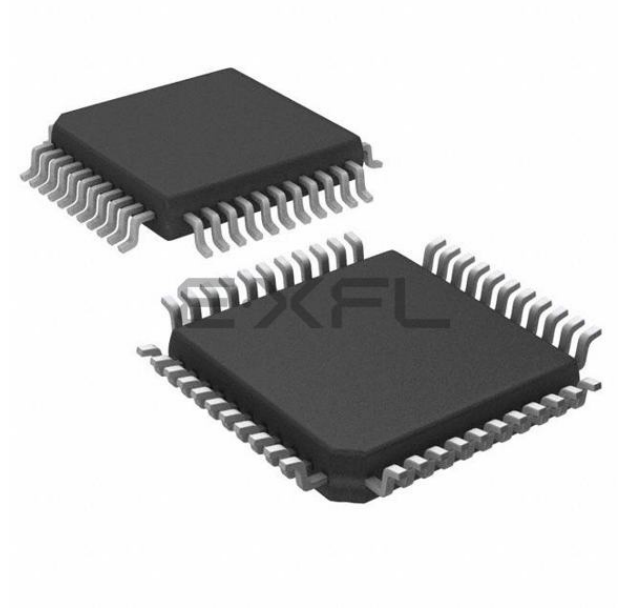


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gt16acfbe

Contents

Section Number **Title** **Page**

Chapter 1
Device Overview

1.1 Introduction 19
1.1.1 Devices in the MC9S08GT16A/GT8A Series 19
1.1.2 MCU Block Diagram 19
1.2 System Clock Distribution 21

Chapter 2
Pins and Connections

2.1 Introduction 23
2.2 Device Pin Assignment 23
2.3 Recommended System Connections 27
2.3.1 V_{DD}, V_{SS}, V_{DDAD}, V_{SSAD}, V_{REFH}, V_{REFL} — Power and Voltage References 28
2.3.2 PTG1/XTAL, PTG2/EXTAL — Oscillator 28
2.3.3 $\overline{\text{RESET}}$ — External Reset Pin 29
2.3.4 PTG0/BKGD/MS — Background / Mode Select 29
2.3.5 IRQ — External Interrupt Request Pin 30
2.3.6 General-Purpose I/O and Peripheral Ports 30
2.3.7 Signal Properties Summary 31

Chapter 3
Modes of Operation

3.1 Introduction 33
3.1.1 Features 33
3.2 Run Mode 33
3.3 Active Background Mode 33
3.4 Wait Mode 34
3.5 Stop Modes 35
3.5.1 Stop1 Mode 35
3.5.2 Stop2 Mode 35
3.5.3 Stop3 Mode 36
3.5.4 Active BDM Enabled in Stop Mode 37
3.5.5 LVD Enabled in Stop Mode 37
3.5.6 On-Chip Peripheral Modules in Stop Modes 38

15.4.3.2 Debug Comparator A Low Register (DBGCAL) 254
15.4.3.3 Debug Comparator B High Register (DBGCBH) 254
15.4.3.4 Debug Comparator B Low Register (DBGCBL) 254
15.4.3.5 Debug FIFO High Register (DBGFH) 255
15.4.3.6 Debug FIFO Low Register (DBGFL) 255
15.4.3.7 Debug Control Register (DBGC) 256
15.4.3.8 Debug Trigger Register (DBGT) 257
15.4.3.9 Debug Status Register (DBGS) 258

Appendix A
Electrical Characteristics

A.1 Introduction 259
A.2 Parameter Classification 259
A.3 Absolute Maximum Ratings 259
A.4 Thermal Characteristics 260
A.5 Electrostatic Discharge (ESD) Protection Characteristics 262
A.6 DC Characteristics 262
A.7 Supply Current Characteristics 266
A.8 ATD Characteristics 272
A.9 Internal Clock Generation Module Characteristics 274
A.9.1 ICG Frequency Specifications 275
A.10 AC Characteristics 276
A.10.1 Control Timing 277
A.10.2 Timer/PWM (TPM) Module Timing 278
A.10.3 SPI Timing 280
A.11 FLASH Specifications 283

Appendix B
Ordering Information and Mechanical Drawings

B.1 Ordering Information 285
B.1.1 Device Numbering Scheme 285
B.2 Mechanical Drawings 285

4.1.1 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the Freescale-provided equate file for the MC9S08GT16A/GT8A. For more details about resets, interrupts, interrupt priority, and local interrupt mask controls, refer to Chapter 5, "Resets, Interrupts, and System Configuration."

