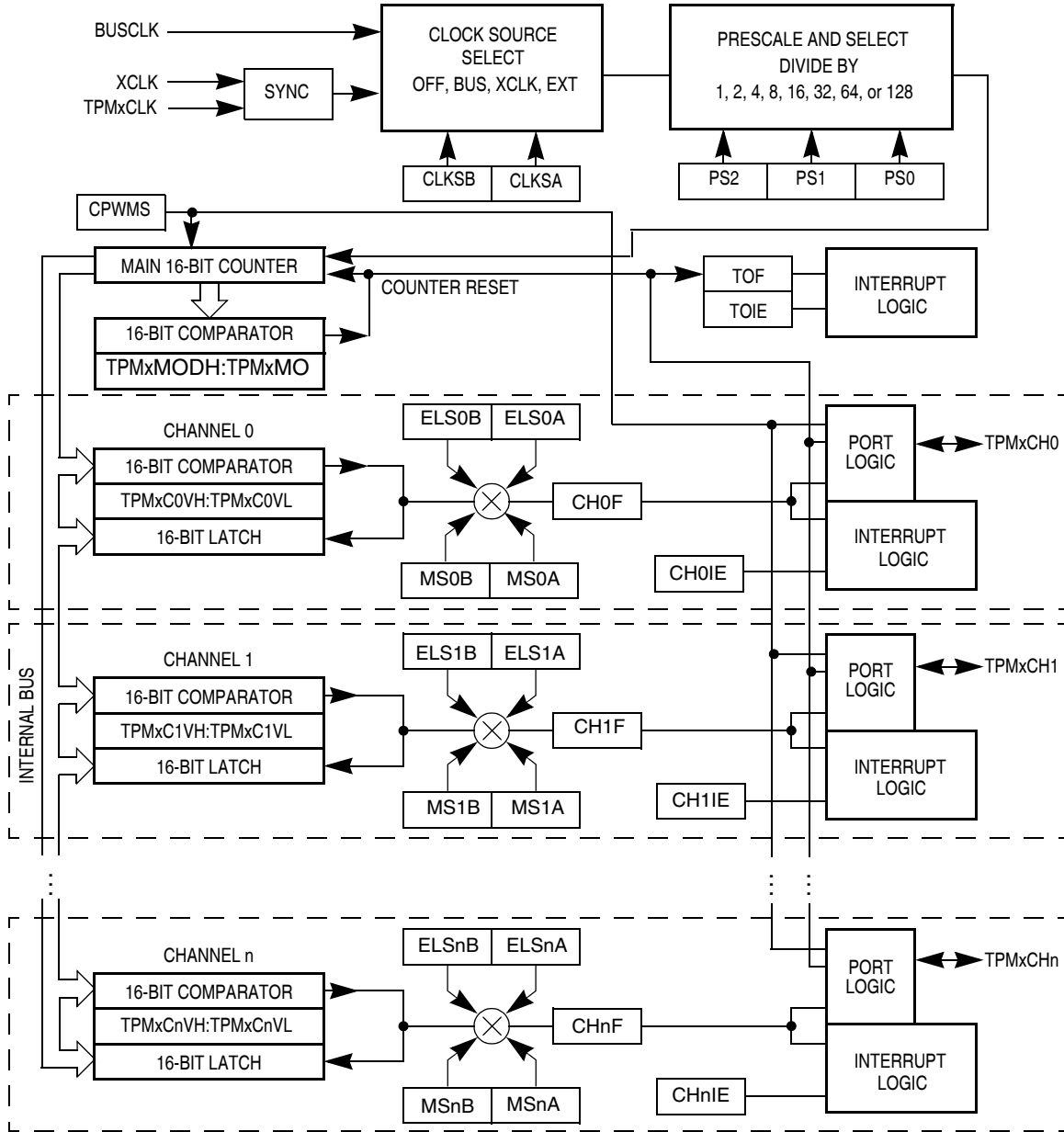


Figure 11-2. TPM Block Diagram

The central component of the TPM is the 16-bit counter that can operate as a free-running counter, a modulo counter, or an up-/down-counter when the TPM is configured for center-aligned PWM. The TPM counter (when operating in normal up-counting mode) provides the timing reference for the input capture, output compare, and edge-aligned PWM functions. The timer counter modulo registers, TPMxMODH:TPMxMODL, control the modulo value of the counter. (The values 0x0000 or 0xFFFF effectively make the counter free running.) Software can read the counter value at any time without affecting the counting sequence. Any write to either byte of the TPMxCNT counter resets the counter regardless of the data value written.

### 11.4.2.2 Output Compare Mode

With the output compare function, the TPM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter reaches the value in the channel value registers of an output compare channel, the TPM can set, clear, or toggle the channel pin.

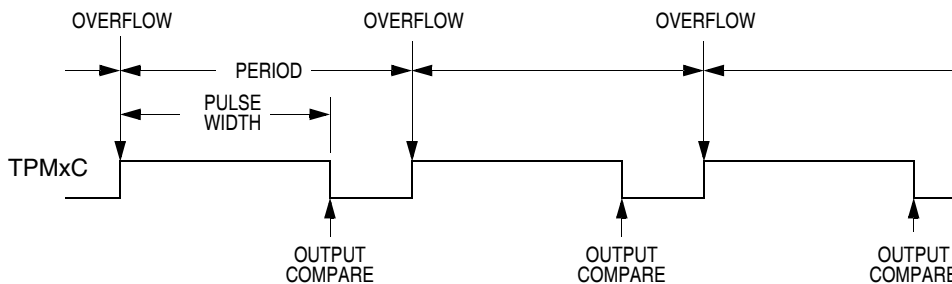
In output compare mode, values are transferred to the corresponding timer channel value registers only after both 8-bit bytes of a 16-bit register have been written. This coherency sequence can be manually reset by writing to the channel status/control register (TPMxCnSC).

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

### 11.4.2.3 Edge-Aligned PWM Mode

This type of PWM output uses the normal up-counting mode of the timer counter (CPWMS = 0) and can be used when other channels in the same TPM are configured for input capture or output compare functions. The period of this