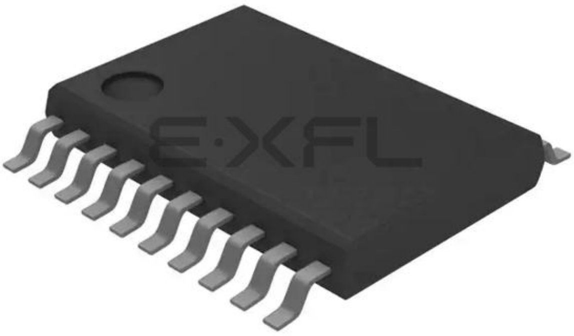


Welcome to E-XFL.COM

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, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg4e2vtjr

Table 4-2. Direct-Page Register Summary (Sheet 1 of 3)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0000	PTAD	0	0	—	—	PTAD3	PTAD2	PTAD1	PTAD0
0x0001	PTADD	0	0	—	—	PTADD3	PTADD2	PTADD1	PTADD0
0x0002	PTBD	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0x0003	PTBDD	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0004	PTCD	0	0	0	0	PTCD3	PTCD2	PTCD1	PTCD0
0x0005	PTCDD	0	0	0	0	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x0006– 0x000D	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x000E	ACMPSC	ACME	ACBGS	ACF	ACIE	ACO	ACOPE	ACMOD1	ACMOD0
0x000F	Reserved	—	—	—	—	—	—	—	—
0x0010	ADCSC1	COCO	AIEN	ADCO	ADCH				
0x0011	ADCSC2	ADACT	ADTRG	ACFE	ACFGT	—	—	—	—
0x0012	ADCRH	0	0	0	0	0	0	ADR9	ADR8
0x0013	ADCRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0x0014	ADCVH	0	0	0	0	0	0	ADCV9	ADCV8
0x0015	ADCVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
0x0016	ADCFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0x0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0x0018	APCTL2	0	0	0	0	ADPC11	ADPC10	ADPC9	ADPC8
0x0019– 0x001B	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x001C	MTIMSC	TOF	TOIE	TRST	TSTP	0	0	0	0
0x001D	MTIMCLK	0	0	CLKS		PS			
0x001E	MTIMCNT	CNT							
0x001F	MTIMMOD	MOD							
0x0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x0022	TPM1CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0028	TPM1C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0029	TPM1C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x002A	TPM1C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x002B– 0x0037	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x0038	SCIBDH	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8

4.5.5 Access Errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will cause the access error flag (FACCERR) in FSTAT to be set. FACCERR must be cleared by writing a 1 to FACCERR in FSTAT before any command can be processed.

- Writing to a FLASH address before the internal FLASH clock frequency has been set by writing to the FCDIV register
- Writing to a FLASH address while FCBEF is not set (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a FLASH address before launching the previous command (There is only one write to FLASH for every command.)
- Writing a second time to FCMD before launching the previous command (There is only one write to FCMD for every command.)
- Writing to any FLASH control register other than FCMD after writing to a FLASH address
- Writing any command code other than the five allowed codes (0x05, 0x20, 0x25, 0x40, or 0x41) to FCMD
- Writing any FLASH control register other than the write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD
- The MCU enters stop mode while a program or erase command is in progress (The command is aborted.)
- Writing the byte program, burst program, or page erase command code (0x20, 0x25, or 0x40) with a background debug command while the MCU is secured (The background debug controller can only do blank check and mass erase commands when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command

4.5.6 FLASH Block Protection

The block protection feature prevents the protected region of FLASH from program or erase changes. Block protection is controlled through the FLASH protection register (FPROT). When enabled, block protection begins at any 512 byte boundary below the last address of FLASH, 0xFFFF. (See Section 4.7.4, “FLASH Protection Register (FPROT and NVPROT)”).

After exit from reset, FPROT is loaded with the contents of the NVPROT location, which is in the nonvolatile register block of the FLASH memory. FPROT cannot be changed directly from application software so a runaway program cannot alter the block protection settings. Because NVPROT is within the last 512 bytes of FLASH, if any amount of memory is protected, NVPROT is itself protected and cannot be altered (intentionally or unintentionally) by the application software. FPROT can be written through background debug commands, which allows a way to erase and reprogram a protected FLASH memory.