Table 4-1. Reset and Interrupt Vectors

Address (High/Low)	Vector	Vector Name
0xFFC0:FFC1 ⇕ 0xFFCA:FFCB	Unused Vector Space (available for user program)	
0xFFCC:FFCD	RTI	Vrti
0xFFCE:FFCF	IIC	Viic
0xFFD0:FFD1	ATD Conversion	Vatd
0xFFD2:FFD3	Keyboard	Vkeyboard
0xFFD4:FFD5	SCI2 Transmit	Vsci2tx
0xFFD6:FFD7	SCI2 Receive	Vsci2rx
0xFFD8:FFD9	SCI2 Error	Vsci2err
0xFFDA:FFDB	SCI1 Transmit	Vsci1tx
0xFFDC:FFDD	SCI1 Receive	Vsci1rx
0xFFDE:FFDF	SCI1 Error	Vsci1err
0xFFE0:FFE1	SPI	Vspi
0xFFE2:FFE3	TPM2 Overflow	Vtpm2ovf
0xFFE4:FFE9	Unused Vector Space (available for user program)	
0xFFEA:FFEB	TPM2 Channel 1	Vtpm2ch1
0xFFEC:FFED	TPM2 Channel 0	Vtpm2ch0
0xFFEE:FFEF	TPM1 Overflow	Vtpm1ovf
0xFFF0:FFF1	TPM1 Channel 2	Vtpm1ch2
0xFFF2:FFF3	TPM1 Channel 1	Vtpm1ch1
0xFFF4:FFF5	TPM1 Channel 0	Vtpm1ch0
0xFFF6:FFF7	ICG	Vicg
0xFFF8:FFF9	Low Voltage Detect	Vlvd
0xFFFA:FFFB	IRQ	Virq
0xFFFC:FFFD	SWI	Vswi
0xFFFE:FFFF	Reset	Vreset

NVPROT) must be programmed to logic 0 to enable block protection. Therefore the value 0xDE must be programmed into NVPROT to protect addresses 0xE000 through 0xFFFF.

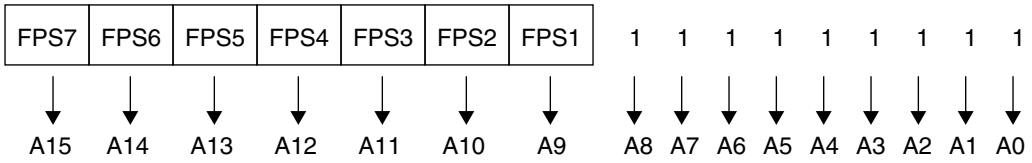


Figure 4-4. Block Protection Mechanism

One use for block protection is to block protect an area of FLASH memory for a bootloader program. This bootloader program then can be used to erase the rest of the FLASH memory and reprogram it. Because the bootloader is protected, it remains intact even if MCU power is lost in the middle of an erase and reprogram operation.

4.4.7 Vector Redirection

Whenever any block protection is enabled, the reset and interrupt vectors will be protected. Vector redirection allows users to modify interrupt vector information without unprotecting bootloader and reset vector space. Vector redirection is enabled by programming the FNORED bit in the NVOPT register located at address 0xFFBF to zero. For redirection to occur, at least some portion but not all of the FLASH memory must be block protected by programming the NVPROT register located at address 0xFFBD. All of the interrupt vectors (memory locations 0xFFC0–0xFFFD) are redirected, while the reset vector (0xFFFE:FFFF) is not.

For example, if 512 bytes of FLASH are protected, the protected address region is from 0xFE00 through 0xFFFF. The interrupt vectors (0xFFC0–0xFFFD) are redirected to the locations 0xFDC0–0xFDFD. Now, if an SPI interrupt is taken for instance, the values in the locations 0xFDE0:FDE1 are used for the vector instead of the values in the locations 0xFFE0:FFE1. This allows the user to reprogram the unprotected portion of the FLASH with new program code including new interrupt vector values while leaving the protected area, which includes the default vector locations, unchanged.

4.5 Security

The MC9S08GT16A/GT8A includes circuitry to prevent unauthorized access to the contents of FLASH and RAM memory. When security is engaged, FLASH and RAM are considered secure resources. Direct-page registers, high-page registers, and the background debug controller are considered unsecured resources. Programs executing within secure memory have normal access to any MCU memory locations and resources. Attempts to access a secure memory location with a program executing from an unsecured memory space or through the background debug interface are blocked (writes are ignored and reads return all 0s).

