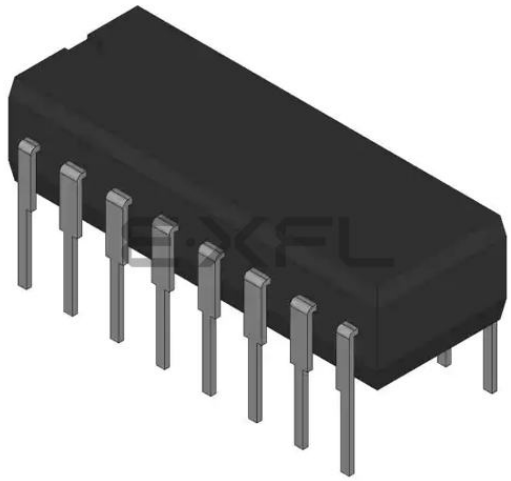


Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	12
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	16-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08qg8mpbe

MC9S08QG8/4 Features

8-Bit HCS08 Central Processor Unit (CPU)

- 20-MHz HCS08 CPU (central processor unit)
- HC08 instruction set with added BGND instruction
- Background debugging system
- Breakpoint capability to allow single breakpoint setting during in-circuit debugging (plus two more breakpoints in on-chip debug module)
- Debug module containing two comparators and nine trigger modes. Eight deep FIFO for storing change-of-flow addresses and event-only data Debug module supports both tag and force breakpoints
- Support for up to 32 interrupt/reset sources

Memory Options

- FLASH read/program/erase over full operating voltage and temperature
- MC9S08QG8 — 8 Kbytes FLASH, 512 bytes RAM MC9S08QG4 — 4 Kbytes FLASH, 256 bytes RAM

Power-Saving Modes

- Wait plus three stops

Clock Source Options

- **ICS** — Internal clock source module containing a frequency-locked-loop (FLL) controlled by internal or external reference; precision trimming of internal reference allows 0.2% resolution and 2% deviation over temperature and voltage; supports bus frequencies from 1 MHz to 10 MHz
- **XOSC** — Low-power oscillator module with software selectable crystal or ceramic resonator range, 31.25 kHz to 38.4 kHz or 1 MHz to 16 MHz, and supports external clock source input up to 20 MHz

System Protection

- Watchdog computer operating properly (COP) reset with option to run from dedicated 1-kHz internal clock source or bus clock
- Low-voltage detection with reset or interrupt
- Illegal opcode detection with reset
- Illegal address detection with reset
- FLASH block protect

Peripherals

- **ADC** — 8-channel, 10-bit analog-to-digital converter with automatic compare function, asynchronous clock source, temperature sensor, and internal bandgap reference channel; ADC is hardware triggerable using the RTI counter
- **ACMP** — Analog comparator module with option to compare to internal reference; output can be optionally routed to TPM module
- **SCI** — Serial communications interface module with option for 13-bit break capabilities
- **SPI** — Serial peripheral interface module
- **IIC** — Inter-integrated circuit bus module
- **TPM**— 2-channel timer/pulse-width modulator; each channel can be used for input capture, output compare, buffered edge-aligned PWM, or buffered center-aligned PWM
- **MTIM** — 8-bit modulo timer module with 8-bit prescaler
- **KBI** — 8-pin keyboard interrupt module with software selectable polarity on edge or edge/level modes

Input/Output

- 12 general-purpose input/output (I/O) pins, one input-only pin and one output-only pin; outputs 10 mA each, 60 mA max for package
- Software selectable pullups on ports when used as input
- Software selectable slew rate control and drive strength on ports when used as output
- Internal pullup on $\overline{\text{RESET}}$ and $\overline{\text{IRQ}}$ pins to reduce customer system cost

Development Support

- Single-wire background debug interface
- On-chip, in-circuit emulation (ICE) with real-time bus capture

Package Options

- 24-pin quad flat no lead (QFN) package
- 16-pin plastic dual in-line package (PDIP) — MC9S08QG8 only
- 16-pin quad flat no lead (QFN) package
- 16-pin thin shrink small outline package (TSSOP)
- 8-pin dual flat no lead (DFN) package
- 8-pin PDIP — *MC9S08QG4 only*
- 8-pin narrow body small outline integrated circuit (SOIC) package

4.5.1 Features

Features of the FLASH memory include:

- FLASH size
  - MC9S08QG8: 8,192 bytes (16 pages of 512 bytes each)
  - MC9S08QG4: 4,096 bytes (8 pages of 512 bytes each)
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature
- Flexible block protection
- Security feature for FLASH and RAM
- Auto power-down for low-frequency read accesses

4.5.2 Program and Erase Times

Before any program or erase command can be accepted, the FLASH clock divider register (FCDIV) must be written to set the internal clock for the FLASH module to a frequency (f_{FCLK}) between 150 kHz and 200 kHz (see Section 4.7.1, "FLASH Clock Divider Register (FCDIV)"). This register can be written only once, so normally this write is done during reset initialization. FCDIV cannot be written if the access error flag, FACCERR in FSTAT, is set. The user must ensure that FACCERR is not set before writing to the FCDIV register. One period of the resulting clock ($1/f_{FCLK}$) is used by the command processor to time program and erase pulses. An integer number of these timing pulses are used by the command processor to complete a program or erase command.

Table 4-5 shows program and erase times. The bus clock frequency and FCDIV determine the frequency of FCLK (f_{FCLK}). The time for one cycle of FCLK is $t_{FCLK} = 1/f_{FCLK}$. The times are shown as a number of cycles of FCLK and as an absolute time for the case where $t_{FCLK} = 5$ µs. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

Table 4-5. Program and Erase Times

Parameter	Cycles of FCLK	Time if FCLK = 200 kHz
Byte program	9	45 µs
Byte program (burst)	4	20 µs[1]
Page erase	4000	20 ms
Mass erase	20,000	100 ms

1 Excluding start/end overhead

NOTE

If the COP is enabled during an erase function, make sure the COP is serviced during the erase command execution.