The block protection mechanism is illustrated in Figure 4-4. The FPS bits are used as the upper bits of the last address of unprotected memory. This address is formed by concatenating FPS7:FPS1 with logic 1 bits as shown. For example, to protect the last 1536 bytes of memory (addresses 0xFA00 through 0xFFFF), the FPS bits must be set to 1111 100, which results in the value 0xF9FF as the last address of unprotected memory. In addition to programming the FPS bits to the appropriate value, FPDIS (bit 0 of NVPROT)

6.6.1.5 Port A Drive Strength Selection Register (PTADS)

	7	6	5	4	3	2	1	0
R	0	0	R	R	PTADS3	PTADS2	PTADS1	PTADS0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 6-7. Drive Strength Selection for Port A Register (PTADS)

Table 6-6. PTADS Register Field Descriptions

Field	Description
5:4 Reserved	**Reserved Bits** — These bits are unused on this MCU, writes have no affect and could read as 1s or 0s.
3:0 PTADS[3:0]	**Output Drive Strength Selection for Port A Bits** — Each of these control bits selects between low and high output drive for the associated PTA pin. For port A pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port A bit n. 1 High output drive strength selected for port A bit n.

6.6.1.6 Port A Interrupt Status and Control Register (PTASC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTAIF	0	PTAIE	PTAMOD
W						PTAACK		
Reset:	0	0	0	0	0	0	0	0

Figure 6-8. Port A Interrupt Status and Control Register (PTASC)

Table 6-7. PTASC Register Field Descriptions

Field	Description
3 PTAIF	**Port A Interrupt Flag** — PTAIF indicates when a port A interrupt is detected. Writes have no effect on PTAIF. 0 No port A interrupt detected. 1 Port A interrupt detected.
2 PTAACK	**Port A Interrupt Acknowledge** — Writing a 1 to PTAACK is part of the flag clearing mechanism. PTAACK always reads as 0.
1 PTAIE	**Port A Interrupt Enable** — PTAIE determines whether a port A interrupt is requested. 0 Port A interrupt request not enabled. 1 Port A interrupt request enabled.
0 PTAMOD	**Port A Detection Mode** — PTAMOD (along with the PTAES bits) controls the detection mode of the port A interrupt pins. 0 Port A pins detect edges only. 1 Port A pins detect both edges and levels.

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

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						VH	I N Z C
TXS	Transfer Index Reg. to SP SP ← (H:X) – 0x0001	INH	94	2	fp	– –	– – – –
WAIT	Enable Interrupts; Wait for Interrupt I bit ← 0; Halt CPU	INH	8F	2+	fp...	– –	0 – – –

Source Form: Everything in the source forms columns, *except expressions in italic characters*, is literal information which must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic and the characters (# , ( ) and +) are always a 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 direct-page address (0x00xx).
opr16a Any label or expression that evaluates to a 16-bit address.
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, used for indexed addressing.
rel Any label or expression that refers to an address that is within –128 to +127 locations from the start of the next instruction.

Operation Symbols:

A Accumulator
CCR Condition code register
H Index register high byte
M Memory location
n Any bit
opr Operand (one or two bytes)
PC Program counter
PCH Program counter high byte
PCL Program counter low byte
rel Relative program counter offset byte
SP Stack pointer
SPL Stack pointer low byte
X Index register low byte
& Logical AND
| Logical OR
⊕ Logical EXCLUSIVE OR
( ) Contents of
+ Add
– Subtract, Negation (two's complement)
× Multiply
÷ Divide
Immediate value
← Loaded with
: Concatenated with

CCR Bits:

V Overflow bit
H Half-carry bit
I Interrupt mask
N Negative bit
Z Zero bit
C Carry/borrow bit

Addressing Modes:

DIR Direct addressing mode
EXT Extended addressing mode
IMM Immediate addressing mode
INH Inherent addressing mode
IX Indexed, no offset addressing mode
IX1 Indexed, 8-bit offset addressing mode
IX2 Indexed, 16-bit offset addressing mode
IX+ Indexed, no offset, post increment addressing mode
IX1+ Indexed, 8-bit offset, post increment addressing mode
REL Relative addressing mode
SP1 Stack pointer, 8-bit offset addressing mode
SP2 Stack pointer 16-bit offset addressing mode

Cycle-by-Cycle Codes:

f Free cycle. This indicates a cycle where the CPU does not require use of the system buses. An f cycle is always one cycle of the system bus clock and is always a read cycle.
p Progryam fetch; read from next consecutive location in program memory
r Read 8-bit operand
s Push (write) one byte onto stack
u Pop (read) one byte from stack
v Read vector from 0xFFxx (high byte first)
w Write 8-bit operand

CCR Effects:

Þ Set or cleared
– Not affected
U Undefined

10.1.2 Features

Key features of the ICS module follow. For device specific information, refer to the ICS Characteristics in the Electricals section of the documentation.