Security is engaged or disengaged based on the state of two nonvolatile register bits (SEC01:SEC00) in the FOPT register. During reset, the contents of the nonvolatile location NVOPT are copied from FLASH into the working FOPT register in high-page register space. A user engages security by programming the NVOPT location which can be done at the same time the FLASH memory is programmed. The 1:0 state disengages security while the other three combinations engage security. Notice the erased state (1:1)

- Computer operating properly (COP) watchdog timer
- Illegal opcode detect
- Illegal address detect
- Background debug forced reset
- The reset pin ($\overline{\text{RESET}}$)
- Clock generator loss of lock and loss of clock reset

Each of these sources, with the exception of the background debug forced reset, has an associated bit in the system reset status register. Whenever the MCU enters reset, the internal clock generator (ICG) module switches to self-clocked mode with the frequency of f_{Self_reset} selected. The reset pin is driven low for 34 internal bus cycles where the internal bus frequency is half the ICG frequency. After the 34 cycles are completed, the pin is released and will be pulled up by the internal pullup resistor, unless it is held low externally. After the pin is released, it is sampled after another 38 cycles to determine whether the reset pin is the cause of the MCU reset.

5.3 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP timer periodically. If the application program gets lost and fails to reset the COP before it times out, a system reset is generated to force the system back to a known starting point. The COP watchdog is enabled by the COPE bit in SOPT (see Section 5.7.4, “System Options Register (SOPT)” for additional information). The COP timer is reset by writing any value to the address of SRS. This write does not affect the data in the read-only SRS. Instead, the act of writing to this address is decoded and sends a reset signal to the COP timer.

After any reset, the COP timer is enabled. This provides a reliable way to detect code that is not executing as intended. If the COP watchdog is not used in an application, it can be disabled by clearing the COPE bit in the write-once SOPT register. Also, the COPT bit can be used to choose one of two timeout periods (2^{18} or 2^{13} cycles of the bus rate clock). Even if the application will use the reset default settings in COPE and COPT, the user should still write to write-once SOPT during reset initialization to lock in the settings. That way, they cannot be changed accidentally if the application program gets lost.

The write to SRS that services (clears) the COP timer should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails.

When the MCU is in active background mode, the COP timer is temporarily disabled.

5.4 Interrupts

Interrupts provide a way to save the current CPU status and registers, execute an interrupt service routine (ISR), and then restore the CPU status so processing resumes where it left off before the interrupt. Other than the software interrupt (SWI), which is a program instruction, interrupts are caused by hardware events such as an edge on the IRQ pin or a timer-overflow event. The debug module can also generate an SWI under certain circumstances.

Table 5-1. Vector Summary

Vector Priority	Vector Number	Address (High/Low)	Vector Name	Module	Source	Enable	Description
Lower	26 through 31	0xFFC0/FFC1 through 0xFFCA/FFCB	Unused Vector Space (available for user program)				
	25	0xFFCC/FFCD	Vrti	System control	RTIF	RTIE	Real-time interrupt
	24	0xFFCE/FFCF	Viic	IIC	IICIS	IICIE	IIC control
	23	0xFFD0/FFD1	Vatd	ATD	COCO	AIEN	AD conversion complete
	22	0xFFD2/FFD3	Vkeyboard	KBI	KBF	KBIE	Keyboard pins
	21	0xFFD4/FFD5	Vsci2tx	SCI2	TDRE TC	TIE TCIE	SCI2 transmit
	20	0xFFD6/FFD7	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2 receive
	19	0xFFD8/FFD9	Vsci2err	SCI2	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI2 error
	18	0xFFDA/FFDB	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1 transmit
	17	0xFFDC/FFDD	Vsci1rx	SCI1	IDLE RDRF	ILIE RIE	SCI1 receive
	16	0xFFDE/FFDF	Vsci1err	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1 error
	15	0xFFE0/FFE1	Vspi	SPI	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI
	14	0xFFE2/FFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow
	11 through 13	0xFFEC/FFED through 0xFFE4/FFE5	Unused Vector Space (available for user program)				
	10	0xFFEA/FFEB	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1
	9	0xFFEC/FFED	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0
	8	0xFFEE/FFEF	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow
	7	0xFFF0/FFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 channel 2
	6	0xFFF2/FFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 channel 1
	5	0xFFF4/FFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 channel 0
	4	0xFFF6/FFF7	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG
	3	0xFFF8/FFF9	Vlvd	System control	LVDF	LVDIE	Low-voltage detect
	2	0xFFFA/FFFB	Virq	IRQ	IRQF	IRQIE	IRQ pin
	1	0xFFFC/FFFD	Vswi	Core	SWI Instruction	—	Software interrupt
Higher	0	0xFFFE/FFFF	Vreset	System control	COP LVD $\overline{\text{RESET}}$ pin Illegal opcode	COPE LVDRE — —	Watchdog timer Low-voltage detect External pin Illegal opcode

5.7.8 System Power Management Status and Control 2 Register (SPMSC2)

This register is used to report the status of the low voltage warning function, and to configure the stop mode behavior of the MCU.