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.

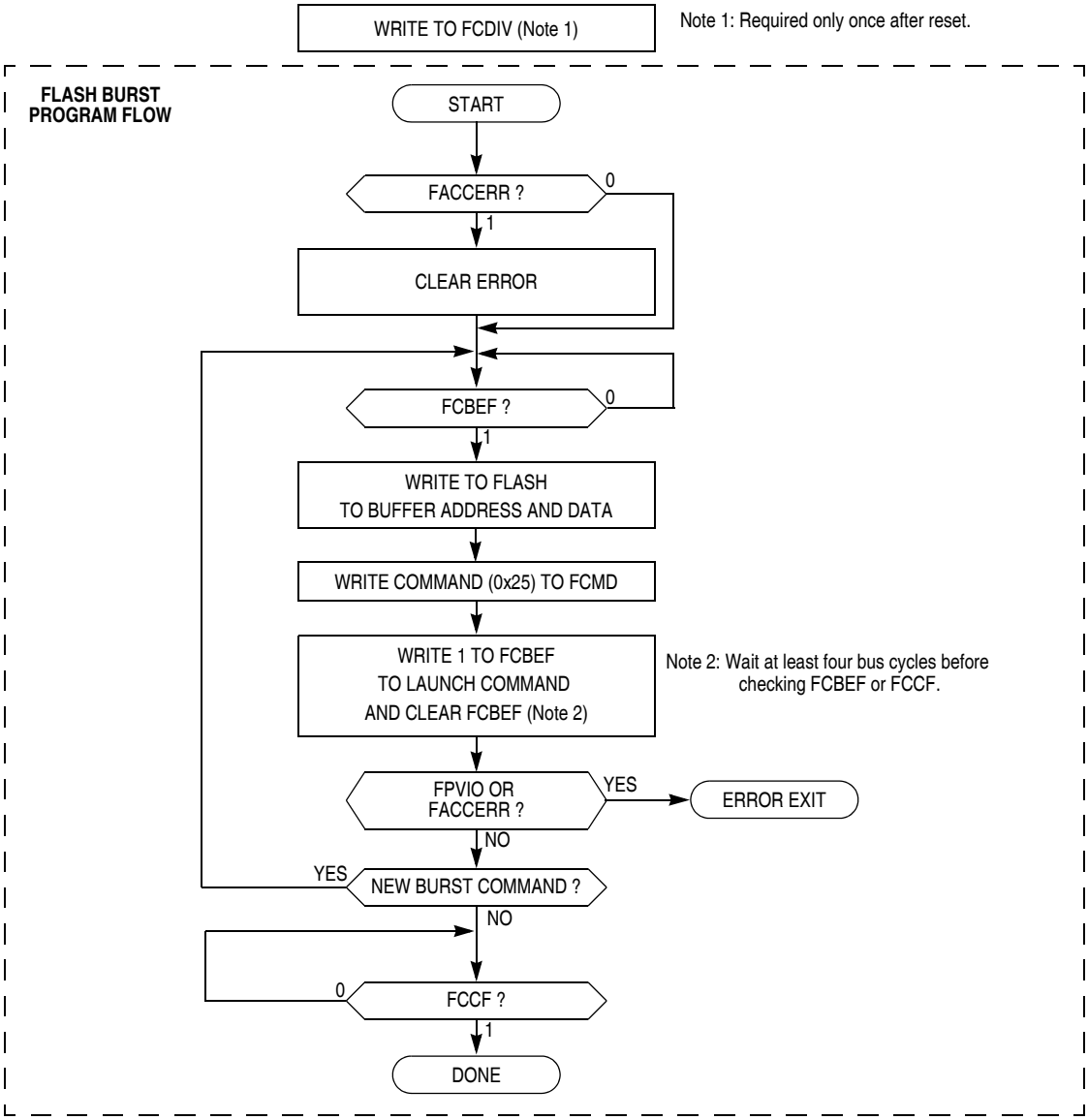


Figure 4-3. FLASH Burst Program Flowchart

Table 4-12. FSTAT Register Field Descriptions (continued)

Field	Description
4 FACCERR	**Access Error Flag** — 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 Section 4.5.5, "Access Errors." FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 No access error. 1 An access error has occurred.
2 FBLANK	**FLASH Verified as All Blank (erased) Flag** — 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. 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).

4.7.6 FLASH Command Register (FCMD)

Only five command codes are recognized in normal user modes as shown in Table 4-13. Refer to Section 4.5.3, "Program and Erase Command Execution," for a detailed discussion of FLASH programming and erase operations.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FCMD							
Reset	0	0	0	0	0	0	0	0

Figure 4-10. FLASH Command Register (FCMD)

Table 4-13. FLASH Commands

Command	FCMD	Equate File Label
Blank check	0x05	mBlank
Byte program	0x20	mByteProg
Byte program — burst mode	0x25	mBurstProg
Page erase (512 bytes/page)	0x40	mPageErase
Mass erase (all FLASH)	0x41	mMassErase

All other command codes are illegal and generate an access error.

It is not necessary to perform a blank check command after a mass erase operation. The blank check command is only required as part of the security unlocking mechanism.