- Frequency-locked loop (FLL) is trimmable for accuracy
  - 0.2% resolution using internal 32kHz reference
  - 1.5% deviation over voltage and temperature using internal 32kHz reference
- Internal or external reference clocks up to 5MHz can be used to control the FLL
  - 3 bit select for reference divider is provided
- Internal reference clock has 9 trim bits available
- Internal or external reference clocks can be selected as the clock source for the MCU
- Whichever clock is selected as the source can be divided down
  - 2 bit select for clock divider is provided
    - Allowable dividers are: 1, 2, 4, 8
    - BDC clock is provided as a constant divide by 2 of the DCO output
- Control signals for a low power oscillator as the external reference clock are provided
  - HGO, RANGE, EREFS, ERCLKEN, EREFSTEN
- FLL Engaged Internal mode is automatically selected out of reset

10.1.3 Block Diagram

Figure 10-2 is the ICS block diagram.

Table 10-5. ICS Status and Control Register Field Descriptions (continued)

Field	Description
1	**OSC Initialization** — If the external reference clock is selected by ERCLKEN or by the ICS being in FEE, FBE, or FBELP mode, and if EREFS is set, then this bit is set after the initialization cycles of the external oscillator clock have completed. This bit is only cleared when either ERCLKEN or EREFS are cleared.
0	**ICS Fine Trim** — The FTRIM bit controls the smallest adjustment of the internal reference clock frequency. Setting FTRIM will increase the period and clearing FTRIM will decrease the period by the smallest amount possible.

10.4 Functional Description

10.4.1 Operational Modes

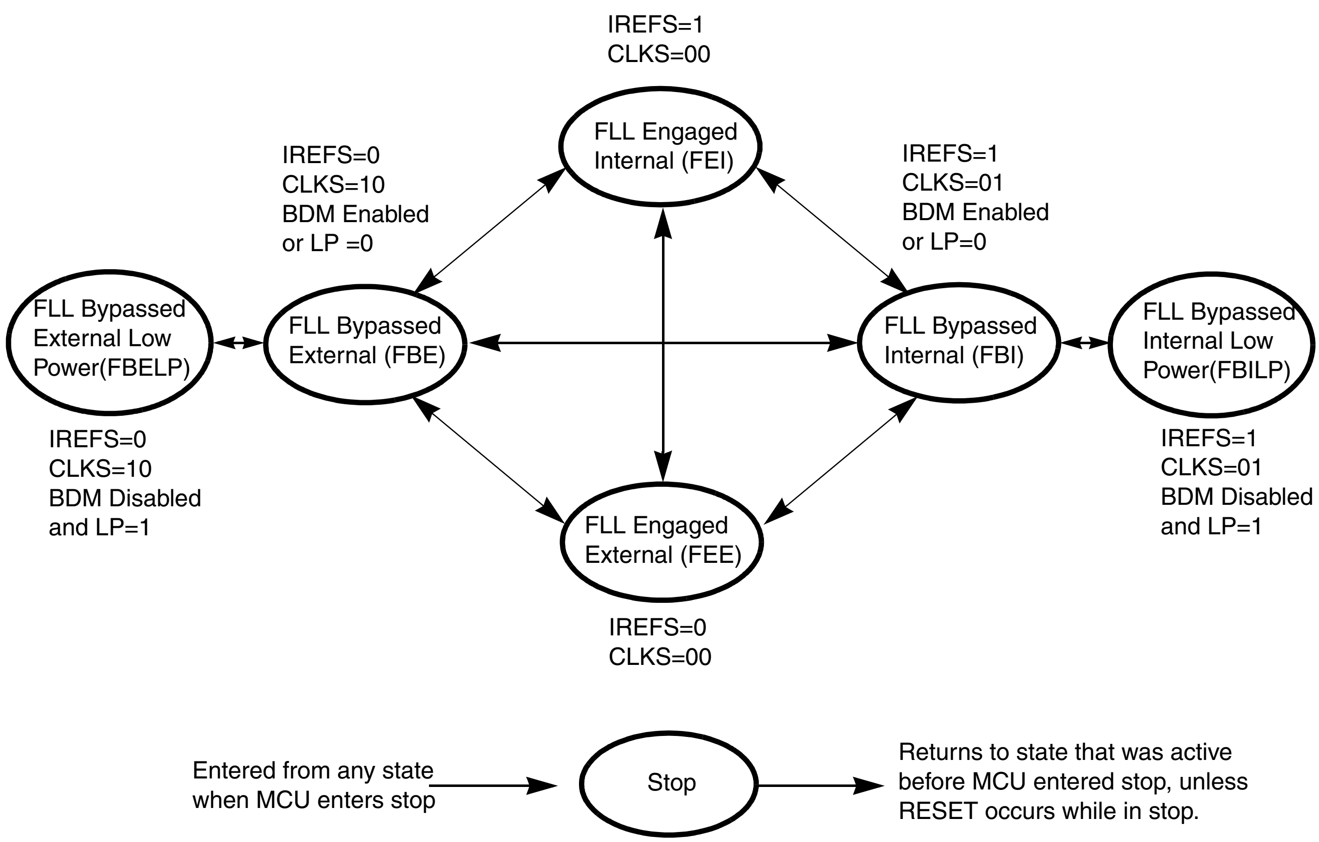


Figure 10-7. Clock Switching Modes

The seven states of the ICS are shown as a state diagram and are described below. The arrows indicate the allowed movements between the states.

