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	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	56
Program Memory Size	32KB (32K 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	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gb32cfuer

Chapter 8
Central Processor Unit (CPU)

Chapter 9
Keyboard Interrupt (KBI) Module

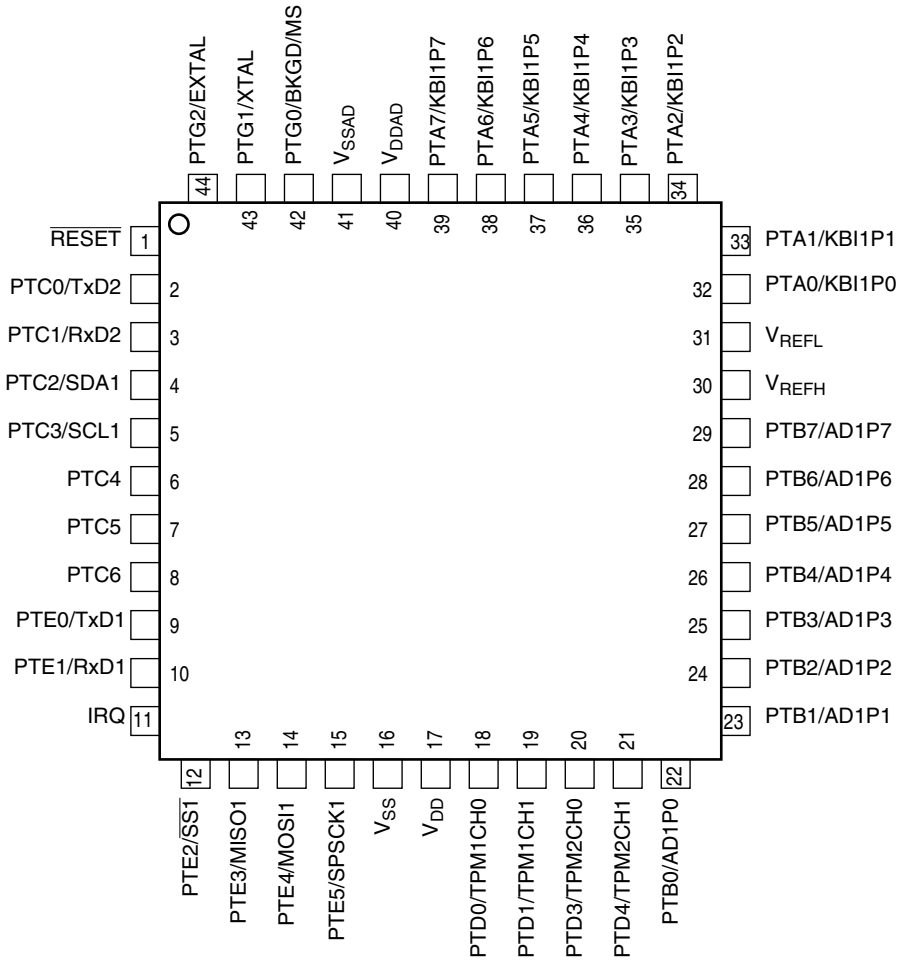

Figure 2-3. MC9S08GTxx in 44-Pin QFP Package

as the IRQ input and is set to detect rising edges, the pullup enable control bit enables a pulldown device rather than a pullup device.

2.3.6 Signal Properties Summary

Table 2-2 summarizes I/O pin characteristics. These characteristics are determined by the way the common pin interfaces are hardwired to internal circuits.