PWM signal is determined by the setting in the modulus register (TPMxMODH:TPMxMODL). The duty cycle is determined by the setting in the timer channel value register (TPMxCnVH:TPMxCnVL). The polarity of this PWM signal is determined by the setting in the ELSnA control bit. Duty cycle cases of 0 percent and 100 percent are possible.

As [Figure 11-11](#) shows, the output compare value in the TPM channel registers determines the pulse width (duty cycle) of the PWM signal. The time between the modulus overflow and the output compare is the pulse width. If ELSnA = 0, the counter overflow forces the PWM signal high and the output compare forces the PWM signal low. If ELSnA = 1, the counter overflow forces the PWM signal low and the output compare forces the PWM signal high.



**Figure 11-11. PWM Period and Pulse Width (ELSnA = 0)**

When the channel value register is set to 0x0000, the duty cycle is 0 percent. By setting the timer channel value register (TPMxCnVH:TPMxCnVL) to a value greater than the modulus setting, 100% duty cycle can be achieved. This implies that the modulus setting must be less than 0xFFFF to get 100% duty cycle.

Because the HCS08 is a family of 8-bit MCUs, the settings in the timer channel registers are buffered to ensure coherent 16-bit updates and to avoid unexpected PWM pulse widths. Writes to either register, TPMxCnVH or TPMxCnVL, write to buffer registers. In edge-PWM mode, values are transferred to the corresponding timer channel registers only after both 8-bit bytes of a 16-bit register have been written and the value in the TPMxCNTH:TPMxCNTL counter is 0x0000. (The new duty cycle does not take effect until the next full period.)