10.4.1.1 FLL Engaged Internal (FEI)

FLL engaged internal (FEI) is the default mode of operation and is entered when all the following conditions occur:

11.4 Functional Description

This section provides a complete functional description of the IIC module.

11.4.1 IIC Protocol

The IIC bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. All devices connected to it must have open drain or open collector outputs. A logic AND function is exercised on both lines with external pullup resistors. The value of these resistors is system dependent.

Normally, a standard communication is composed of four parts:

- Start signal
- Slave address transmission
- Data transfer
- Stop signal

The stop signal should not be confused with the CPU stop instruction. The IIC bus system communication is described briefly in the following sections and illustrated in Figure 11-9.

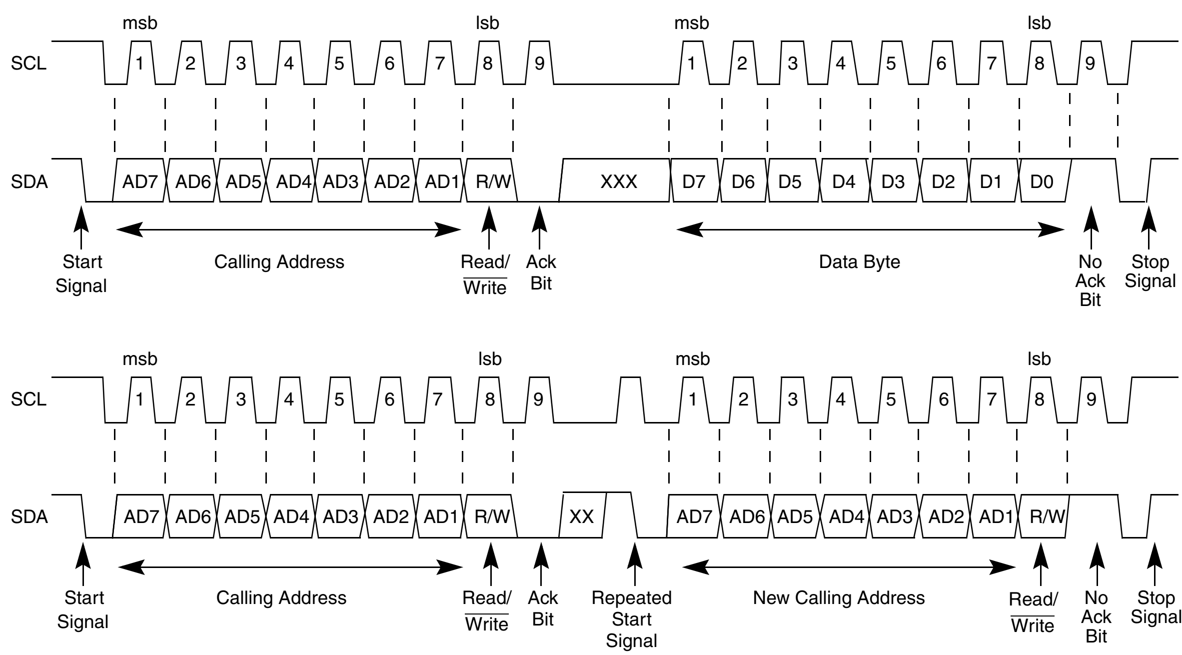


Figure 11-9. IIC Bus Transmission Signals

11.4.1.1 Start Signal

When the bus is free, no master device is engaging the bus (SCL and SDA lines are at logical high), a master may initiate communication by sending a start signal. As shown in Figure 11-9, a start signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and brings all slaves out of their idle states.

12.4.1 MTIM Operation Example

This section shows an example of the MTIM operation as the counter reaches a matching value from the modulo register.