Table 2-2. Signal Properties

Pin Name	Dir	High Current Pin	Output Slew [1]	Pull-Up[2]	Comments
V_{DD}		—	—	—	
V_{SS}		—	—	—	The 48-pin QFN package has two V_{SS} pins — V_{SS1} and V_{SS2}.
V_{DDAD}		—	—	—	
V_{SSAD}		—	—	—	
V_{REFH}		—	—	—	
V_{REFL}		—	—	—	
$\overline{RESET}$	I/O	Y	N	Y	Pin contains integrated pullup.
IRQ	I	—	—	Y	IRQPE must be set to enable IRQ function. IRQ does not have a clamp diode to V_{DD}. IRQ should not be driven above V_{DD}. Pullup/pulldown active when IRQ pin function enabled. Pullup forced on when IRQ enabled for falling edges; pulldown forced on when IRQ enabled for rising edges.
PTA0/KBI1P0	I/O	N	SWC	SWC	
PTA1/KBI1P1	I/O	N	SWC	SWC	
PTA2/KBI1P2	I/O	N	SWC	SWC	
PTA3/KBI1P3	I/O	N	SWC	SWC	
PTA4/KBI1P4	I/O	N	SWC	SWC	Pullup/pulldown active when KBI pin function enabled. Pullup forced on when KBI1Px enabled for falling edges; pulldown forced on when KBI1Px enabled for rising edges.
PTA5/KBI1P5	I/O	N	SWC	SWC	
PTA6/KBI1P6	I/O	N	SWC	SWC	
PTA7/KBI1P7	I/O	N	SWC	SWC	
PTB0/AD1P0	I/O	N	SWC	SWC	
PTB1/AD1P1	I/O	N	SWC	SWC	
PTB2/AD1P2	I/O	N	SWC	SWC	
PTB3/AD1P3	I/O	N	SWC	SWC	
PTB4/AD1P4	I/O	N	SWC	SWC	
PTB5/AD1P5	I/O	N	SWC	SWC	
PTB6/AD1P6	I/O	N	SWC	SWC	
PTB7/AD1P7	I/O	N	SWC	SWC	
PTC0/TxD2	I/O	Y	SWC	SWC	When pin is configured for SCI function, pin is configured for partial output drive.
PTC1/RxD2	I/O	Y	SWC	SWC	
PTC2/SDA1	I/O	Y	SWC	SWC	
PTC3/SCL1	I/O	Y	SWC	SWC	

I/O Pins

- All I/O pin states remain unchanged when the MCU enters stop3 mode.
- If the MCU is configured to go into stop2 mode, all I/O pins states are latched before entering stop.
- If the MCU is configured to go into stop1 mode, all I/O pins are forced to their default reset state upon entry into stop.

Memory

- All RAM and register contents are preserved while the MCU is in stop3 mode.
- All registers will be reset upon wake-up from stop2, but the contents of RAM are preserved and pin states remain latched until the PPDACK bit is written. The user may save any memory-mapped register data into RAM before entering stop2 and restore the data upon exit from stop2.
- All registers will be reset upon wake-up from stop1 and the contents of RAM are not preserved. The MCU must be initialized as upon reset. The contents of the FLASH memory are nonvolatile and are preserved in any of the stop modes.

ICG — In stop3 mode, the ICG enters its low-power standby state. Either the oscillator or the internal reference may be kept running when the ICG is in standby by setting the appropriate control bit. In both stop2 and stop1 modes, the ICG is turned off. Neither the oscillator nor the internal reference can be kept running in stop2 or stop1, even if enabled within the ICG module.

TPM — When the MCU enters stop mode, the clock to the TPM1 and TPM2 modules stop. The modules halt operation. If the MCU is configured to go into stop2 or stop1 mode, the TPM modules will be reset upon wake-up from stop and must be reinitialized.

ATD — When the MCU enters stop mode, the ATD will enter a low-power standby state**.** No conversion operation will occur while in stop. If the MCU is configured to go into stop2 or stop1 mode, the ATD will be reset upon wake-up from stop and must be reinitialized.

KBI — During stop3, the KBI pins that are enabled continue to function as interrupt sources that are capable of waking the MCU from stop3. The KBI is disabled in stop1 and stop2 and must be reinitialized after waking up from either of these modes.

SCI — When the MCU enters stop mode, the clocks to the SCI1 and SCI2 modules stop. The modules halt operation. If the MCU is configured to go into stop2 or stop1 mode, the SCI modules will be reset upon wake-up from stop and must be reinitialized.

SPI — When the MCU enters stop mode, the clocks to the SPI module stop. The module halts operation. If the MCU is configured to go into stop2 or stop1 mode, the SPI module will be reset upon wake-up from stop and must be reinitialized.

IIC — When the MCU enters stop mode, the clocks to the IIC module stops. The module halts operation. If the MCU is configured to go into stop2 or stop1 mode, the IIC module will be reset upon wake-up from stop and must be reinitialized.

Voltage Regulator — The voltage regulator enters a low-power standby state when the MCU enters any of the stop modes unless the LVD is enabled in stop mode or BDM is enabled.