## Chapter 12

# Development Support

### 12.1 Introduction

Development support systems in the HCS08 include the background debug controller (BDC). The BDC provides a single-wire debug interface to the target MCU that provides a convenient interface for programming the on-chip flash and other nonvolatile memories. The BDC is also the primary debug interface for development and allows non-intrusive access to memory data and traditional debug features such as CPU register modify, breakpoints, and single instruction trace commands.

In the HCS08 Family, address and data bus signals are not available on external pins (not even in test modes). Debug is done through commands fed into the target MCU via the single-wire background debug interface. The debug module provides a means to selectively trigger and capture bus information so an external development system can reconstruct what happened inside the MCU on a cycle-by-cycle basis without having external access to the address and data signals.

#### 12.1.1 Forcing Active Background

The method for forcing active background mode depends on the specific HCS08 derivative. For the MC9S08QD4 series, you can force active background mode by holding the BKGD pin low as the MCU exits the reset condition independent of what caused the reset. If no debug pod is connected to the BKGD pin, the MCU will always reset into normal operating mode.

#### 12.1.2 Module Configuration

The alternative BDC clock source for MC9S08QD4 series is the ICGCLK. See [Chapter 9, “Internal Clock Source \(S08ICSV1\)”](#) for more information about ICGCLK and how to select clock sources.

Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first)

- / = separates parts of the command
- d = delay 16 target BDC clock cycles
- AAAA = a 16-bit address in the host-to-target direction
- RD = 8 bits of read data in the target-to-host direction
- WD = 8 bits of write data in the host-to-target direction
- RD16 = 16 bits of read data in the target-to-host direction
- WD16 = 16 bits of write data in the host-to-target direction
- SS = the contents of BDCSCR in the target-to-host direction (STATUS)
- CC = 8 bits of write data for BDCSCR in the host-to-target direction (CONTROL)
- RBKP = 16 bits of read data in the target-to-host direction (from BDCBKPT breakpoint register)
- WBKP = 16 bits of write data in the host-to-target direction (for BDCBKPT breakpoint register)

## A.7 Internal Clock Source Characteristics

**Table A-7. Internal Clock Source Specifications**

Characteristic	Symbol	Min	Typ <sup>1</sup>	Max	Unit
Average internal reference frequency - untrimmed	$f_{int\_ut}$	25	31.25	41.66	kHz
Average internal reference frequency - trimmed	$f_{int\_t}$	—	31.25	—	kHz
DCO output frequency range - untrimmed	$f_{dco\_ut}$	12.8	16	21.33	MHz
DCO output frequency range - trimmed	$f_{dco\_t}$	—	16	—	MHz
Resolution of trimmed DCO output frequency at fixed voltage and temperature (Consumer and Industrial MC9S08QDx) <sup>2</sup>	$\Delta f_{dco\_res\_t}$	—	—	± 0.2	% $f_{dco}$
Resolution of trimmed DCO output frequency at fixed voltage and temperature (Automotive S9S08QDx) <sup>2</sup> -40°C to 0°C 0 to 125°C		—	—	± 0.3 ± 0.2	
Total deviation of trimmed DCO output frequency over voltage and temperature <sup>2</sup> Consumer and Industrial MC9S08QDx Automotive S9S08QDx	$\Delta f_{dco\_t}$	—	—	± 2 ± 3	% $f_{dco}$
FLL acquisition time <sup>2,3</sup>	$t_{acquire}$			1	ms
Long term Jitter of DCO output clock (averaged over 2 ms interval) <sup>4</sup>	$C_{Jitter}$	—	—	0.6	% $f_{dco}$

<sup>1</sup> Data in Typical column was characterized at 3.0 V and 3.0 V, 25°C or is typical recommended value.

<sup>2</sup> Characterized, but not tested.

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

<sup>4</sup> Jitter is the average deviation from the programmed frequency measured over the specified interval at maximum  $f_{BUS}$ . Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the FLL circuitry via  $V_{DD}$  and  $V_{SS}$  and variation in crystal oscillator frequency increase the  $C_{Jitter}$  percentage for a given interval.