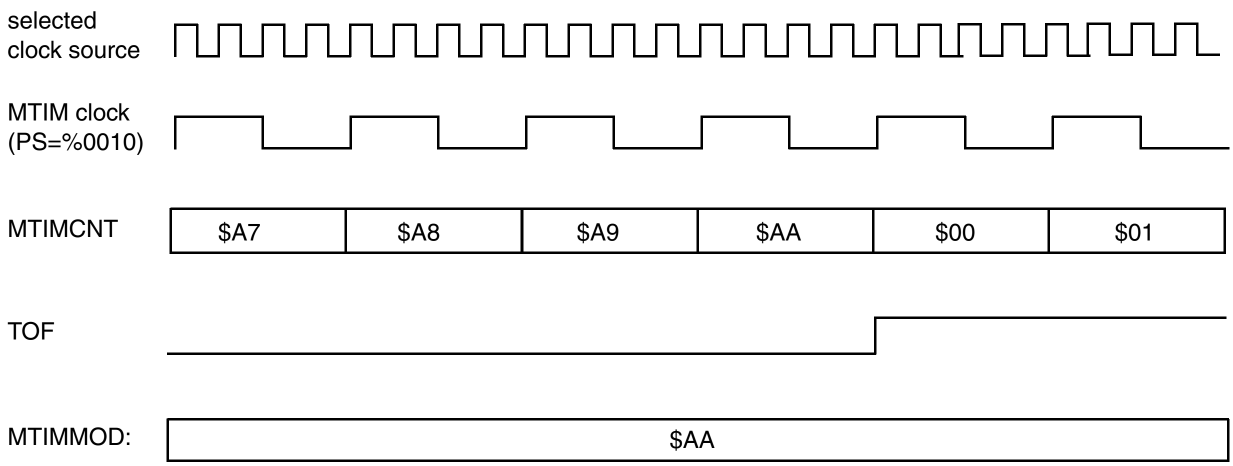


Figure 12-8. MTIM counter overflow example

In the example of Figure 12-8, the selected clock source could be any of the five possible choices. The prescaler is set to PS = %0010 or divide-by-4. The modulo value in the MTIMMOD register is set to $AA. When the counter, MTIMCNT, reaches the modulo value of $AA, the counter overflows to $00 and continues counting. The timer overflow flag, TOF, sets when the counter value changes from $AA to $00. An MTIM overflow interrupt is generated when TOF is set, if TOIE = 1.

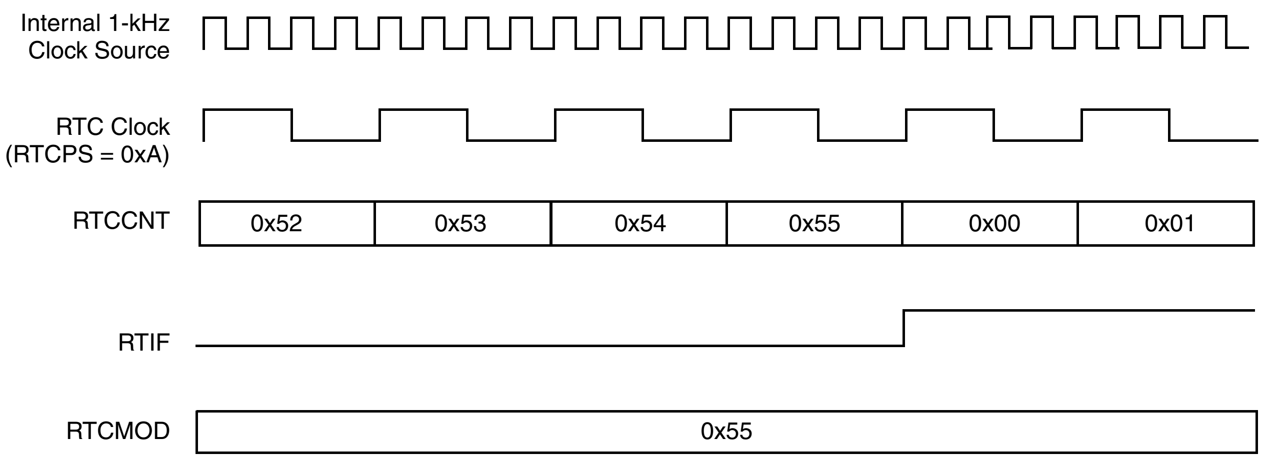


Figure 13-6. RTC Counter Overflow Example

In the example of Figure 13-6, the selected clock source is the 1-kHz internal oscillator clock source. The prescaler (RTCPS) is set to 0xA or divide-by-4. The modulo value in the RTCMOD register is set to 0x55. When the counter, RTCCNT, reaches the modulo value of 0x55, the counter overflows to 0x00 and continues counting. The real-time interrupt flag, RTIF, sets when the counter value changes from 0x55 to 0x00. A real-time interrupt is generated when RTIF is set, if RTIE is set.

13.5 Initialization/Application Information

This section provides example code to give some basic direction to a user on how to initialize and configure the RTC module. The example software is implemented in C language.