	7	6	5	4	3	2	1	0
R	LVWF	0	LVDV	LVWV	PPDF	0	PDC	PPDC
W		LVWACK				PPDACK		
Power-on reset:	0 Note (1)	0	0	0	0	0	0	0
LVD reset:	0 Note (1)	0	U	U	0	0	0	0
Any other reset:	0 Note (1)	0	U	U	0	0	0	0

= Unimplemented or Reserved U = Unaffected by reset

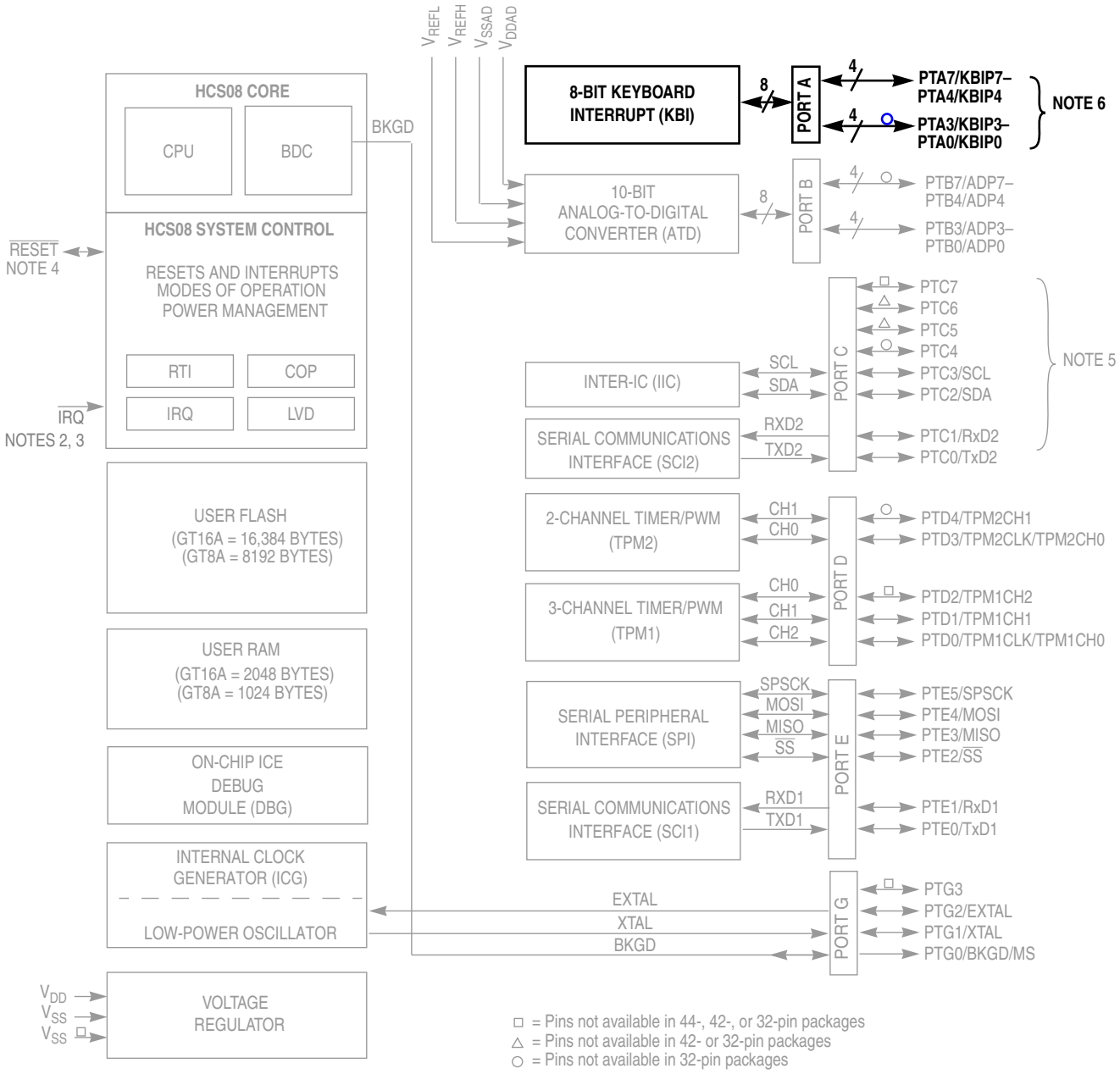
1 LVWF will be set in the case when V_{Supply} transitions below the trip point or after reset and V_{Supply} is already below V_{LVW}.

