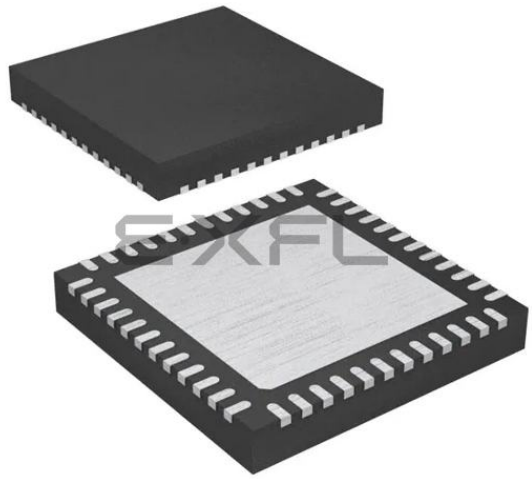

Welcome to E-XFL.COM

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	39
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	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gt16acfder

Part Number	Package Description	Original (gold wire) package document number	Current (copper wire) package document number
MC68HC908JW32	48 QFN	98ARH99048A	98ASA00466D
MC9S08AC16			
MC9S908AC60			
MC9S08AC128			
MC9S08AW60			
MC9S08GB60A			
MC9S08GT16A			
MC9S08JM16			
MC9S08JM60			
MC9S08LL16			
MC9S08QE128			
MC9S08QE32			
MC9S08RG60			
MCF51CN128			
MC9RS08LA8	48 QFN	98ARL10606D	98ASA00466D
MC9S08GT16A	32 QFN	98ARH99035A	98ASA00473D
MC9S908QE32	32 QFN	98ARE10566D	98ASA00473D
MC9S908QE8	32 QFN	98ASA00071D	98ASA00736D
MC9S08JS16	24 QFN	98ARL10608D	98ASA00734D
MC9S08QB8			
MC9S08QG8	24 QFN	98ARL10605D	98ASA00474D
MC9S08SH8	24 QFN	98ARE10714D	98ASA00474D
MC9RS08KB12	24 QFN	98ASA00087D	98ASA00602D
MC9S08QG8	16 QFN	98ARE10614D	98ASA00671D
MC9RS08KB12	8 DFN	98ARL10557D	98ASA00672D
MC9S08QG8			
MC9RS08KA2	6 DFN	98ARL10602D	98ASA00735D

Chapter 6
Parallel Input/Output

Chapter 7
Keyboard Interrupt (S08KBIV1)

makes the MCU secure. During development, whenever the FLASH is erased, it is good practice to immediately program the SEC00 bit to 0 in NVOPT so SEC01:SEC00 = 1:0. This would allow the MCU to remain unsecured after a subsequent reset.

The on-chip debug module cannot be enabled while the MCU is secure. The separate background debug controller can still be used for background memory access commands, but the MCU cannot enter active background mode except by holding BKGD/MS low at the rising edge of reset.

A user can choose to allow or disallow a security unlocking mechanism through an 8-byte backdoor security key. If the nonvolatile KEYEN bit in NVOPT/FOPT is 0, the backdoor key is disabled and there is no way to disengage security without completely erasing all FLASH locations. If KEYEN is 1, a secure user program can temporarily disengage security by:

1. Writing 1 to KEYACC in the FCNFG register. This makes the FLASH module interpret writes to the backdoor comparison key locations (NVBACKKEY through NVBACKKEY+7) as values to be compared against the key rather than as the first step in a FLASH program or erase command.
2. Writing the user-entered key values to the NVBACKKEY through NVBACKKEY+7 locations. These writes must be done in order, starting with the value for NVBACKKEY and ending with NVBACKKEY+7. STHX should not be used for these writes because these writes cannot be done on adjacent bus cycles. User software normally would get the key codes from outside the MCU system through a communication interface such as a serial I/O.
3. Writing 0 to KEYACC in the FCNFG register. If the 8-byte key that was just written matches the key stored in the FLASH locations, SEC01:SEC00 are automatically changed to 1:0 and security will be disengaged until the next reset.

The security key can be written only from RAM, so it cannot be entered through background commands without the cooperation of a secure user program.

The backdoor comparison key (NVBACKKEY through NVBACKKEY+7) is located in FLASH memory locations in the nonvolatile register space so users can program these locations exactly as they would program any other FLASH memory location. The nonvolatile registers are in the same 512-byte block of FLASH as the reset and interrupt vectors, so block protecting that space also block protects the backdoor comparison key. Block protects cannot be changed from user application programs, so if the vector space is block protected, the backdoor security key mechanism cannot permanently change the block protect, security settings, or the backdoor key.

Security can always be disengaged through the background debug interface by performing these steps:

1. Disable any block protections by writing FPROT. FPROT can be written only with background debug commands, not from application software.
2. Mass erase FLASH, if necessary.
3. Blank check FLASH. Provided FLASH is completely erased, security is disengaged until the next reset.