The example below shows how to implement time of day with the RTC using the 1-kHz clock source to achieve the lowest possible power consumption. Because the 1-kHz clock source is not as accurate as a crystal, software can be added for any adjustments. For accuracy without adjustments at the expense of additional power consumption, the external clock (ERCLK) or the internal clock (IRCLK) can be selected with appropriate prescaler and modulo values.

```
/* Initialize the elapsed time counters */
Seconds = 0;
Minutes = 0;
Hours = 0;
Days=0;

/* Configure RTC to interrupt every 1 second from 1-kHz clock source */
RTCMOD.byte = 0x00;
RTCSC.byte = 0x1F;

/**********************************************************************
Function Name : RTC_ISR
Notes : Interrupt service routine for RTC module.
**********************************************************************/
```

Chapter 15
Serial Peripheral Interface (S08SPIV3)

15.1 Introduction

The serial peripheral interface (SPI) module provides for full-duplex, synchronous, serial communication between the MCU and peripheral devices. These peripheral devices can include other microcontrollers, analog-to-digital converters, shift registers, sensors, memories, etc.

The SPI runs at a baud rate up to the bus clock divided by two. Software can poll the status flags, or SPI operation can be interrupt driven.

Figure 15-1 shows the MC9S08SG8 block diagram with the SPI module highlighted.

16.1.5 Features

The TPM includes these distinctive features:

- One to eight channels:
  - Each channel may be input capture, output compare, or edge-aligned PWM
  - Rising-Edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
  - Selectable polarity on PWM outputs
- Module may be configured for buffered, center-aligned pulse-width-modulation (CPWM) on all channels
- Timer clock source selectable as prescaled bus clock, fixed system clock, or an external clock pin
  - Prescale taps for divide-by 1, 2, 4, 8, 16, 32, 64, or 128
  - Fixed system clock source are synchronized to the bus clock by an on-chip synchronization circuit
  - External clock pin may be shared with any timer channel pin or a separated input pin
- 16-bit free-running or modulo up/down count operation
- Timer system enable
- One interrupt per channel plus terminal count interrupt

16.1.6 Modes of Operation

In general, TPM channels may be independently configured to operate in input capture, output compare, or edge-aligned PWM modes. A control bit allows the whole TPM (all channels) to switch to center-aligned PWM mode. When center-aligned PWM mode is selected, input capture, output compare, and edge-aligned PWM functions are not available on any channels of this TPM module.

When the microcontroller is in active BDM background or BDM foreground mode, the TPM temporarily suspends all counting until the microcontroller returns to normal user operating mode. During stop mode, all system clocks, including the main oscillator, are stopped; therefore, the TPM is effectively disabled until clocks resume. During wait mode, the TPM continues to operate normally. Provided the TPM does not need to produce a real time reference or provide the interrupt source(s) needed to wake the MCU from wait mode, the user can save power by disabling TPM functions before entering wait mode.

- Input capture mode

  When a selected edge event occurs on the associated MCU pin, the current value of the 16-bit timer counter is captured into the channel value register and an interrupt flag bit is set. Rising edges, falling edges, any edge, or no edge (disable channel) may be selected as the active edge which triggers the input capture.
- Output compare mode

  When the value in the timer counter register matches the channel value register, an interrupt flag bit is set, and a selected output action is forced on the associated MCU pin. The output compare action may be selected to force the pin to zero, force the pin to one, toggle the pin, or ignore the pin (used for software timing functions).

17.2.3 BDC Commands

BDC commands are sent serially from a host computer to the BKGD pin of the target HCS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. Active background mode commands require that the target MCU is currently in the active background mode while non-intrusive commands may be issued at any time whether the target MCU is in active background mode or running a user application program.

Table 17-1 shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

Coding Structure Nomenclature

This nomenclature is used in Table 17-1 to describe the coding structure of the BDC commands.

		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.5 ESD Protection and Latch-Up Immunity

Although damage from electrostatic discharge (ESD) is much less common on these devices than on early CMOS circuits, normal handling precautions should be used to avoid exposure to static discharge. Qualification tests are performed to ensure that these devices can withstand exposure to reasonable levels of static without suffering any permanent damage.

All ESD testing is in conformity with AEC-Q100 Stress Test Qualification for Automotive Grade Integrated Circuits. During the device qualification ESD stresses were performed for the human body model (HBM) and the charge device model (CDM).