Figure 5-10. System Power Management Status and Control 2 Register (SPMSC2)

Table 5-11. SPMSC2 Field Descriptions

Field	Description
7 LVWF	**Low-Voltage Warning Flag** — The LVWF bit indicates the low voltage warning status. 0 Low voltage warning **not** present. 1 Low voltage warning is present or was present.
6 LVWACK	**Low-Voltage Warning Acknowledge** — The LVWACK bit is the low-voltage warning acknowledge. Writing a 1 to LVWACK clears LVWF to 0 if a low voltage warning is not present.
5 LVDV	**Low-Voltage Detect Voltage Select** — The LVDV bit selects the LVD trip point voltage (V_{LVD}). 0 Low trip point selected ($V_{LVD} = V_{LVDL}$). 1 High trip point selected ($V_{LVD} = V_{LVDH}$).
4 LVWV	**Low-Voltage Warning Voltage Select** — The LVWV bit selects the LVW trip point voltage (V_{LVW}). 0 Low trip point selected ($V_{LVW} = V_{LVWL}$). 1 High trip point selected ($V_{LVW} = V_{LVWH}$).
3 PPDF	**Partial Power Down Flag** — The PPDF bit indicates that the MCU has exited the stop2 mode. 0 Not stop2 mode recovery. 1 Stop2 mode recovery.
2 PPDACK	**Partial Power Down Acknowledge** — Writing a 1 to PPDACK clears the PPDF bit.
1 PDC	**Power Down Control** — The write-once PDC bit controls entry into the power down (stop2 and stop1) modes. 0 Power down modes are disabled. 1 Power down modes are enabled.
0 PPDC	**Partial Power Down Control** — The write-once PPDC bit controls which power down mode, stop1 or stop2, is selected. 0 Stop1, full power down, mode enabled if PDC set. 1 Stop2, partial power down, mode enabled if PDC set.

- Eight port B pins shared with ATD
- Eight high-current port C pins shared with SCI2 and IIC
- Five port D pins shared with TPM1 and TPM2
- Six port E pins shared with SCI1 and SPI
- Four port G pins shared with EXTAL, XTAL, and BKGD/MS



NOTES:

1. Port pins are software configurable with pullup device if input port.
2. Pin contains pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to VDD. IRQ should not be driven above VDD.
4. Pin contains integrated pullup device.
5. High current drive
6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown available when KBI enabled (KBIPn = 1).

Figure 7-2. Block Diagram Highlighting the KBI Module

0	=	Bit forced to 0
1	=	Bit forced to 1
	=	Bit set or cleared according to results of operation
U	=	Undefined after the operation

Machine coding notation

dd	=	Low-order 8 bits of a direct address 0x0000–0x00FF (high byte assumed to be 0x00)
ee	=	Upper 8 bits of 16-bit offset
ff	=	Lower 8 bits of 16-bit offset or 8-bit offset
ii	=	One byte of immediate data
jj	=	High-order byte of a 16-bit immediate data value
kk	=	Low-order byte of a 16-bit immediate data value
hh	=	High-order byte of 16-bit extended address
ll	=	Low-order byte of 16-bit extended address
rr	=	Relative offset

Source form

Everything in the source forms columns, *except expressions in italic characters,* is literal information that must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic is always a literal expression. All commas, pound signs (#), parentheses, and plus signs (+) are literal characters.

n	—	Any label or expression that evaluates to a single integer in the range 0–7
opr8i	—	Any label or expression that evaluates to an 8-bit immediate value
opr16i	—	Any label or expression that evaluates to a 16-bit immediate value
opr8a	—	Any label or expression that evaluates to an 8-bit value. The instruction treats this 8-bit value as the low order 8 bits of an address in the direct page of the 64-Kbyte address space (0x00xx).
opr16a	—	Any label or expression that evaluates to a 16-bit value. The instruction treats this value as an address in the 64-Kbyte address space.
oprx8	—	Any label or expression that evaluates to an unsigned 8-bit value, used for indexed addressing
oprx16	—	Any label or expression that evaluates to a 16-bit value. Because the HCS08 has a 16-bit address bus, this can be either a signed or an unsigned value.
rel	—	Any label or expression that refers to an address that is within –128 to +127 locations from the next address after the last byte of object code for the current instruction. The assembler will calculate the 8-bit signed offset and include it in the object code for this instruction.