   To avoid returning to secure mode after the next reset, program NVOPT so SEC01:SEC00 = 1:0.

Table 4-8. FOPT Field Descriptions (continued)

Field	Description
6 FNORED	**Vector Redirection Disable** — When this bit is 1, vector redirection is disabled. 0 Vector redirection enabled. 1 Vector redirection disabled.
1:0 SEC0[1:0]	**Security State Code** — This 2-bit field determines the security state of the MCU as shown below. When the MCU is secure, the contents of RAM and FLASH memory cannot be accessed by instructions from any unsecured source including the background debug interface. For more detailed information about security, refer to Section 4.5, “Security.” 00 Secure 01 Secure 10 Unsecured 11 Secure SEC0[1:0] changes to 10 after successful backdoor key entry or a successful blank check of FLASH.

4.6.3 FLASH Configuration Register (FCNFG)

Bits 7 through 5 may be read or written at any time. Bits 4 through 0 always read 0 and cannot be written.

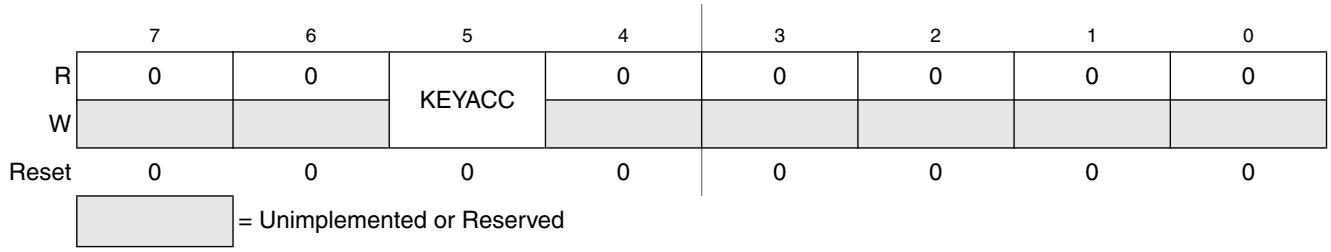

Figure 4-7. FLASH Configuration Register (FCNFG)

Table 4-9. FCNFG Field Descriptions

Field	Description
5 KEYACC	**Enable Writing of Access Key** — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to Section 4.5, “Security.” 0 Writes to 0xFFB0–0xFFB7 are interpreted as the start of a FLASH programming or erase command. 1 Writes to NVBACKKEY (0xFFB0–0xFFB7) are interpreted as comparison key writes. Reads of the FLASH return invalid data.

4.6.4 FLASH Protection Register (FPROT and NVPROT)

During reset, the contents of the nonvolatile location NVPROT are copied from FLASH into FPROT. This register may be read at any time, but user program writes have no meaning or effect. Background debug commands can write to FPROT.

Chapter 6
Parallel Input/Output

6.1 Introduction

This section explains software controls related to parallel input/output (I/O). The MC9S08GT16A/GT8A has six I/O ports which include a total of up to 39 general-purpose I/O pins (one pin, PTG0, is output only). See Chapter 2, "Pins and Connections," for more information about the logic and hardware aspects of these pins.

Many of these pins are shared with on-chip peripherals such as timer systems, external interrupts, or keyboard interrupts. When these other modules are not controlling the port pins, they revert to general-purpose I/O control. For each I/O pin, a port data bit provides access to input (read) and output (write) data, a data direction bit controls the direction of the pin, and a pullup enable bit enables an internal pullup device (provided the pin is configured as an input), and a slew rate control bit controls the rise and fall times of the pins.

Pins that are not used in the application must be terminated. This prevents excess current caused by floating inputs and enhances immunity during noise or transient events. Termination methods include:

- Configuring unused pins as outputs driving high or low
- Configuring unused pins as inputs and using internal or external pullups

Never connect unused pins to V_{DD} or V_{SS}.

NOTE

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

6.1.1 Features

Parallel I/O features, depending on package choice, include:

- A total of 39 general-purpose I/O pins in six ports (PTG0 is output only)
- High-current drivers on port C pins
- Hysteresis input buffers
- Software-controlled pullups on each input pin
- Software-controlled slew rate output buffers
- Eight port A pins shared with KBI

Refer to Chapter 9, “Internal Clock Generator (S08ICGV4),” for more information about using these pins as oscillator pins.

6.3 Parallel I/O Controls

Provided no on-chip peripheral is controlling a port pin, the pins operate as general-purpose I/O pins that are accessed and controlled by a data register (PTxD), a data direction register (PTxDD), a pullup enable register (PTxPE), and a slew rate control register (PTxSE) where x is A, B, C, D, E, or G.

Reads of the data register return the pin value (if PTxDDn = 0) or the contents of the port data register (if PTxDDn = 1). Writes to the port data register are latched into the port register whether the pin is controlled by an on-chip peripheral or the pin is configured as an input. If the corresponding pin is not controlled by a peripheral and is configured as an output, this level will be driven out the port pin.