Table 5-2. Vector Summary

Vector Priority	Vector Number	Address (High:Low)	Vector Name	Module	Source	Enable	Description
Lower	31 through 24	0xFFC0:FFC1 through 0xFFCE:FFCF	Unused Vector Space (available for user program)				
	23	0xFFD0:FFD1	Vrti	System control	RTIF	RTIE	Real-time interrupt
	22	0xFFD2:FFD3	—	—	—	—	—
	21	0xFFD4:FFD5	—	—	—	—	—
	20	0xFFD6:FFD7	Vacmp	ACMP	ACF	ACIE	ACMP
	19	0xFFD8:FFD9	Vadc	ADC	COCO	AIEN	ADC
	18	0xFFDA:FFDB	Vkeyboard	KBI	KBF	KBIE	Keyboard pins
	17	0xFFDC:FFDD	Viic	IIC	IICIF	IICIE	IIC control
	16	0xFFDE:FFDF	Vscitx	SCI	TDRE TC	TIE TCIE	SCI transmit
	15	0xFFE0:FFE1	Vscirx	SCI	IDLE RDRF	ILIE RIE	SCI receive
	14	0xFFE2:FFE3	Vscierr	SCI	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI error
	13	0xFFE4:FFE5	Vspi	SPI	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI
	12	0xFFE6:FFE7	Vmtim	MTIM	TOF	TOIE	MTIM
	11	0xFFE8:FFE9	—	—	—	—	—
	10	0xFFEA:FFEB	—	—	—	—	—
	9	0xFFEC:FFED	—	—	—	—	—
	8	0xFFEE:FFEF	—	—	—	—	—
	7	0xFFF0:FFF1	Vtpmovf	TPM	TOF	TOIE	TPM overflow
	6	0xFFF2:FFF3	Vtpmch1	TPM	CH1F	CH1IE	TPM channel 1
	5	0xFFF4:FFF5	Vtpmch0	TPM	CH0F	CH0IE	TPM channel 0
	4	0xFFF6:FFF7	—	—	—	—	—
	3	0xFFF8:FFF9	Vlvd	System control	LVDF	LVDIE	Low-voltage detect
	2	0xFFFA:FFFB	Virq	IRQ	IRQF	IRQIE	$\overline{\text{IRQ}}$ pin
	1	0xFFFC:FFFD	Vswi	CPU	SWI Instruction	—	Software interrupt
Higher	0	0xFFFE:FFFF	Vreset	System control	COP LVD $\overline{\text{RESET}}$ pin Illegal opcode Illegal address POR	COPE LVDRE RSTPE — — —	Watchdog timer Low-voltage detect External pin Illegal opcode Illegal address power-on-reset

interrupt service routine, this would allow nesting of interrupts (which is not recommended because it leads to programs that are difficult to debug and maintain).

For compatibility with the earlier M68HC05 MCUs, the high-order half of the H:X index register pair (H) is not saved on the stack as part of the interrupt sequence. The user must use a PSHH instruction at the beginning of the service routine to save H and then use a PULH instruction just before the RTI that ends the interrupt service routine. It is not necessary to save H if you are certain that the interrupt service routine does not use any instructions or auto-increment addressing modes that might change the value of H.

The software interrupt (SWI) instruction is like a hardware interrupt except that it is not masked by the global I bit in the CCR and it is associated with an instruction opcode within the program so it is not asynchronous to program execution.

7.4.3 Wait Mode Operation

The WAIT instruction enables interrupts by clearing the I bit in the CCR. It then halts the clocks to the CPU to reduce overall power consumption while the CPU is waiting for the interrupt or reset event that will wake the CPU from wait mode. When an interrupt or reset event occurs, the CPU clocks will resume and the interrupt or reset event will be processed normally.

If a serial BACKGROUND command is issued to the MCU through the background debug interface while the CPU is in wait mode, CPU clocks will resume and the CPU will enter active background mode where other serial background commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in wait mode.

7.4.4 Stop Mode Operation

Usually, all system clocks, including the crystal oscillator (when used), are halted during stop mode to minimize power consumption. In such systems, external circuitry is needed to control the time spent in stop mode and to issue a signal to wake up the target MCU when it is time to resume processing. Unlike the earlier M68HC05 and M68HC08 MCUs, the HCS08 can be configured to keep a minimum set of clocks running in stop mode. This optionally allows an internal periodic signal to wake the target MCU from stop mode.

When a host debug system is connected to the background debug pin (BKGD) and the ENBDM control bit has been set by a serial command through the background interface (or because the MCU was reset into active background mode), the oscillator is forced to remain active when the MCU enters stop mode. In this case, if a serial BACKGROUND command is issued to the MCU through the background debug interface while the CPU is in stop mode, CPU clocks will resume and the CPU will enter active background mode where other serial background commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in stop mode.

Recovery from stop mode depends on the particular HCS08 and whether the oscillator was stopped in stop mode. Refer to the Modes of Operation chapter for more details.