4.2 Register Addresses and Bit Assignments

The registers in the MC9S08GB/GT are divided into these three groups:

- Direct-page registers are located in the first 128 locations in the memory map, so they are accessible with efficient direct addressing mode instructions.
- High-page registers are used much less often, so they are located above $1800 in the memory map. This leaves more room in the direct page for more frequently used registers and variables.
- The nonvolatile register area consists of a block of 16 locations in FLASH memory at $FFB0–$FFBF.

  Nonvolatile register locations include:

  — Three values which are loaded into working registers at reset

  — An 8-byte backdoor comparison key which optionally allows a user to gain controlled access to secure memory

  Because the nonvolatile register locations are FLASH memory, they must be erased and programmed like other FLASH memory locations.

Direct-page registers can be accessed with efficient direct addressing mode instructions. Bit manipulation instructions can be used to access any bit in any direct-page register. Table 4-2 is a summary of all user-accessible direct-page registers and control bits.

The direct page registers in Table 4-2 can use the more efficient direct addressing mode which only requires the lower byte of the address. Because of this, the lower byte of the address in column one is shown in bold text. In Table 4-3 and Table 4-4 the whole address in column one is shown in bold. In Table 4-2, Table 4-3, and Table 4-4, the register names in column two are shown in bold to set them apart from the bit names to the right. Cells that are not associated with named bits are shaded. A shaded cell with a 0 indicates this unused bit always reads as a 0. Shaded cells with dashes indicate unused or reserved bit locations that could read as 1s or 0s.

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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$004F	Reserved	0	0	0	0	0	0	0	0
$0050	ATD1C	ATDPU	DJM	RES8	SGN	PRS			
$0051	ATD1SC	CCF	ATDIE	ATDCO	ATDCH				
$0052	ATD1RH	Bit 7	6	5	4	3	2	1	Bit 0
$0053	ATD1RL	Bit 7	6	5	4	3	2	1	Bit 0
$0054	ATD1PE	ATDPE7	ATDPE6	ATDPE5	ATDPE4	ATDPE3	ATDPE2	ATDPE1	ATDPE0
$0055–$0057	Reserved	—	—	—	—	—	—	—	—
$0058	IIC1A	ADDR							0
$0059	IIC1F	MULT		ICR					
$005A	IIC1C	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
$005B	IIC1S	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
$005C	IIC1D	DATA							
$005D–$005F	Reserved	—	—	—	—	—	—	—	—
$0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0061	TPM2CNTH	Bit 15	14	13	12	11	10	9	Bit 8
$0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
$0063	TPM2MODH	Bit 15	14	13	12	11	10	9	Bit 8
$0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
$0065	TPM2C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
$0066	TPM2C0VH	Bit 15	14	13	12	11	10	9	Bit 8
$0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
$0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
$0069	TPM2C1VH	Bit 15	14	13	12	11	10	9	Bit 8
$006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
$006B	TPM2C2SC	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
$006C	TPM2C2VH	Bit 15	14	13	12	11	10	9	Bit 8
$006D	TPM2C2VL	Bit 7	6	5	4	3	2	1	Bit 0
$006E	TPM2C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
$006F	TPM2C3VH	Bit 15	14	13	12	11	10	9	Bit 8
$0070	TPM2C3VL	Bit 7	6	5	4	3	2	1	Bit 0
$0071	TPM2C4SC	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	0	0
$0072	TPM2C4VH	Bit 15	14	13	12	11	10	9	Bit 8
$0073	TPM2C4VL	Bit 7	6	5	4	3	2	1	Bit 0
$0074–$007F	Reserved	—	—	—	—	—	—	—	—

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 FLASH memory cannot be accessed by read operations while KEYACC is set.

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 just 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.

4.6 FLASH Registers and Control Bits

The FLASH module has nine 8-bit registers in the high-page register space, three locations in the nonvolatile register space in FLASH memory that are copied into three corresponding high-page control registers at reset. There is also an 8-byte comparison key in FLASH memory. Refer to Table 4-3 and Table 4-4 for the absolute address assignments for all FLASH registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

When the CPU receives a qualified interrupt request, it completes the current instruction before responding to the interrupt. The interrupt sequence follows the same cycle-by-cycle sequence as the SWI instruction and consists of:

- Saving the CPU registers on the stack
- Setting the I bit in the CCR to mask further interrupts
- Fetching the interrupt vector for the highest-priority interrupt that is currently pending
- Filling the instruction queue with the first three bytes of program information starting from the address fetched from the interrupt vector locations

While the CPU is responding to the interrupt, the I bit is automatically set to avoid the possibility of another interrupt interrupting the ISR itself (this is called nesting of interrupts). Normally, the I bit is restored to 0 when the CCR is restored from the value stacked on entry to the ISR. In rare cases, the I bit may be cleared inside an ISR (after clearing the status flag that generated the interrupt) so that other interrupts can be serviced without waiting for the first service routine to finish. This practice is not recommended for anyone other than the most experienced programmers because it can lead to subtle program errors that are difficult to debug.