6.3.1 Data Direction Control

The data direction control bits determine whether the pin output driver is enabled, and they control what is read for port data register reads. Each port pin has a data direction control bit. When PTxDDn = 0, the corresponding pin is an input and reads of PTxD return the pin value. When PTxDDn = 1, the corresponding pin is an output and reads of PTxD return the last value written to the port data register. When a peripheral module or system function is in control of a port pin, the data direction control still controls what is returned for reads of the port data register, even though the peripheral system has overriding control of the actual pin direction.

For the MC9S08GT16A/GT8A MCU, reads of PTG0/BKGD/MS will return the value on the output pin.

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

6.3.2 Internal Pullup Control

An internal pullup device can be enabled for each port pin that is configured as an input (PTxDDn = 0). The pullup device is available for a peripheral module to use, provided the peripheral is enabled and is an input function as long as the PTxDDn = 0.

For the four configurable KBI module inputs on PTA7–PTA4, when a pin is configured to detect rising edges, the port pullup enable associated with the pin (PTAPEn) selects a pulldown rather than a pullup device.

6.3.3 Slew Rate Control

Slew rate control can be enabled for each port pin that is configured as an output (PTxDDn = 1) or if a peripheral module is enabled and its function is an output. Not all peripheral modules’ outputs have slew rate control; refer to Chapter 2, “Pins and Connections,” for more information about which pins have slew rate control.

6.5.4 Port D Registers (PTDD, PTDPE, PTSE, and PTDDD)

Port D includes five pins shared between general-purpose I/O, TPM1, and TPM2. Port D pins used as general-purpose I/O pins are controlled by the port D data (PTDD), data direction (PTDDD), pullup enable (PTDPE), and slew rate control (PTDSE) registers.

If a TPM takes control of a port D pin, the corresponding PTDDD bit is ignored. When the TPM is in output compare mode, the corresponding PTDSE can be used to provide slew rate on the pin. When the TPM is in input capture mode, the corresponding PTDPE can be used, provided the corresponding PTDDD bit is 0, to provide a pullup device on the pin.

Reads of PTDD will return the logic value of the corresponding pin, provided PTDDD is 0.

	7	6	5	4	3	2	1	0
R	0	0	0	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-20. Port D Data Register (PTDD)

Table 6-13. PTDD Field Descriptions

Field	Description
4:0 PTDD[4:0]	**Port D Data Register Bits** — For port D pins that are inputs, reads return the logic level on the pin. For port D pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port D pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTDD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled.

	7	6	5	4	3	2	1	0
R	0	0	0	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-21. Pullup Enable for Port D (PTDPE)

Table 6-14. PTDPE Field Descriptions

Field	Description
4:0 PTDPE[4:0]	**Pullup Enable for Port D Bits** — For port D pins that are inputs, these read/write control bits determine whether internal pullup devices are enabled. For port D pins that are configured as outputs, these bits are ignored and the internal pullup devices are disabled. 0 Internal pullup device disabled. 1 Internal pullup device enabled.

	7	6	5	4	3	2	1	0
R	0	0	0	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-22. Slew Rate Control Enable for Port D (PTDSE)

Table 6-15. PTDSE Field Descriptions

Field	Description
4:0 PTDSE[4:0]	**Slew Rate Control Enable for Port D Bits** — For port D pins that are outputs, these read/write control bits determine whether the slew rate controlled outputs are enabled. For port D pins that are configured as inputs, these bits are ignored. 0 Slew rate control disabled. 1 Slew rate control enabled.

	7	6	5	4	3	2	1	0
R	0	0	0	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-23. Data Direction for Port D (PTDDD)

Table 6-16. PTDDD Field Descriptions

Field	Description
4:0 PTDDD[4:0]	**Data Direction for Port D Bits** — These read/write bits control the direction of port D pins and what is read for PTDD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port D bit n and PTDD reads return the contents of PTDDn.

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTGSE3	PTGSE2	PTGSE1	PTGSE0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-30. Slew Rate Control Enable for Port G (PTGSE)

Table 6-23. PTGSE Field Descriptions

Field	Description
3:0 PTGSE[3:0]	**Slew Rate Control Enable for Port G Bits** — For port G pins that are outputs, these read/write control bits determine whether the slew rate controlled outputs are enabled. For port G pins that are configured as inputs, these bits are ignored. 0 Slew rate control disabled. 1 Slew rate control enabled.

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTGDD3	PTGDD2	PTGDD1	PTGDD0 Note (1)
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-31. Data Direction for Port G (PTGDD)

1 Although PTGDD0 is implemented, this bit actually has no effect on the operation of PTG0/BKGD.

Table 6-24. PTGDD Field Descriptions

Field	Description
3:0 PTGDD[3:0]	**Data Direction for Port G Bits** — These read/write bits control the direction of port G pins and what is read for PTGD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port G bit n and PTGD reads return the contents of PTGDn.