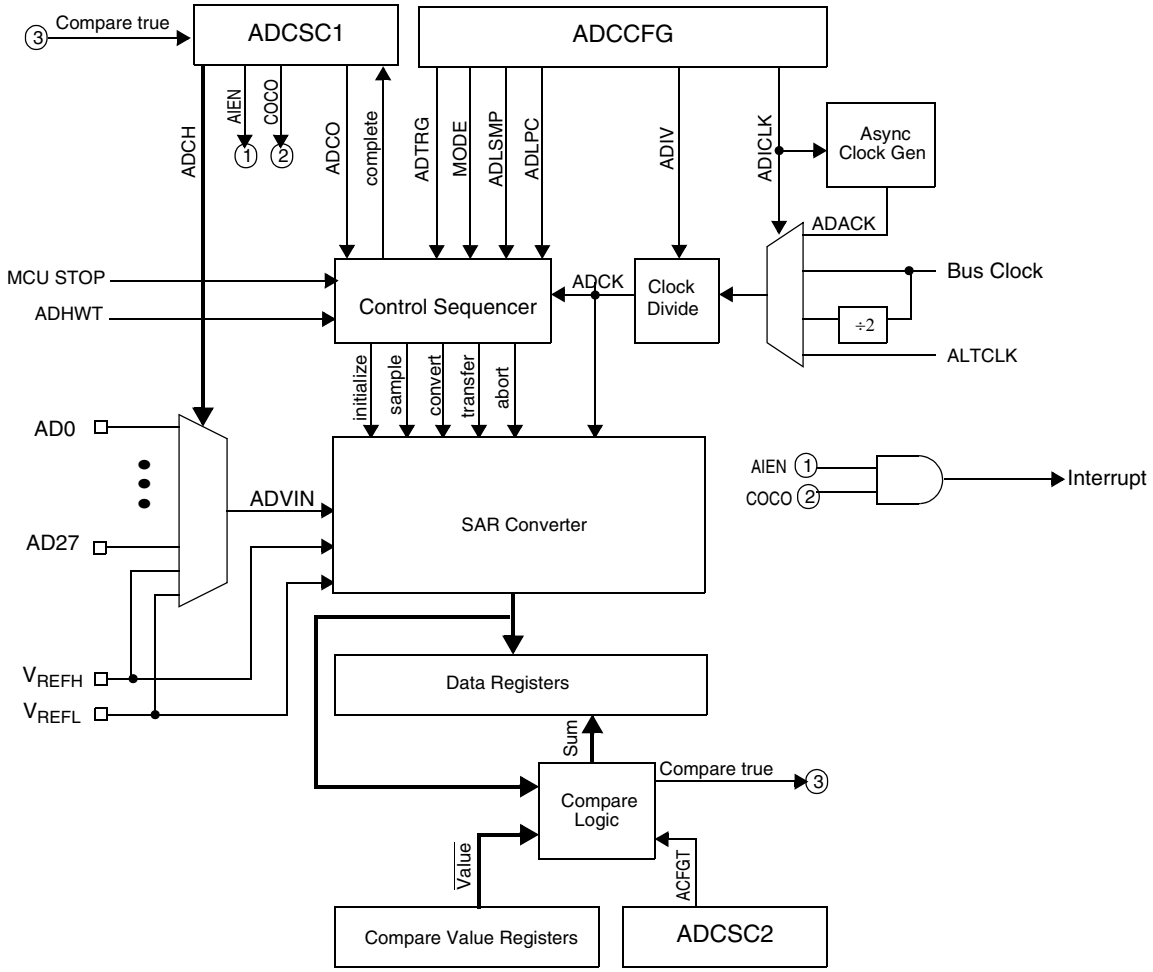


Figure 9-2. ADC Block Diagram

9.2 External Signal Description

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

Table 9-2. Signal Properties

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

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 9-12.

Table 9-12. Total Conversion Time vs. Control Conditions

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 µs + 20 ADCK + 5 bus clock cycles
Single or first continuous 10-bit	11	0	5 µs + 23 ADCK + 5 bus clock cycles
Single or first continuous 8-bit	11	1	5 µs + 40 ADCK + 5 bus clock cycles
Single or first continuous 10-bit	11	1	5 µs + 43 ADCK + 5 bus clock cycles
Subsequent continuous 8-bit; $f_{BUS} \geq f_{ADCK}$	xx	0	17 ADCK cycles
Subsequent continuous 10-bit; $f_{BUS} \geq f_{ADCK}$	xx	0	20 ADCK cycles
Subsequent continuous 8-bit; $f_{BUS} \geq f_{ADCK}/11$	xx	1	37 ADCK cycles
Subsequent continuous 10-bit; $f_{BUS} \geq f_{ADCK}/11$	xx	1	40 ADCK cycles

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:

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

$$\text{Number of bus cycles} = 3.5 \text{ µs} \times 8 \text{ MHz} = 28 \text{ cycles}$$

NOTE

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

10.1.4.5 FLL Bypassed External (FBE)

In FLL bypassed external mode, the FLL is enabled and controlled by an external reference clock, but is bypassed. The ICS supplies a clock derived from the external reference clock. The external reference clock can be an external crystal/resonator supplied by an OSC controlled by the ICS, or it can be another external clock source. The BDC clock is supplied from the FLL.

10.1.4.6 FLL Bypassed External Low Power (FBELP)

In FLL bypassed external low power mode, the FLL is disabled and bypassed, and the ICS supplies a clock derived from the external reference clock. The external reference clock can be an external crystal/resonator supplied by an OSC controlled by the ICS, or it can be another external clock source. The BDC clock is not available.

10.1.4.7 Stop (STOP)

In stop mode, the FLL is disabled and the internal or external reference clock can be selected to be enabled or disabled. The BDC clock is not available. ICS does not provide an MCU clock source.

10.1.5 Block Diagram

This section contains the ICS block diagram.