The interrupt service routine ends with a return-from-interrupt (RTI) instruction which restores the CCR, A, X, and PC registers to their pre-interrupt values by reading the previously saved information off the stack.

NOTE

For compatibility with the M68HC08, the H register is not automatically saved and restored. It is good programming practice to push H onto the stack at the start of the interrupt service routine (ISR) and restore it just before the RTI that is used to return from the ISR.

When two or more interrupts are pending when the I bit is cleared, the highest priority source is serviced first (see Table 5-1).

5.5.1 Interrupt Stack Frame

Figure 5-1 shows the contents and organization of a stack frame. Before the interrupt, the stack pointer (SP) points at the next available byte location on the stack. The current values of CPU registers are stored on the stack starting with the low-order byte of the program counter (PCL) and ending with the CCR. After stacking, the SP points at the next available location on the stack which is the address that is one less than the address where the CCR was saved. The PC value that is stacked is the address of the instruction in the main program that would have executed next if the interrupt had not occurred.

LVD — Low Voltage Detect

If the LVD reset is enabled (LVDE = LVDRE = 1) and the supply drops below the LVD trip voltage, an LVD reset occurs. The LVD function is disabled when the MCU enters stop. To maintain LVD operation in stop, the LVDSE bit must be set.

1 = Reset caused by LVD trip or POR.

0 = Reset not caused by LVD trip or POR.

5.8.3 System Background Debug Force Reset Register (SBDFR)

This register contains a single write-only control bit. A serial background command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return $00.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								BDFR[1]
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

[1] BDFR is writable only through serial background debug commands, not from user programs.

Figure 5-4. System Background Debug Force Reset Register (SBDFR)

BDFR — Background Debug Force Reset

A serial background mode command such as WRITE_BYTE allows an external debug host to force a target system reset. Writing 1 to this bit forces an MCU reset. This bit cannot be written from a user program.

5.8.4 System Options Register (SOPT)

This register may be read at any time. Bits 3 and 2 are unimplemented and always read 0. This is a write-once register so only the first write after reset is honored. Any subsequent attempt to write to SOPT (intentionally or unintentionally) is ignored to avoid accidental changes to these sensitive settings. SOPT should be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COPE	COPT	STOPE		0	0	BKGDPE	
Write:								
Reset:	1	1	0	1	0	0	1	1

= Unimplemented or Reserved

Figure 5-5. System Options Register (SOPT)

6.3.6 Port F and High-Current Drivers

Port F		Bit 7	6	5	4	3	2	1	Bit 0
	MCU Pin:	PTF7	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0

Figure 6-7. Port F Pin Names

Port F is an 8-bit port general-purpose I/O that is not shared with any peripheral module. Port F has high current output drivers.

Port F pins are available as general-purpose I/O pins controlled by the port F data (PTFD), data direction (PTFDD), pullup enable (PTFPE), and slew rate control (PTFSE) registers. Refer to Section 6.4, “Parallel I/O Controls” for more information about general-purpose I/O control.

6.3.7 Port G, BKGD/MS, and Oscillator

Port G		Bit 7	6	5	4	3	2	1	Bit 0
	MCU Pin:	PTG7	PTG6	PTG5	PTG4	PTG3	PTG2/ EXTAL	PTG1/ XTAL	PTG0/ BKGD/MS

Figure 6-8. Port G Pin Names

Port G is an 8-bit port which is shared among the background/mode select function, oscillator, and general-purpose I/O. When the background/mode select function or oscillator is enabled, the pin direction will be controlled by the module function.

Port G pins are available as general-purpose I/O pins controlled by the port G data (PTGD), data direction (PTGDD), pullup enable (PTGPE), and slew rate control (PTGSE) registers. Refer to Section 6.4, “Parallel I/O Controls” for more information about general-purpose I/O control.