Chapter 7
Keyboard Interrupt (S08KBIV1)

7.1 Introduction

The MC9S08GT16A/GT8A has one KBI module with eight keyboard interrupt inputs that share port A pins. See Chapter 2, “Pins and Connections,” for more information about the logic and hardware aspects of these pins.

7.1.1 Port A and Keyboard Interrupt Pins

MCU Pin:	PTA7/ KBIP7	PTA6/ KBIP6	PTA5/ KBIP5	PTA4/ KBIP4	PTA3/ KBIP3	PTA2/ KBIP2	PTA1/ KBIP1	PTA0/ KBIP0

Figure 7-1. Port A Pin Names

The following paragraphs discuss controlling the keyboard interrupt pins.

Port A is an 8-bit port which is shared among the KBI keyboard interrupt inputs and general-purpose I/O. The eight KBIPEn control bits in the KBIPE register allow selection of any combination of port A pins to be assigned as KBI inputs. Any pins which are enabled as KBI inputs will be forced to act as inputs and the remaining port A pins are available as general-purpose I/O pins controlled by the port A data (PTAD), data direction (PTADD), and pullup enable (PTAPE) registers.

KBI inputs can be configured for edge-only sensitivity or edge-and-level sensitivity. Bits 3 through 0 of port A are falling-edge/low-level sensitive while bits 7 through 4 can be configured for rising-edge/high-level or for falling-edge/low-level sensitivity.

The eight PTAPEn control bits in the PTAPE register allow you to select whether an internal pullup device is enabled on each port A pin that is configured as an input. When any of bits 7 through 4 of port A are enabled as KBI inputs and are configured to detect rising edges/high levels, the pullup enable bits enable pulldown rather than pullup devices.

An enabled keyboard interrupt can be used to wake the MCU from wait or standby (stop3).

7.1.2 Features

The keyboard interrupt (KBI) module features include:

- Keyboard interrupts selectable on eight port pins:
  - Four falling-edge/low-level sensitive
  - Four falling-edge/low-level or rising-edge/high-level sensitive
  - Choice of edge-only or edge-and-level sensitivity
  - Common interrupt flag and interrupt enable control
  - Capable of waking up the MCU from stop3 or wait mode

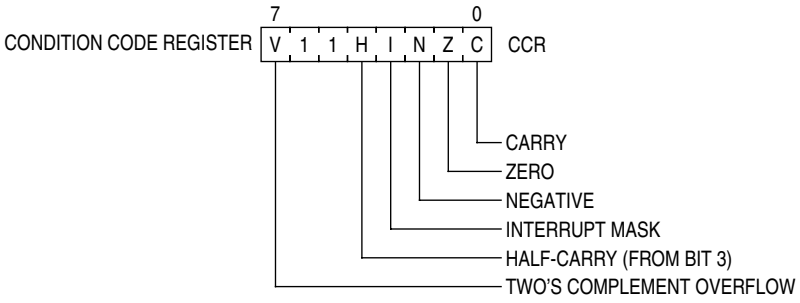

Figure 8-2. Condition Code Register

Table 8-1. CCR Register Field Descriptions

Field	Description
7 V	**Two's Complement Overflow Flag** — The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag. 0 No overflow 1 Overflow
4 H	**Half-Carry Flag** — The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C condition code bits to automatically add a correction value to the result from a previous ADD or ADC on BCD operands to correct the result to a valid BCD value. 0 No carry between bits 3 and 4 1 Carry between bits 3 and 4
3 I	**Interrupt Mask Bit** — When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the first instruction of the interrupt service routine is executed. Interrupts are not recognized at the instruction boundary after any instruction that clears I (CLI or TAP). This ensures that the next instruction after a CLI or TAP will always be executed without the possibility of an intervening interrupt, provided I was set. 0 Interrupts enabled 1 Interrupts disabled
2 N	**Negative Flag** — The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result. Simply loading or storing an 8-bit or 16-bit value causes N to be set if the most significant bit of the loaded or stored value was 1. 0 Non-negative result 1 Negative result
1 Z	**Zero Flag** — The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of 0x00 or 0x0000. Simply loading or storing an 8-bit or 16-bit value causes Z to be set if the loaded or stored value was all 0s. 0 Non-zero result 1 Zero result
0 C	**Carry/Borrow Flag** — The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag. 0 No carry out of bit 7 1 Carry out of bit 7

9.4.1 Off Mode (Off)

Normally when the CPU enters stop mode, the ICG will cease all clock activity and is in the off state. However there are two cases to consider when clock activity continues while the CPU is in stop mode,

9.4.1.1 BDM Active

When the BDM is enabled, the ICG continues activity as originally programmed. This allows access to memory and control registers via the BDC controller.