Address modes

INH	=	Inherent (no operands)
IMM	=	8-bit or 16-bit immediate
DIR	=	8-bit direct
EXT	=	16-bit extended

Table 8-2. HCS08 Instruction Set Summary (Sheet 3 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles[1]
			V	H	I	N	Z	C				
BRCLR *n,opr8a,rel*	Branch if Bit *n* in Memory Clear	Branch if (Mn) = 0	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	Uses 3 Bus Cycles	–	–	–	–	–	–	REL	21	rr	3
BRSET *n,opr8a,rel*	Branch if Bit *n* in Memory Set	Branch if (Mn) = 1	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n,opr8a*	Set Bit *n* in Memory	Mn ← 1	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 0x0002 push (PCL); SP ← (SP) – 0x0001 push (PCH); SP ← (SP) – 0x0001 PC ← (PC) + *rel*	–	–	–	–	–	–	REL	AD	rr	5
CBEQ *oprx8,rel* CBEQA *#opr8i,rel* CBEQX *#opr8i,rel* CBEQ *oprx8*,X+,*rel* CBEQ ,X+,*rel* CBEQ *oprx8*,SP,*rel*	Compare and Branch if Equal	Branch if (A) = (M) Branch if (A) = (M) Branch if (X) = (M) Branch if (A) = (M) Branch if (A) = (M) Branch if (A) = (M)	–	–	–	–	–	–	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff rr	5 4 4 5 5 6
CLC	Clear Carry Bit	C ← 0	–	–	–	–	–	0	INH	98		1
CLI	Clear Interrupt Mask Bit	I ← 0	–	–	0	–	–	–	INH	9A		1
CLR *opr8a* CLRA CLRX CLRH CLR *oprx8*,X CLR ,X CLR *oprx8*,SP	Clear	M ← 0x00 A ← 0x00 X ← 0x00 H ← 0x00 M ← 0x00 M ← 0x00 M ← 0x00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	5 1 1 1 5 4 6
CMP *#opr8i* CMP *opr8a* CMP *opr16a* CMP *oprx16*,X CMP *oprx8*,X CMP ,X CMP *oprx16*,SP CMP *oprx8*,SP	Compare Accumulator with Memory	(A) – (M) (CCR Updated But Operands Not Changed)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A1 B1 C1 D1 E1 F1 9ED1 9EE1	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
COM *oprx8* COMA COMX COM *oprx8*,X COM ,X COM *oprx8*,SP	Complement (One's Complement)	M ← (M̄)= 0xFF – (M) A ← (Ā) = 0xFF – (A) X ← (X̄) = 0xFF – (X) M ← (M̄) = 0xFF – (M) M ← (M̄) = 0xFF – (M) M ← (M̄) = 0xFF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	5 1 1 5 4 6
CPHX *opr16a* CPHX *#opr16i* CPHX *opr8a* CPHX *oprx8*,SP	Compare Index Register (H:X) with Memory	(H:X) – (M:M + 0x0001) (CCR Updated But Operands Not Changed)	↕	–	–	↕	↕	↕	EXT IMM DIR SP1	3E 65 75 9EF3	hh ll jj kk dd ff	6 3 5 6

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.

10.4.2 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.

10.4.2.1 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.

10.4.2.2 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.

10.4.2.3 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 10-11 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.

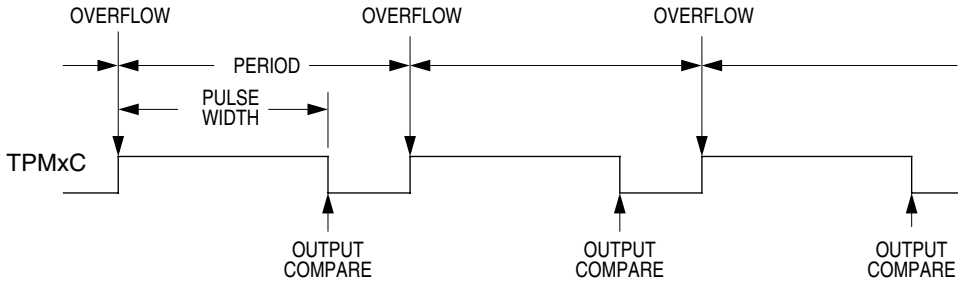


Figure 10-11. PWM Period and Pulse Width (ELSnA = 0)

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.)

10.5.3 Channel Event Interrupt Description

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

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select rising edges, falling edges, any edge, or no edge (off) as the edge that triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the 2-step sequence described in Section 10.5.1, “Clearing Timer Interrupt Flags.”

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the 2-step sequence described in Section 10.5.1, “Clearing Timer Interrupt Flags.”

10.5.4 PWM End-of-Duty-Cycle Events

For channels that are configured for PWM operation, there are two possibilities:

- When the channel is configured for edge-aligned PWM, the channel flag is set when the timer counter matches the channel value register that marks the end of the active duty cycle period.
- When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle, which are the times when the timer counter matches the channel value register.

The flag is cleared by the 2-step sequence described in Section 10.5.1, “Clearing Timer Interrupt Flags.”