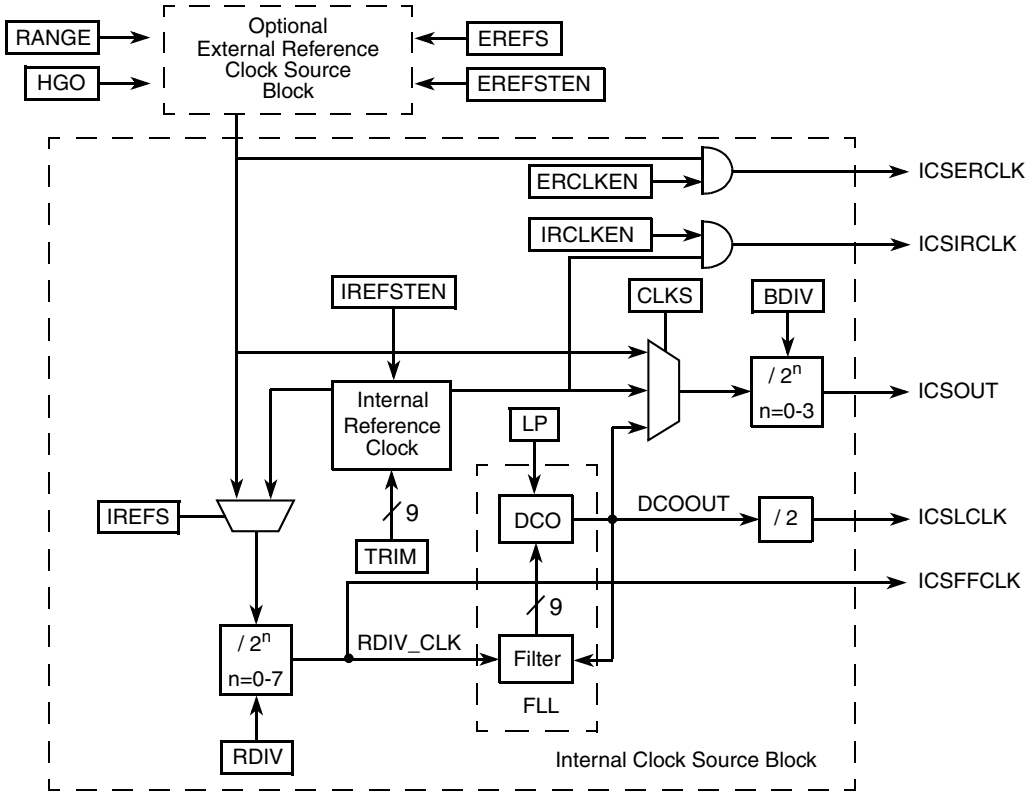


Figure 10-2. Internal Clock Source (ICS) Block Diagram

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to T8 again. When data is transferred from the transmit data buffer to the transmit shifter, the value in T8 is copied at the same time data is transferred from SCID to the shifter.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wakeup so the ninth data bit can serve as the wakeup bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

14.4.2 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.

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.

14.4.3 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 RxD1 pin is not used by the SCI, so it reverts to a general-purpose port I/O pin.

14.4.4 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 TxD1 pin. The RxD1 pin is not used and reverts to a general-purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD1 pin. When TXDIR = 0, the TxD1 pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD1 pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD1 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.

transferred to the corresponding timer channel registers only after both 8-bit bytes of a 16-bit register have been written and the timer counter overflows (reverses direction from up-counting to down-counting at the end of the terminal count in the modulus register). This TPMCNT overflow requirement only applies to PWM channels, not output compares.

Optionally, when TPMCNTH:TPMCNTL = TPMMODH:TPMMODL, the TPM can generate a TOF interrupt at the end of this count. The user can choose to reload any number of the PWM buffers, and they will all update simultaneously at the start of a new period.

Writing to TPMSC cancels any values written to TPMMODH and/or TPMMODL and resets the coherency mechanism for the modulo registers. Writing to TPMCnSC cancels any values written to the channel value registers and resets the coherency mechanism for TPMCnVH:TPMCnVL.

16.5 TPM Interrupts

The TPM generates an optional interrupt for the main counter overflow and an interrupt for each channel. The meaning of channel interrupts depends on the mode of operation for each channel. If the channel is configured for input capture, the interrupt flag is set each time the selected input capture edge is recognized. If the channel is configured for output compare or PWM modes, the interrupt flag is set each time the main timer counter matches the value in the 16-bit channel value register. See the Resets, Interrupts, and System Configuration chapter for absolute interrupt vector addresses, priority, and local interrupt mask control bits.

For each interrupt source in the TPM, a flag bit is set on recognition of the interrupt condition such as timer overflow, channel input capture, or output compare events. This flag may be read (polled) by software to verify that the action has occurred, or an associated enable bit (TOIE or CHnIE) can be set to enable hardware interrupt generation. While the interrupt enable bit is set, a static interrupt will be generated whenever the associated interrupt flag equals 1. It is the responsibility of user software to perform a sequence of steps to clear the interrupt flag before returning from the interrupt service routine.

16.5.1 Clearing Timer Interrupt Flags

TPM interrupt flags are cleared by a 2-step process that includes a read of the flag bit while it is set (1) followed by a write of 0 to the bit. If a new event is detected between these two steps, the sequence is reset and the interrupt flag remains set after the second step to avoid the possibility of missing the new event.

16.5.2 Timer Overflow Interrupt Description

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the 16-bit timer 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 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.)

A force-type breakpoint waits for the current instruction to finish and then acts upon the breakpoint request. The usual action in response to a breakpoint is to go to active background mode rather than continuing to the next instruction in the user application program.

The tag vs. force terminology is used in two contexts within the debug module. The first context refers to breakpoint requests from the debug module to the CPU. The second refers to match signals from the comparators to the debugger control logic. When a tag-type break request is sent to the CPU, a signal is entered into the instruction queue along with the opcode so that if/when this opcode ever executes, the CPU will effectively replace the tagged opcode with a BGND opcode so the CPU goes to active background mode rather than executing the tagged instruction. When the TRGSEL control bit in the DBGT register is set to select tag-type operation, the output from comparator A or B is qualified by a block of logic in the debug module that tracks opcodes and only produces a trigger to the debugger if the opcode at the compare address is actually executed. There is separate opcode tracking logic for each comparator so more than one compare event can be tracked through the instruction queue at a time.

17.3.5 Trigger Modes

The trigger mode controls the overall behavior of a debug run. The 4-bit TRG field in the DBGT register selects one of nine trigger modes. When TRGSEL = 1 in the DBGT register, the output of the comparator must propagate through an opcode tracking circuit before triggering FIFO actions. The BEGIN bit in DBGT chooses whether the FIFO begins storing data when the qualified trigger is detected (begin trace), or the FIFO stores data in a circular fashion from the time it is armed until the qualified trigger is detected (end trigger).