A device is defined as a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

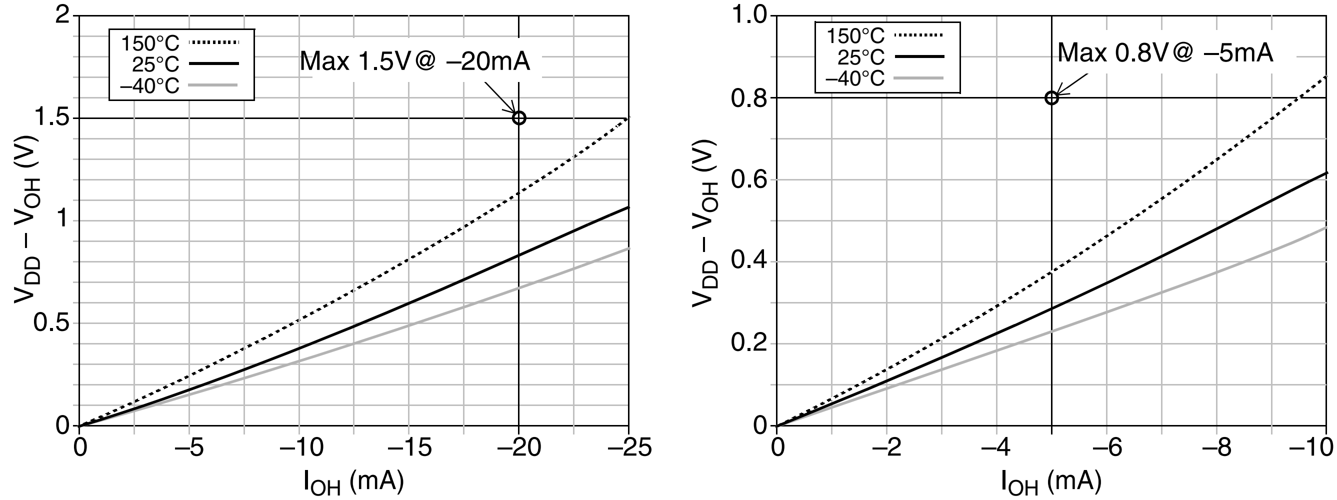
Table A-4. ESD and Latch-up Test Conditions

Model	Description	Symbol	Value	Unit
Human Body	Series resistance	R1	1500	Ω
	Storage capacitance	C	100	pF
	Number of pulses per pin	—	3	
Latch-up	Minimum input voltage limit		– 2.5	V
	Maximum input voltage limit		7.5	V

Table A-5. ESD and Latch-Up Protection Characteristics

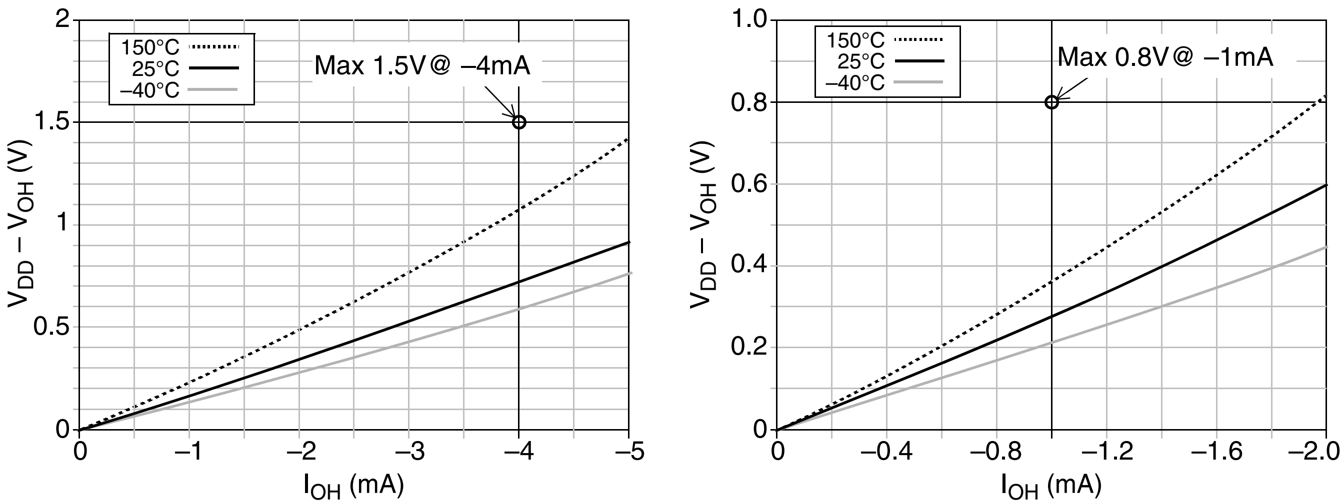
No.	Rating[1]	Symbol	Min	Max	Unit
1	Human body model (HBM)	V_{HBM}	± 2000	—	V
2	Charge device model (CDM)	V_{CDM}	± 500	—	V
3	Latch-up current at T_A = 125°C	I_{LAT}	± 100	—	mA