9.4.1.2 OSCSTEN Bit Set

When the oscillator is enabled in stop mode (OSCSTEN = 1), the individual clock generators are enabled but the clock feed to the rest of the MCU is turned off. This option is provided to avoid long oscillator startup times if necessary, or to run the RTI from the oscillator during stop3.

9.4.1.3 Stop/Off Mode Recovery

Upon the CPU exiting stop mode due to an interrupt, the previously set control bits are valid and the system clock feed resumes. If FEE is selected, the ICG will source the internal reference until the external clock is stable. If FBE is selected, the ICG will wait for the external clock to stabilize before enabling ICGOUT.

Upon the CPU exiting stop mode due to a reset, the previously set ICG control bits are ignored and the default reset values applied. Therefore the ICG will exit stop in SCM mode configured for an approximately 8 MHz DCO output (4 MHz bus clock) with trim value maintained. If using a crystal, 4096 clocks are detected prior to engaging ICGERCLK. This is incorporated in crystal start-up time.

9.4.2 Self-Clocked Mode (SCM)

Self-clocked mode (SCM) is the default mode of operation and is entered when any of the following conditions occur:

- After any reset.
- Exiting from off mode when CLKS does not equal 10. If CLKS = X1, the ICG enters this state temporarily until the DCO is stable (DCOS = 1).
- CLKS bits are written from X1 to 00.
- CLKS = 1X and ICGERCLK is not detected (both ERCS = 0 and LOCS = 1).

In this state, the FLL loop is open. The DCO is on, and the output clock signal ICGOUT frequency is given by $f_{ICGDCLK}$ / R. The ICGDCLK frequency can be varied from 8 MHz to 40 MHz by writing a new value into the filter registers (ICGFLTH and ICGFLTL). This is the only mode in which the filter registers can be written.

If this mode is entered due to a reset, $f_{ICGDCLK}$ will default to f_{Self_reset} which is nominally 8 MHz. If this mode is entered from FLL engaged internal, $f_{ICGDCLK}$ will maintain the previous frequency.If this mode is entered from FLL engaged external (either by programming CLKS or due to a loss of external reference clock), $f_{ICGDCLK}$ will maintain the previous frequency, but ICGOUT will double if the FLL was unlocked. If this mode is entered from off mode, $f_{ICGDCLK}$ will be equal to the frequency of ICGDCLK before

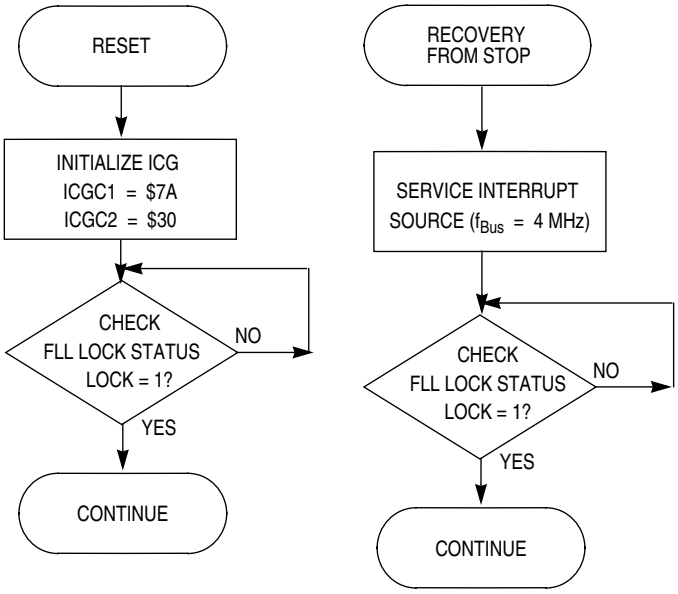

Figure 9-15. ICG Initialization and Stop Recovery for Example #2

10.1.2 Features

The TPM has the following features:

- Each TPM may be configured for buffered, center-aligned pulse-width modulation (CPWM) on all channels
- Clock sources independently selectable per TPM (multiple TPMs device)
- Selectable clock sources (device dependent): bus clock, fixed system clock, external pin
- Clock prescaler taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit free-running or up/down (CPWM) count operation
- 16-bit modulus register to control counter range
- Timer system enable
- One interrupt per channel plus a terminal count interrupt for each TPM module (multiple TPMs device)
- Channel features:
  - Each channel may be input capture, output compare, or buffered 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

10.1.3 Block Diagram

Figure 10-2 shows the structure of a TPM. Some MCUs include more than one TPM, with various numbers of channels.

Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

Some MCU systems have more than one TPM, so register names include placeholder characters to identify which TPM and which channel is being referenced. For example, TPMxCnSC refers to timer (TPM) x, channel n and TPM1C2SC is the status and control register for timer 1, channel 2.

10.3.1 Timer x Status and Control Register (TPMxSC)

TPMxSC contains the overflow status flag and control bits that are used to configure the interrupt enable, TPM configuration, clock source, and prescale divisor. These controls relate to all channels within this timer module.