A debug run is started by writing a 1 to the ARM bit in the DBGC register, which sets the ARMF flag and clears the AF and BF flags and the CNT bits in DBGS. A begin-trace debug run ends when the FIFO gets full. An end-trace run ends when the selected trigger event occurs. Any debug run can be stopped manually by writing a 0 to ARM or DBGEN in DBGC.

In all trigger modes except event-only modes, the FIFO stores change-of-flow addresses. In event-only trigger modes, the FIFO stores data in the low-order eight bits of the FIFO.

The BEGIN control bit is ignored in event-only trigger modes and all such debug runs are begin type traces. When TRGSEL = 1 to select opcode fetch triggers, it is not necessary to use R/W in comparisons because opcode tags would only apply to opcode fetches that are always read cycles. It would also be unusual to specify TRGSEL = 1 while using a full mode trigger because the opcode value is normally known at a particular address.

The following trigger mode descriptions only state the primary comparator conditions that lead to a trigger. Either comparator can usually be further qualified with R/W by setting RWAEN (RWBEN) and the corresponding RWA (RWB) value to be matched against R/W. The signal from the comparator with optional R/W qualification is used to request a CPU breakpoint if BRKEN = 1 and TAG determines whether the CPU request will be a tag request or a force request.

A-Only — Trigger when the address matches the value in comparator A

A OR B — Trigger when the address matches either the value in comparator A or the value in comparator B

17.3.6 Hardware Breakpoints

The BRKEN control bit in the DBGC register may be set to 1 to allow any of the trigger conditions described in Section 17.3.5, “Trigger Modes,” to be used to generate a hardware breakpoint request to the CPU. TAG in DBGC controls whether the breakpoint request will be treated as a tag-type breakpoint or a force-type breakpoint. A tag breakpoint causes the current opcode to be marked as it enters the instruction queue. If a tagged opcode reaches the end of the pipe, the CPU executes a BGND instruction to go to active background mode rather than executing the tagged opcode. A force-type breakpoint causes the CPU to finish the current instruction and then go to active background mode.

If the background mode has not been enabled (ENBDM = 1) by a serial WRITE_CONTROL command through the BKGD pin, the CPU will execute an SWI instruction instead of going to active background mode.

17.4 Register Definition

This section contains the descriptions of the BDC and DBG registers and control bits.

Refer to the high-page register summary in the device overview chapter of this data sheet for the absolute address assignments for all DBG registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

17.4.1 BDC Registers and Control Bits

The BDC has two registers:

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

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

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

Table A-7. Supply Current Characteristics

Parameter	Symbol	V_{DD} (V)[1]	Typical[2]	Max	T (°C)
RTI adder to stop1, stop2 or stop3 [4]	—	3	300 nA	—	85
		2	300 nA	—	85
LVD adder to stop3 (LVDE = LVDSE = 1)	—	3	70 µA	—	85
		2	60 µA	—	85
Adder to stop3 for oscillator enabled [5] (EREFSTEN =1)	—	3	5 µA	—	85
		2	4 µA	—	85

1 3-V values are 100% tested; 2-V values are characterized but not tested.

2 Typicals are measured at 25°C.

3 Does not include any DC loads on port pins.

4 Most customers are expected to find that auto-wakeup from a stop mode can be used instead of the higher current wait mode.

5 Values given under the following conditions: low range operation (RANGE = 0), Loss-of-clock disabled (LOCD = 1), low-power oscillator (HGO = 0).