12.3 Modes of Operation

12.3.1 SPI in Stop Modes

The SPI is disabled in all stop modes, regardless of the settings before executing the STOP instruction. During either stop1 or stop2 mode, the SPI module will be fully powered down. Upon wake-up from stop1 or stop2 mode, the SPI module will be in the reset state. During stop3 mode, clocks to the SPI module are halted. No registers are affected. If stop3 is exited with a reset, the SPI will be put into its reset state. If stop3 is exited with an interrupt, the SPI continues from the state it was in when stop3 was entered.

12.4 Register Definition

The SPI has five 8-bit registers to select SPI options, control baud rate, report SPI status, and for transmit/receive data.

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

12.4.1 SPI Control Register 1 (SPIC1)

This read/write register includes the SPI enable control, interrupt enables, and configuration options.

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Figure 12-6. SPI Control Register 1 (SPIC1)

Table 12-1. SPIC1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable (for SPRF and MODF)** — This is the interrupt enable for SPI receive buffer full (SPRF) and mode fault (MODF) events. 0 Interrupts from SPRF and MODF inhibited (use polling) 1 When SPRF or MODF is 1, request a hardware interrupt
6 SPE	**SPI System Enable** — Disabling the SPI halts any transfer that is in progress and initializes internal state machines. SPRF is cleared and SPTEF is set to indicate the SPI transmit data buffer is empty. 0 SPI system inactive 1 SPI system enabled
5 SPTIE	**SPI Transmit Interrupt Enable** — This is the interrupt enable bit for SPI transmit buffer empty (SPTEF). 0 Interrupts from SPTEF inhibited (use polling) 1 When SPTEF is 1, hardware interrupt requested

Chapter 14
Analog-to-Digital Converter (S08ATDV3)

The MC9S08GT16A/GT8A provides one 8-channel analog-to-digital (ATD) module. The eight ATD channels share port B. Each channel individually can be configured for general-purpose I/O or for ATD functionality. All features of the ATD module as described in this section are available on the MC9S08GT16A/GT8A. Electrical parametric information for the ATD may be found in Appendix A, “Electrical Characteristics.”

of how straight the line is (how far it deviates from a straight line). The adjusted ideal transition voltage is:

Eqn. 14-6

$$\text{Adjusted Ideal Trans. V} = \frac{(\text{Current Code} - 1/2)}{2^N} * ((V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS}))$$

- Zero scale error (E_{ZS}) — This is the difference between the transition voltage to the first valid code and the ideal transition to that code. Normally, it is defined as the difference between the actual and ideal transition to code $001, but in some cases the first transition may be to a higher code. The ideal transition to any code is:

Eqn. 14-7

$$\text{Ideal Transition V} = \frac{(\text{Current Code} - 1/2)}{2^N} *(V_{REFH} - V_{REFL})$$

- Full scale error (E_{FS}) — This is the difference between the transition voltage to the last valid code and the ideal transition to that code. Normally, it is defined as the difference between the actual and ideal transition to code $3FF, but in some cases the last transition may be to a lower code. The ideal transition to any code is:

Eqn. 14-8

$$\text{Ideal Transition V} = \frac{(\text{Current Code} - 1/2)}{2^N} *(V_{REFH} - V_{REFL})$$

- Total unadjusted error (E_{TU}) — This is the difference between the transition voltage to a given code and the ideal straight-line transfer function. An alternate definition (with the same result) is the difference between the actual transfer function and the ideal straight-line transfer function. This measure of error includes inherent quantization error and all forms of circuit error (INL, DNL, zero-scale, and full-scale) except input leakage error, which is not due to the ATD.
- Input leakage error (E_{IL}) — This is the error between the transition voltage to the current code and the ideal transition to that code that is the result of input leakage across the real portion of the impedance of the network that drives the analog input. This error is a system-observable error which is not inherent to the ATD, so it is not added to total error. This error is:

$$\mathbf{E_{IL}\ (in\ V) = input\ leakage * R_{AS}}$$ *Eqn. 14-9*

There are two other forms of error which are not specified which can also affect ATD accuracy. These are:

- Sampling error (E_S) — The error due to inadequate time to charge the ATD circuitry
- Noise error (E_N) — The error due to noise on V_{AIN}, V_{REFH}, or V_{REFL} due to either direct coupling (noise source capacitively coupled directly on the signal) or power supply (V_{DDAD}, V_{SSAD}, V_{DD}, and V_{SS}) noise interfering with the ATD's ability to resolve the input accurately. The error due to internal sources can be reduced (and specified operation achieved) by operating the ATD conversion in wait mode and ceasing all IO activity. Reducing the error due to external sources is dependent on system activity and board layout.