The internal pullup for PTG0 is enabled when the background/mode select function is enabled, regardless of the state of PTGPE0. During reset, the BKGD/MS pin functions as a mode select pin. After the MCU is out of reset, the BKGD/MS pin becomes the background communications input/output pin. The PTG0 can be configured to be a general-purpose output pin. Refer to Chapter 3, “Modes of Operation”, Chapter 5, “Resets, Interrupts, and System Configuration”, and Chapter 15, “Development Support” for more information about using this pin.

The ICG module can be configured to use PTG2–PTG1 ports as crystal oscillator or external clock pins.

Refer to Chapter 13, “Inter-Integrated Circuit (IIC) Module” for more information about using these pins as oscillator pins.

6.4 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, F, or G.

LOCRE — Loss of Clock Reset Enable

The LOCRE bit determines how the system handles a loss of clock condition.

1 = Generate a reset request on loss of clock.

0 = Generate an interrupt request on loss of clock.

RFD — Reduced Frequency Divider

The RFD bits control the value of the divider following the clock select circuitry. The value specified by the RFD bits establishes the division factor (R) applied to the selected output clock source. Writes to the RFD bits will not take effect if a previous write is not complete.

Table 7-8. RFD Reduced Frequency Divider Select

RFD	Division Factor (R)
000	÷1
001	÷2
010	÷4
011	÷8
100	÷16
101	÷32
110	÷64
111	÷128

7.5.3 ICG Status Register 1 (ICGS1)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CLKST		REFST	LOLS	LOCK	LOCS	ERCS	ICGIF
Write:								1
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-15. ICG Status Register 1 (ICGS1)

CLKST — Clock Mode Status

The CLKST bits indicate the current clock mode. The CLKST bits don't update immediately after a write to the CLKS bits due to internal synchronization between clock domains.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the TxD1 pin, the transmitter sets the transmit complete flag and enters an idle mode, with TxD1 high, waiting for more characters to transmit.

Writing 0 to TE does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity that is in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

11.5.2 Send Break and Queued Idle

The SBK control bit in SCIxC2 is used to send break characters that were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (including a 0 where the stop bit would be normally). Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another Freescale Semiconductor SCI, the break characters will be received as 0s in all eight (or nine) data bits and a framing error (FE = 1).

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD1 pin. If there is a possibility of the shifter finishing while TE = 0, set the general-purpose I/O controls so the pin that is shared with TxD1 is an output driving a logic 1. This ensures that the TxD1 line will look like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE.

11.6 Receiver Functional Description

In this section, the receiver block diagram (Figure 11-4) is used as a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wakeup function are explained.

11.6.1 Receiver Block Diagram

Figure 11-4 shows the receiver portion of the SCI.

SCISWAI — SCI Stops in Wait Mode

1 = SCI clocks freeze while CPU is in wait mode.

0 = SCI clocks continue to run in wait mode so the SCI can be the source of an interrupt that wakes up the CPU.

RSRC — Receiver Source Select

This bit has no meaning or effect unless the LOOPS bit is set to 1. When LOOPS = 1, the receiver input is internally connected to the TxD1 pin and RSRC determines whether this connection is also connected to the transmitter output.

1 = Single-wire SCI mode where the TxD1 pin is connected to the transmitter output and receiver input.

0 = Provided LOOPS = 1, RSRC = 0 selects internal loop back mode and the SCI does not use the RxD1 or TxD1 pins.

M — 9-Bit or 8-Bit Mode Select

1 = Receiver and transmitter use 9-bit data characters start + 8 data bits (LSB first) + 9th data bit + stop.

0 = Normal — start + 8 data bits (LSB first) + stop.

WAKE — Receiver Wakeup Method Select

Refer to Section 11.6.3, “Receiver Wakeup Operation,” for more information.

1 = Address-mark wakeup.

0 = Idle-line wakeup.

ILT — Idle Line Type Select

Setting this bit to 1 ensures that the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of the logic high level by the idle line detection logic. Refer to Section 11.6.3.1, “Idle-Line Wakeup,” for more information.

1 = Idle character bit count starts after stop bit.

0 = Idle character bit count starts after start bit.

PE — Parity Enable

Enables hardware parity generation and checking. When parity is enabled, the most significant bit (MSB) of the data character (eighth or ninth data bit) is treated as the parity bit.

1 = Parity enabled.

0 = No hardware parity generation or checking.

PT — Parity Type

Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even.