	7	6	5	4	3	2	1	0
R	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-3. Timer x Status and Control Register (TPMxSC)

Table 10-1. TPMxSC Register Field Descriptions

Field	Description
7 TOF	**Timer Overflow Flag** — This flag is set when the TPM counter changes to 0x0000 after reaching the modulo value programmed in the TPM counter modulo registers. When the TPM is configured for CPWM, TOF is set after the counter has reached the value in the modulo register, at the transition to the next lower count value. Clear TOF by reading the TPM status and control register when TOF is set and then writing a 0 to TOF. If another TPM overflow occurs before the clearing sequence is complete, the sequence is reset so TOF would remain set after the clear sequence was completed for the earlier TOF. Reset clears TOF. Writing a 1 to TOF has no effect. 0 TPM counter has not reached modulo value or overflow 1 TPM counter has overflowed
6 TOIE	**Timer Overflow Interrupt Enable** — This read/write bit enables TPM overflow interrupts. If TOIE is set, an interrupt is generated when TOF equals 1. Reset clears TOIE. 0 TOF interrupts inhibited (use software polling) 1 TOF interrupts enabled
5 CPWMS	**Center-Aligned PWM Select** — This read/write bit selects CPWM operating mode. Reset clears this bit so the TPM operates in up-counting mode for input capture, output compare, and edge-aligned PWM functions. Setting CPWMS reconfigures the TPM to operate in up-/down-counting mode for CPWM functions. Reset clears CPWMS. 0 All TPMx channels operate as input capture, output compare, or edge-aligned PWM mode as selected by the MSnB:MSnA control bits in each channel's status and control register 1 All TPMx channels operate in center-aligned PWM mode
4:3 CLKS[B:A]	**Clock Source Select** — As shown in Table 10-2, this 2-bit field is used to disable the TPM system or select one of three clock sources to drive the counter prescaler. The external source and the XCLK are synchronized to the bus clock by an on-chip synchronization circuit.
2:0 PS[2:0]	**Prescale Divisor Select** — This 3-bit field selects one of eight divisors for the TPM clock input as shown in Table 10-3. This prescaler is located after any clock source synchronization or clock source selection, so it affects whatever clock source is selected to drive the TPM system.

pin from a master and the MISO waveform applies to the MISO output from a slave. The $\overline{SS}$ OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master $\overline{SS}$ output goes to active low one-half SPSCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The $\overline{SS}$ IN waveform applies to the slave select input of a slave.