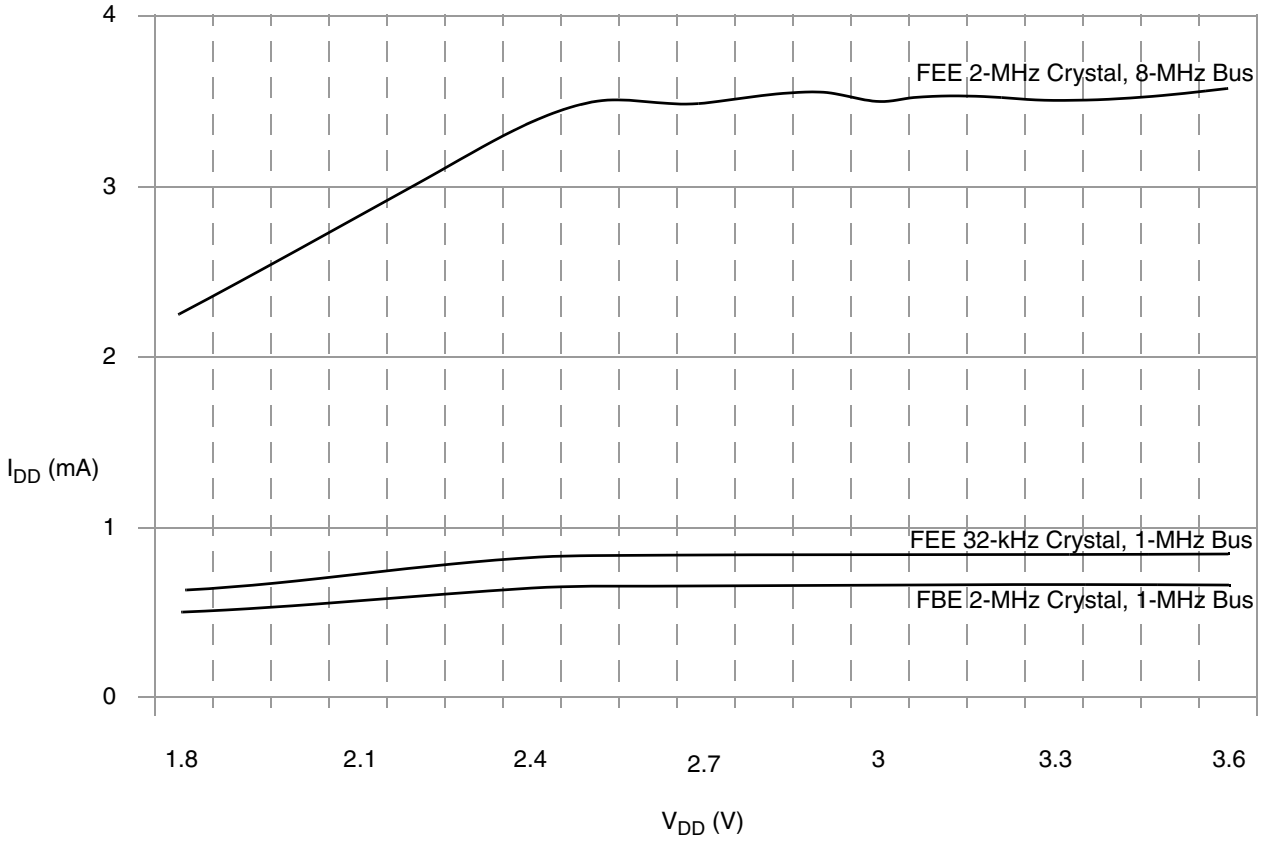


Figure A-6. Typical Run I_{DD} for FBE and FEE, I_{DD} vs. V_{DD} (ACMP and ADC off, All Other Modules Enabled)

A.7 External Oscillator (XOSC) and Internal Clock Source (ICS) Characteristics

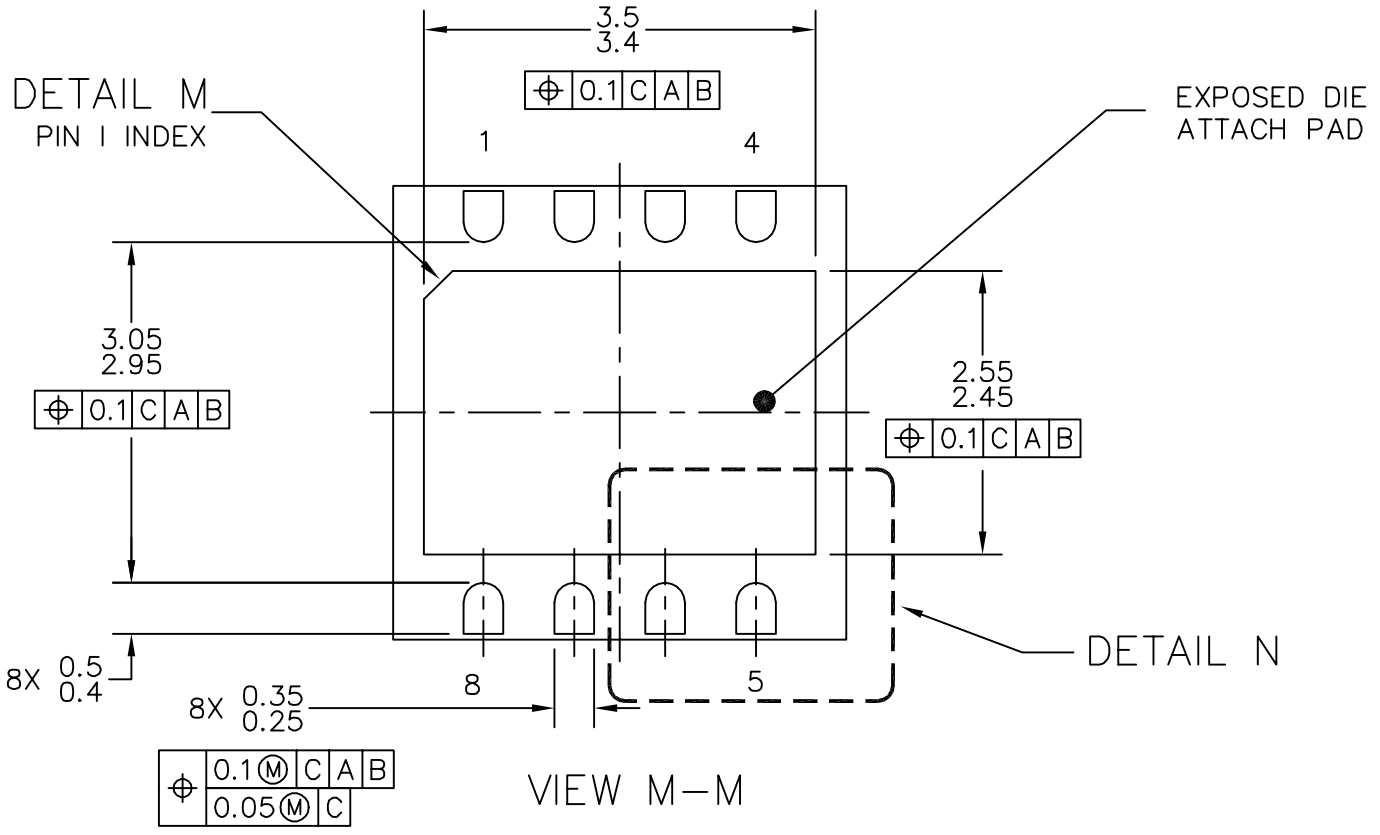
Reference Figure A-7 for crystal or resonator circuit.