1 = Odd parity.

0 = Even parity.

12.3.1 SPI Clock Formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a clock polarity (CPOL) bit and a clock phase (CPHA) control bit to select one of four clock formats for data transfers. CPOL selectively inserts an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data.

Figure 12-5 shows the clock formats when CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCK edge and bit 8 ending one-half SPSCK cycle after the sixteenth SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output 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.

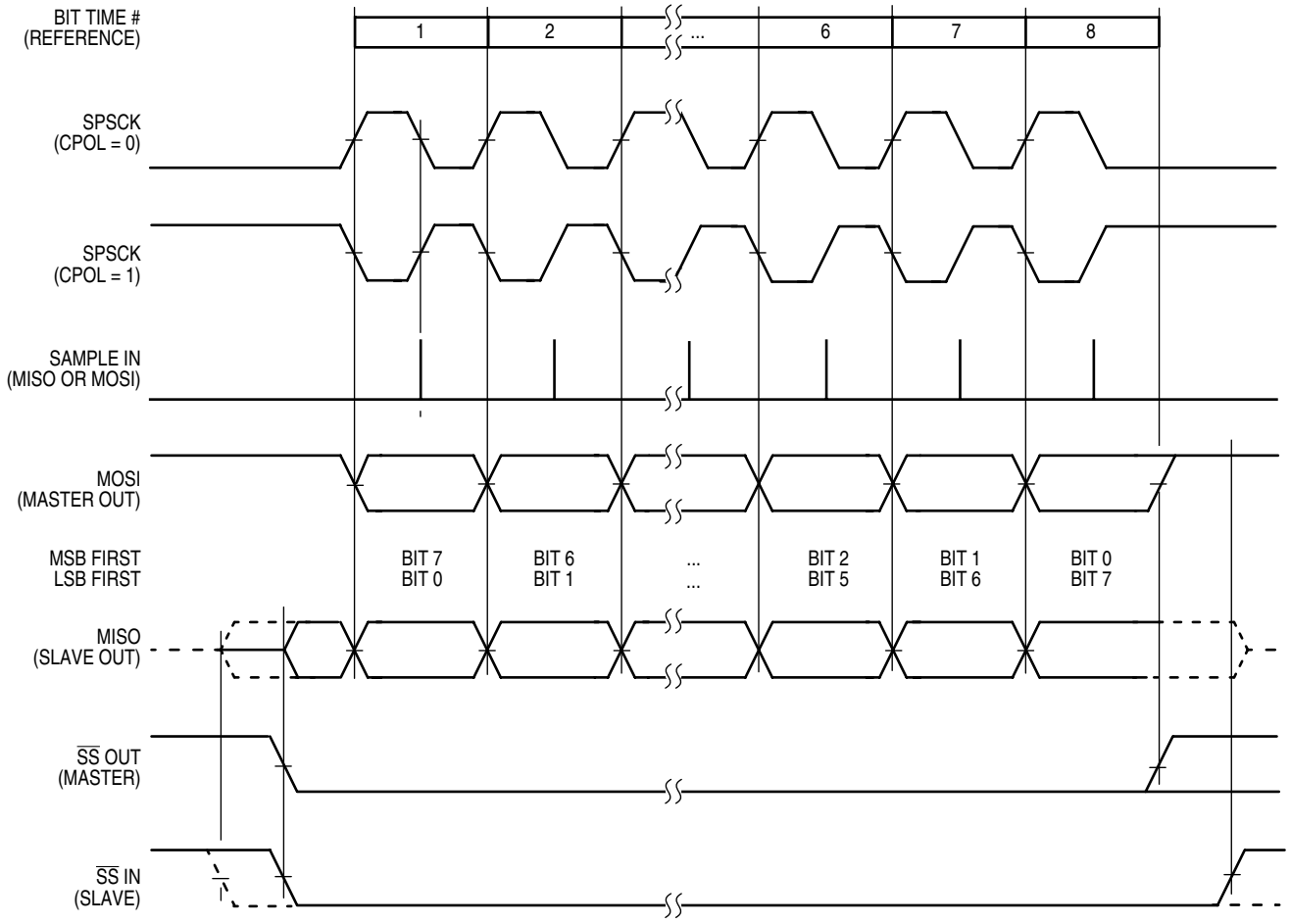

Figure 12-5. SPI Clock Formats (CPHA = 1)

13.2.1.1 START Signal

When the bus is free; i.e., no master device is engaging the bus (both SCL and SDA lines are at logical high), a master may initiate communication by sending a START signal. As shown in Figure 13-3, 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.