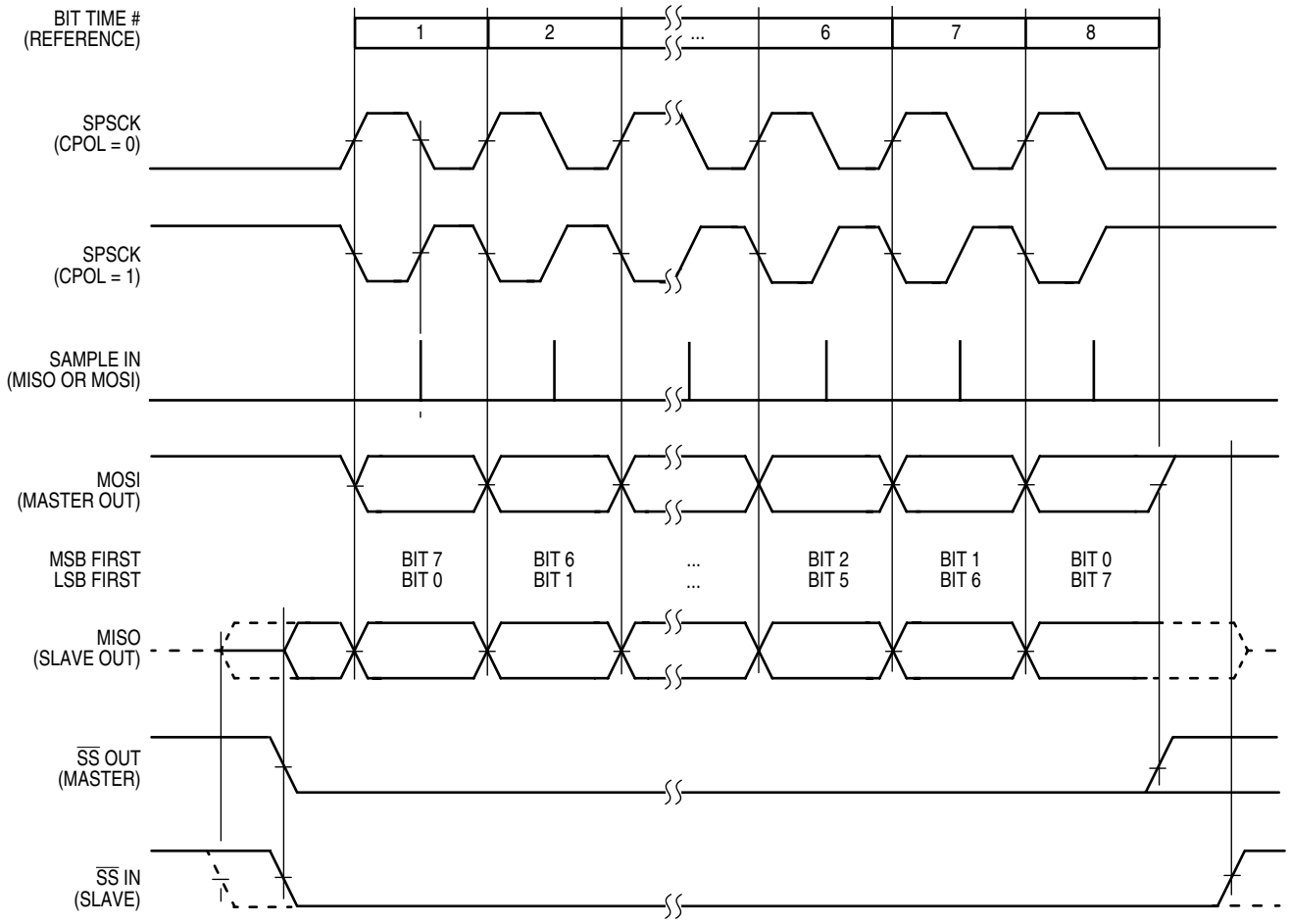

Figure 12-11. SPI Clock Formats (CPHA = 1)

When CPHA = 1, the slave begins to drive its MISO output when $\overline{SS}$ goes to active low, but the data is not defined until the first SPSCK edge. The first SPSCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the third SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled, and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CHPA = 1, the slave's $\overline{SS}$ input is not required to go to its inactive high level between transfers.

Figure 12-12 shows the clock formats when CPHA = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected ($\overline{SS}$ IN goes low), and bit 8 ends at the last SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting

14.3.2 ATD Status and Control (ATDSC)

Writes to the ATD status and control register clears the CCF flag, cancels any pending interrupts, and initiates a new conversion.

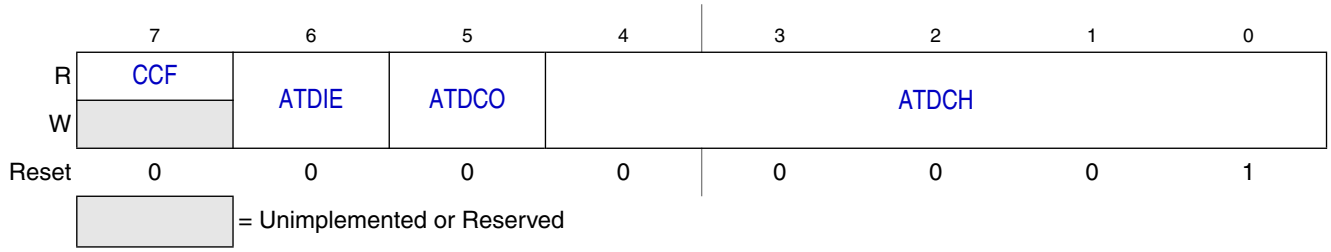

Figure 14-6. ATD Status and Control Register (ATDSC)

Table 14-5. ATDSC Register Field Descriptions

Field	Description
7 CCF	**Conversion Complete Flag** — The CCF is a read-only bit which is set each time a conversion is complete. The CCF bit is cleared whenever the ATDSC register is written. It is also cleared whenever the result registers, ATDRH or ATDRL, are read. 0 Current conversion is not complete. 1 Current conversion is complete.
6 ATDIE	**ATD Interrupt Enabled** — When this bit is set, an interrupt is generated upon completion of an ATD conversion. At this time, the result registers contain the result data generated by the conversion. The interrupt will remain pending as long as the conversion complete flag CCF is set. If the ATDIE bit is cleared, then the CCF bit must be polled to determine when the conversion is complete. Note that system reset clears pending interrupts. 0 ATD interrupt disabled. 1 ATD interrupt enabled.
5 ATDCO	**ATD Continuous Conversion** — When this bit is set, the ATD will convert samples continuously and update the result registers at the end of each conversion. When this bit is cleared, only one conversion is completed between writes to the ATDSC register. 0 Single conversion mode. 1 Continuous conversion mode.
4:0 ATDCH	**Analog Input Channel Select** — This field of bits selects the analog input channel whose signal is sampled and converted to digital codes. Table 14-6 lists the coding used to select the various analog input channels.

Table 14-6. Analog Input Channel Select Coding

ATDCH	Analog Input Channel
00	AD0
01	AD1
02	AD2
03	AD3
04	AD4
05	AD5
06	AD6
07	AD7
08–1D	Reserved (default to V_{REFL})
1E	V_{REFH}
1F	V_{REFL}

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD pin so it reverts to high impedance
- Monitors the BKGD pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

15.2.4 BDC Hardware Breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter active background mode at the first instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU will enter active background mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits. The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more flexible than the simple breakpoint in the BDC module.