1 Parameter is achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted.



a) V_{DD} = 5V, High Drive   b) V_{DD} = 3V, High Drive

Figure A-3. Typical $V_{DD} – V_{OH}$ vs I_{OH}, High Drive Strength



a) V_{DD} = 5V, Low Drive   b) V_{DD} = 3V, Low Drive

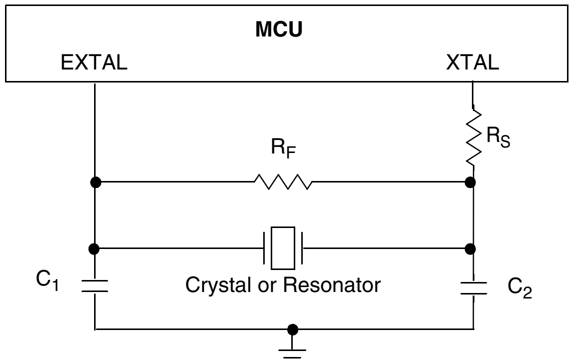
Figure A-4. Typical $V_{DD} – V_{OH}$ vs I_{OH}, Low Drive Strength

[2] Electrical characteristics only apply to the temperature rated devices marked with x.

[3] The input clock source must be divided using RDIV to within the range of 31.25 kHz to 39.0625 kHz.

[4] This parameter is characterized and not tested on each device. Proper PC board layout procedures must be followed to achieve specifications.

[5] 4 MHz crystal



Appendix B
Ordering Information and Mechanical Drawings

B.1 Ordering Information

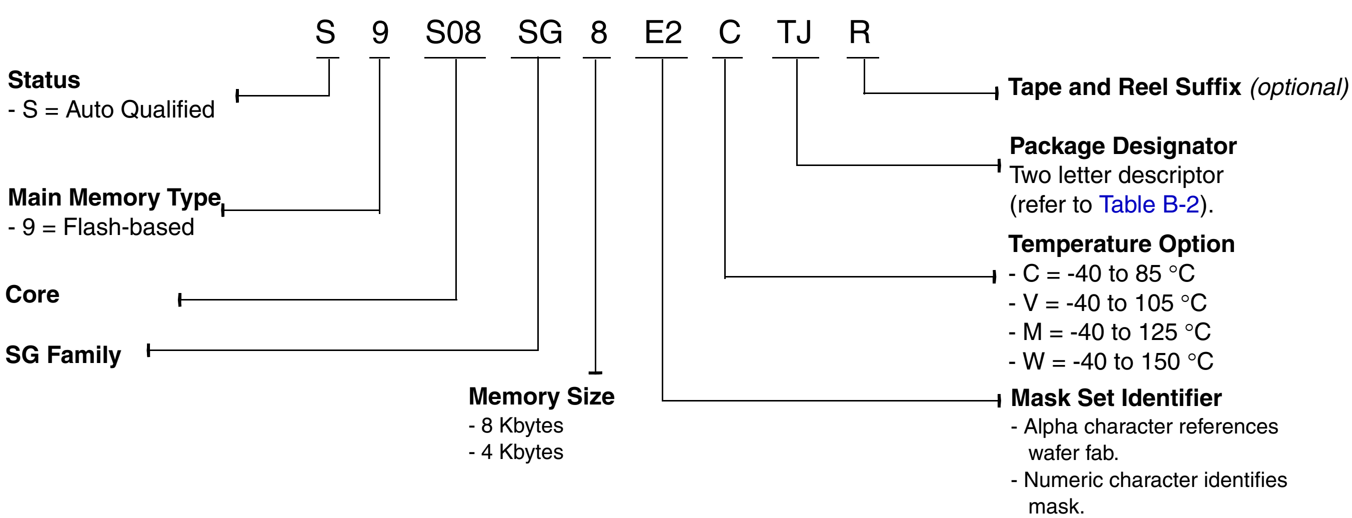
This section contains ordering information for MC9S08SG8 and MC9S08SG4 devices.

Table B-1. Device Numbering System

Device Number[1]	Memory		Temp Rated[2]		Available Packages[3]		
	FLASH	RAM	Standard	AEC Grade 0	20-Pin	16-Pin	8-Pin
S9S08SG8	8 K	512	x		20 TSSOP	16 TSSOP	8 NB SOIC
S9S08SG4	4 K	256	x				
S9S08SG8	8 K	512		x	—	16 TSSOP	—
S9S08SG4	4 K	256		x			

[1] See Table 1-1 for a complete description of modules included on each device.
[2] Apply to the temperature rated devices marked with x only.
[3] See Table B-2 for package information.

B.1.1 Device Numbering Scheme



How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

MC9S08SG8
Rev. 8
1/2014