13.2.1.2 Slave Address Transmission

The first byte of data transferred immediately after the START signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/W bit. The R/W bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master will respond by sending back an acknowledge bit. This is done by pulling the SDA low at the 9th clock (see Figure 13-3).

No two slaves in the system may have the same address. If the IIC module is the master, it must not transmit an address that is equal to its own slave address. The IIC cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the IIC will revert to slave mode and operate correctly even if it is being addressed by another master.

13.2.1.3 Data Transfer

Before successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/W bit sent by the calling master.

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 13-3. There is one clock pulse on SCL for each data bit, the MSB being transferred first. Each data byte is followed by a 9th (acknowledge) bit, which is signalled from the receiving device. An acknowledge is signalled by pulling the SDA low at the ninth clock. In summary, one complete data transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master in the 9th bit time, the SDA line must be left high by the slave. The master interprets the failed acknowledge as an unsuccessful data transfer.

If the master receiver does not acknowledge the slave transmitter after a data byte transmission, the slave interprets this as an end of data transfer and releases the SDA line.

In either case, the data transfer is aborted and the master does one of two things:

- Relinquishes the bus by generating a STOP signal.
- Commences a new calling by generating a repeated START signal.

14.4 Resets

The ATD module is reset on system reset. If the system reset signal is activated, the ATD registers are initialized back to their reset state and the ATD module is powered down. This occurs as a function of the register file initialization; the reset definition of the ATDPU bit (power down bit) is zero or disabled.

The MCU places the module back into an initialized state. If the module is performing a conversion, the current conversion is terminated, the conversion complete flag is cleared, and the SAR register bits are cleared. Any pending interrupts are also cancelled. Note that the control, test, and status registers are initialized on reset; the initialized register state is defined in the register description section of this specification.

Enabling the module (using the ATDPU bit) does not cause the module to reset since the register file is not initialized. Finally, writing to control register ATD1C does not cause the module to reset; the current conversion will be terminated.

14.5 Interrupts

The ATD module originates interrupt requests and the MCU handles or services these requests. Details on how the ATD interrupt requests are handled can be found in Chapter 5, “Resets, Interrupts, and System Configuration”.

The ATD interrupt function is enabled by setting the ATDIE bit in the ATD1SC register. When the ATDIE bit is set, an interrupt is generated at the end of an ATD conversion and the ATD result registers (ATD1RH and ATD1RL) contain the result data generated by the conversion. If the interrupt function is disabled (ATDIE = 0), then the CCF flag must be polled to determine when a conversion is complete.

The interrupt will remain pending as long as the CCF flag is set. The CCF bit is cleared whenever the ATD status and control (ATD1SC) register is written. The CCF bit is also cleared whenever the ATD result registers (ATD1RH or ATD1RL) are read.

Table 14-2. Interrupt Summary

Interrupt	Local Enable	Description
CCF	ATDIE	Conversion complete

14.6 ATD Registers and Control Bits

The ATD has seven registers which control ATD functions.

Refer to the direct-page register summary in Chapter 4, “Memory” of this data sheet for the absolute address assignments for all ATD registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

RWAEN — Enable R/W for Comparator A

Controls whether the level of R/W is considered for a comparator A match.
1 = R/W is used in comparison A.
0 = R/W is not used in comparison A.

RWB — R/W Comparison Value for Comparator B

When RWBEN = 1, this bit determines whether a read or a write access qualifies comparator B. When RWBEN = 0, RWB and the R/W signal do not affect comparator B.
1 = Comparator B can match only on a read cycle.
0 = Comparator B can match only on a write cycle.

RWBEN — Enable R/W for Comparator B

Controls whether the level of R/W is considered for a comparator B match.
1 = R/W is used in comparison B.
0 = R/W is not used in comparison B.

15.5.3.8 Debug Trigger Register (DBGT)

This register can be read any time, but may be written only if ARM = 0, except bits 4 and 5 are hard-wired to 0s.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-8. Debug Trigger Register (DBGT)

TRGSEL — Trigger Type

Controls whether the match outputs from comparators A and B are qualified with the opcode tracking logic in the debug module. If TRGSEL is set, a match signal from comparator A or B must propagate through the opcode tracking logic and a trigger event is only signalled to the FIFO logic if the opcode at the match address is actually executed.
1 = Trigger if opcode at compare address is executed (tag).
0 = Trigger on access to compare address (force).