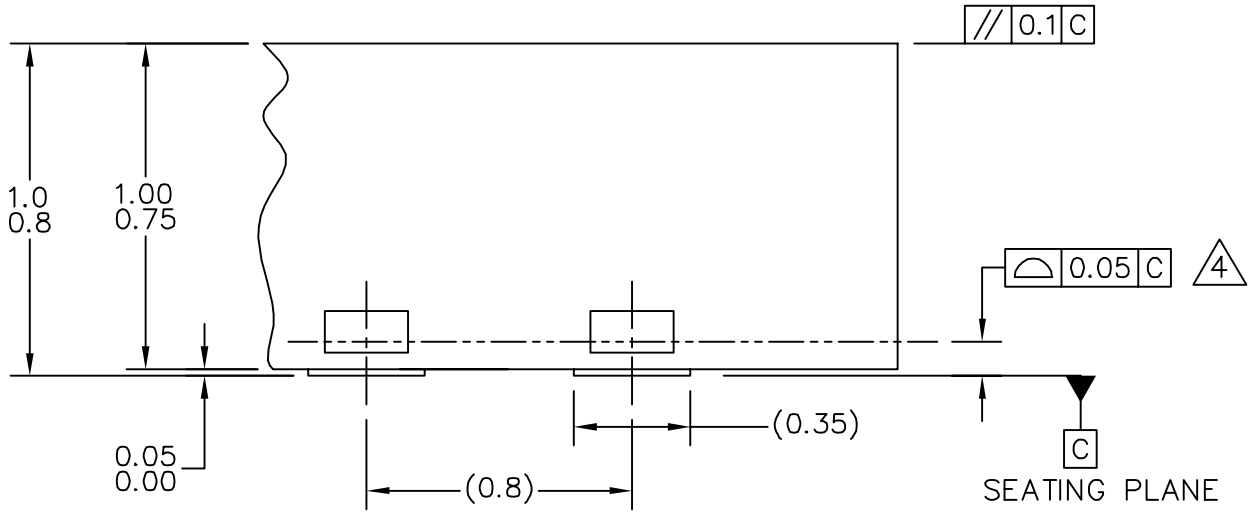
Table A-8. XOSC and ICS Specifications (Temperature Range = –40 to 125°C Ambient)

Characteristic	Symbol	Min	Typ	Max	Unit
Internal reference frequency — factory trimmed at V_{DD} = 3.6V and temperature = 25°**C**	f_{int_ft}	—	31.25	—	kHz
Oscillator crystal or resonator (EREFS = 1, ERCLKEN = 1) Low range (RANGE = 0) High range (RANGE = 1) FEE or FBE mode [1] High range (RANGE = 1), high gain (HGO = 1), FBELP mode High range (RANGE = 1), low power (HGO = 0), FBELP mode	 f_{lo} f_{hi} f_{hi} f_{hi}	 32 1 1 1	 — — — —	 38.4 5 16 8	 kHz MHz MHz MHz
Load capacitors	C_1 C_2	See Note [2]			
Feedback resistor Low range (32 kHz to 38.4 kHz) High range (1 MHz to 16 MHz)	R_F		10 1		MΩ MΩ
Series resistor — Low range Low Gain (HGO = 0) High Gain (HGO = 1)	R_S	— —	0 100	— —	kΩ
Series resistor — High range Low Gain (HGO = 0) High Gain (HGO = 1) ≥ 8 MHz 4 MHz 1 MHz	R_S	 — — —	 0 0 0	 0 10 20	kΩ
Crystal start-up time [3, 4] Low range, low power Low range, high power High range, low power High range, high power	 t_{CSTL} t_{CSTH}	 — — — —	 200 400 5 15	 — — — —	ms
Internal reference start-up time	t_{IRST}	—	60	100	µs
Square wave input clock frequency (EREFS = 0, ERCLKEN = 1) FEE or FBE mode [2] FBELP mode	f_{extal}	0.03125 0	— —	5 20	MHz MHz
Internal reference frequency - untrimmed[5]	f_{int_ut}	25	32.7	41.66	kHz
Internal reference frequency - trimmed	f_{int_t}	31.25	—	39.06	kHz
DCO output frequency range - untrimmed[5] f_{dco}= 512 * f_{int_ut}	f_{dco_ut}	12.8	16.8	21.33	MHz
DCO output frequency range - trimmed	f_{dco_t}	16	—	20	MHz

Table A-11. SPI Timing (continued)

No.	Function	Symbol	Min	Max	Unit
4	Clock (SPSCK) high or low time Master Slave	t_{WSPSCK}	 $t_{cyc} - 30$ $t_{cyc} - 30$	 1024 t_{cyc} —	 ns ns
5	Data setup time (inputs) Master Slave	t_{SU}	 15 15	 — —	 ns ns
6	Data hold time (inputs) Master Slave	t_{HI}	 0 25	 — —	 ns ns
7	Slave access time	t_a	—	1	t_{cyc}
8	Slave MISO disable time	t_{dis}	—	1	t_{cyc}
9	Data valid (after SPSCK edge) Master Slave	t_v	 — —	 25 25	 ns ns
10	Data hold time (outputs) Master Slave	t_{HO}	 0 0	 — —	 ns ns
11	Rise time Input Output	 t_{RI} t_{RO}	 — —	 $t_{cyc} - 25$ 25	 ns ns
12	Fall time Input Output	 t_{FI} t_{FO}	 — —	 $t_{cyc} - 25$ 25	 ns ns





DETAIL G

MECHANICAL OUTLINE

PRINT VERSION NOT TO SCALE

TITLE: THERMALLY ENHANCED DUAL FLAT NO LEAD PACKAGE (DFN) 8 TERMINAL, 0.8 PITCH (4 X 4 X 1)	DOCUMENT NO: 98ARL10557D	REV: B
	CASE NUMBER: 1452–02	28 DEC 2005
	STANDARD: NON–JEDEC	