A.5 Electrostatic Discharge (ESD) Protection Characteristics

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), the Machine Model (MM) 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 Pulse per pin	—	3	
Machine	Series Resistance	R1	0	Ω
	Storage Capacitance	C	200	pF
	Number of Pulse per pin	—	3	
Charge Device Model	Series Resistance	R1		Ω
	Storage Capacitance	C		pF
	Number of Pulse per pin	—		
Latch-Up	Minimum input voltage limit		–2.5	V
	Maximum input voltage limit		7.5	V

A.6 DC Characteristics

This section includes information about power supply requirements, I/O pin characteristics, and power supply current in various operating modes.

Table A-5. MCU Operating Conditions

Characteristic	Min	Typ	Max	Unit
Supply Voltage	1.8	—	3.6	V
Temperature M C	 –40 –40	 — —	 125 85	°C

Table A-9. ATD Timing/Performance Characteristics[1] (continued)

No.	Characteristic	Condition	Symbol	Min	Typ	Max	Unit
7	Ideal resolution (1 LSB)[5]	2.08V ≤ V_{DDAD} ≤ 3.6V	RES	2.031	—	3.516	mV
		1.80V ≤ V_{DDAD} < 2.08V		1.758	—	2.031	
8	Differential non-linearity[6]	1.80V ≤ V_{DDAD} ≤ 3.6V	DNL	—	±0.5	±1.0	LSB
9	Integral non-linearity[7]	1.80 V ≤ V_{DDAD} ≤ 3.6V	INL	—	±0.5	±1.0	LSB
10	Zero-scale error[8]	1.80V ≤ V_{DDAD} ≤ 3.6V	E_{ZS}	—	±0.4	±1.0	LSB
11	Full-scale error[9]	1.80V ≤ V_{DDAD} ≤ 3.6V	E_{FS}	—	±0.4	±1.0	LSB
12	Input leakage error [10]	1.80V ≤ V_{DDAD} ≤ 3.6V	E_{IL}	—	±0.05	±5	LSB
13	Total unadjusted error[11]	1.80V ≤ V_{DDAD} ≤ 3.6V	E_{TU}	—	±1.1	±2.5	LSB
14	Input resistance		R_{AIN}	—	5	7	kΩ
15	Input capacitance		C_{AIN}	—	—	25	pF

1 All ACCURACY numbers are based on processor and system being in WAIT state (very little activity and no IO switching) and that adequate low-pass filtering is present on analog input pins (filter with 0.01 µF to 0.1 µF capacitor between analog input and V_{REFL}). Failure to observe these guidelines may result in system or microcontroller noise causing accuracy errors which will vary based on board layout and the type and magnitude of the activity.

2 This is the conversion time for subsequent conversions in continuous convert mode. Actual conversion time for single conversions or the first conversion in continuous mode is extended by one ATD clock cycle and 2 bus cycles due to starting the conversion and setting the CCF flag. The total conversion time in Bus Cycles for a conversion is:

SC Bus Cycles = ((PRS+1)*2) * (28+1) + 2 CC Bus Cycles = ((PRS+1)*2) * (28)

3 R_{AS} is the real portion of the impedance of the network driving the analog input pin. Values greater than this amount may not fully charge the input circuitry of the ATD resulting in accuracy error.

4 Analog input must be between V_{REFL} and V_{REFH} for valid conversion. Values greater than V_{REFH} will convert to $3FF less the full scale error (E_{FS}).

5 The resolution is the ideal step size or 1LSB = (V_{REFH}–V_{REFL})/1024

6 Differential non-linearity is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to and from the current code.

7 Integral non-linearity is the difference between the transition voltage to the current code and the adjusted ideal transition voltage for the current code. The adjusted ideal transition voltage is (Current Code–1/2)*(1/((V_{REFH}+E_{FS})–(V_{REFL}+E_{ZS}))).

8 Zero-scale error is the difference between the transition to the first valid code and the ideal transition to that code. The Ideal transition voltage to a given code is (Code–1/2)*(1/(V_{REFH}–V_{REFL})).

9 Full-scale error is the difference between the transition to the last valid code and the ideal transition to that code. The ideal transition voltage to a given code is (Code–1/2)*(1/(V_{REFH}–V_{REFL})).

10 Input leakage error is error due to input leakage across the real portion of the impedance of the network driving the analog pin. Reducing the impedance of the network reduces this error.

11 Total unadjusted error is the difference between the transition voltage to the current code and the ideal straight-line transfer function. This measure of error includes inherent quantization error (1/2LSB) and circuit error (differential, integral, zero-scale, and full-scale) error. The specified value of E_T assumes zero E_{IL} (no leakage or zero real source impedance).