BEGIN — Begin/End Trigger Select

Controls whether the FIFO starts filling at a trigger or fills in a circular manner until a trigger ends the capture of information. In event-only trigger modes, this bit is ignored and all debug runs are assumed to be begin traces.
1 = Trigger initiates data storage (begin trace).
0 = Data stored in FIFO until trigger (end trace).

Table A-7. ATD Timing/Performance Characteristics[1]

Num	Characteristic	Symbol	Condition	Min	Typ	Max	Unit
1	ATD conversion clock frequency	f_{ATDCLK}	$2.08V \le V_{DDAD} \le 3.6V$	0.5	—	2.0	MHz
			$1.80V \le V_{DDAD} < 2.08V$	0.5	—	1.0	
2	Conversion cycles (continuous convert)[2]	CC		28	28	<30	ATDCLK cycles
3	Conversion time	T_{conv}	$2.08V \le V_{DDAD} \le 3.6V$	14.0	—	60.0	μS
			$1.80V \le V_{DDAD} < 2.08V$	28.0	—	60.0	
4	Source impedance at input[3]	R_{AS}		—	—	10	kΩ
5	Analog Input Voltage[4]	V_{AIN}		V_{REFL}		V_{REFH}	V
6	Ideal resolution (1 LSB)[5]	RES	$2.08V \le V_{DDAD} \le 3.6V$	2.031	—	3.516	mV
			$1.80V \le V_{DDAD} < 2.08V$	1.758	—	2.031	
7	Differential non-linearity[6]	DNL	$1.80V \le V_{DDAD} \le 3.6V$	—	±0.5	±1.0	LSB
8	Integral non-linearity[7]	INL	$1.80\ V \le V_{DDAD} \le 3.6V$	—	±0.5	±1.0	LSB
9	Zero-scale error[8]	E_{ZS}	$1.80V \le V_{DDAD} \le 3.6V$	—	±0.4	±1.0	LSB
10	Full-scale error[9]	E_{FS}	$1.80V \le V_{DDAD} \le 3.6V$	—	±0.4	±1.0	LSB
11	Input leakage error [10]	E_{IL}	$1.80V \le V_{DDAD} \le 3.6V$	—	±0.05	±5	LSB
12	Total unadjusted error[11]	E_{TU}	$1.80V \le V_{DDAD} \le 3.6V$	—	±1.1	±2.5	LSB

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).

Table A-11. TPM Input Timing

Function	Symbol	Min	Max	Unit
External clock frequency	f_{TPMext}	dc	$f_{Bus}/4$	MHz
External clock period	t_{TPMext}	4	—	t_{cyc}
External clock high time	t_{clkh}	1.5	—	t_{cyc}
External clock low time	t_{clkl}	1.5	—	t_{cyc}
Input capture pulse width	t_{ICPW}	1.5	—	t_{cyc}

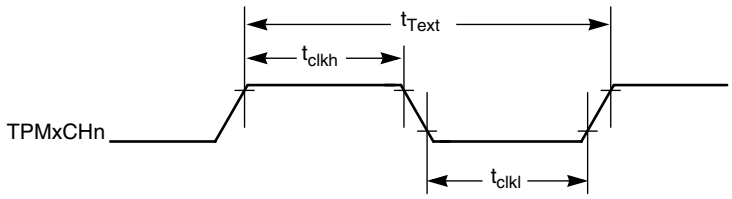

Figure A-14. Timer External Clock

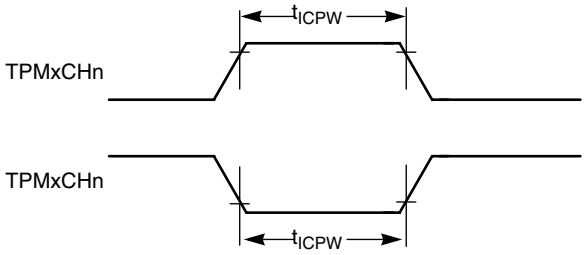

Figure A-15. Timer Input Capture Pulse

A.9.3 SPI Timing

Table A-12 and Figure A-16 through Figure A-19 describe the timing requirements for the SPI system.

B.4 48-Pin QFN Package Drawing