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.

15.3.5 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.

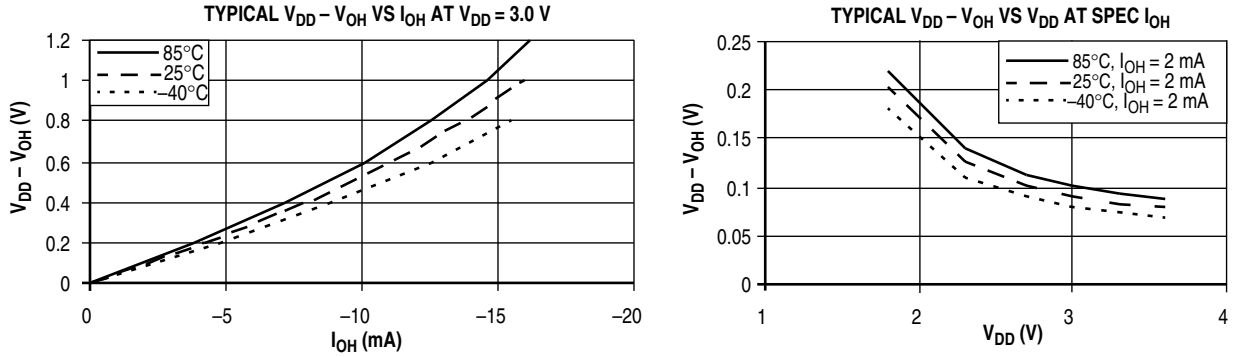


Figure A-5. Typical High-Side (Source) Characteristics (Ports A, B, D, E, and G)

A.7 Supply Current Characteristics

Table A-7. Supply Current Characteristics

Parameter	Symbol	V_{DD} (V)	Typical[1]	Max[2]	Temp. (°C)
Run supply current[3] measured at (CPU clock = 2 MHz, f_{Bus} = 1 MHz)	RI_{DD}	3	0.8 mA	1.3 mA[4]	125
		2	0.66 mA	1.0 mA(4)	125
Run supply current (3) measured at (CPU clock = 16 MHz, f_{Bus} = 8 MHz)	RI_{DD}	3	4.3 mA	7.0 mA[5]	125
		2	3.3 mA	4.5 mA(4)	125
Stop1 mode supply current	$S1I_{DD}$	3	25 nA	0.6 µA(4) 1.8 µA(4) 4.0 µA(5) 13 µA(5)	55 70 85 125
		2	20 nA	500 nA(4) 1.5 µA(4) 3.3 µA(4) 10 µA(4)	55 70 85 125
Stop2 mode supply current	$S2I_{DD}$	3	550 nA	3.0 µA(4) 5.5 µA(4) 11 µA(5) 20 µA(5)	55 70 85 125
		2	400 nA	2.4 µA(4) 5.0 µA(4) 9.5 µA(4) 17 µA(4)	55 70 85 125

Table A-7. Supply Current Characteristics (continued)

Parameter	Symbol	V_{DD} (V)	Typical[1]	Max[2]	Temp. (°C)
Stop3 mode supply current	$S3I_{DD}$	3	675 nA	4.3 µA[(4)] 7.2 µA[(4)] 17.0 µA[(5)] 45 µA[(5)]	55 70 85 125
		2	500 nA	3.5 µA[(4)] 6.2 µA[(4)] 15.0 µA[(4)] 40 µA[(4)]	55 70 85 125
RTI adder to stop2 or stop3[6]		3	300 nA		
		2	300 nA		
LVI adder to stop3 (LVDSE = LVDE = 1)		3	70 µA		
		2	60 µA		
Adder to stop3 for oscillator enabled[7] (OSCSTEN =1)		3	5 µA		
		2	5 µA		
Adder for loss-of-clock enabled (LOCD=0)		3	9 µA		
		2	9 µA		
Adder for high gain oscillator enabled (HGO=1)		3	28 µA		
		2	2 µA		

1 Typicals are measured at 25°C. See Table A-6 through Table A-9 for typical curves across voltage/temperature.

2 Values given here are preliminary estimates prior to completing characterization.

3 All modules except ATD active, ICG configured for FBE, and does not include any dc loads on port pins

4 Values are characterized but not tested on every part.

5 Every unit tested to this parameter. All other values in the Max column are guaranteed by characterization.

6 Most customers are expected to find that auto-wakeup from stop2 or stop3 can be used instead of the higher current wait mode. Wait mode typical is 560 µA at 3 V and 422 µA at 2V with f_{Bus} = 1 MHz.

7 Values given under the following conditions: low range operation (RANGE = 0), low power mode (HGO = 0), clock monitor disabled (LOCD = 1).