



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 | I ² C, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 54 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 16x10b |
| 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/s9s08aw32e5cpue |

| Section Number | Title | Page |
|----------------|---|------|
| 8.4.4 | FLL Engaged Internal Unlocked | 143 |
| 8.4.5 | FLL Engaged Internal Locked | 143 |
| 8.4.6 | FLL Bypassed, External Clock (FBE) Mode | 143 |
| 8.4.7 | FLL Engaged, External Clock (FEE) Mode | 143 |
| 8.4.8 | FLL Lock and Loss-of-Lock Detection | 144 |
| 8.4.9 | FLL Loss-of-Clock Detection | 145 |
| 8.4.10 | Clock Mode Requirements | 146 |
| 8.4.11 | Fixed Frequency Clock | 147 |
| 8.4.12 | High Gain Oscillator | 147 |
| 8.5 | Initialization/Application Information | 147 |
| 8.5.1 | Introduction | 147 |
| 8.5.2 | Example #1: External Crystal = 32 kHz, Bus Frequency = 4.19 MHz | 149 |
| 8.5.3 | Example #2: External Crystal = 4 MHz, Bus Frequency = 20 MHz | 151 |
| 8.5.4 | Example #3: No External Crystal Connection, 5.4 MHz Bus Frequency | 153 |
| 8.5.5 | Example #4: Internal Clock Generator Trim | 155 |

Chapter 9 Keyboard Interrupt (S08KBIV1)

| | | |
|-------|--|-----|
| 9.1 | Introduction | 157 |
| 9.2 | Keyboard Pin Sharing | 157 |
| 9.3 | Features | 158 |
| 9.3.1 | KBI Block Diagram | 160 |
| 9.4 | Register Definition | 160 |
| 9.4.1 | KBI Status and Control Register (KBI1SC) | 161 |
| 9.4.2 | KBI Pin Enable Register (KBI1PE) | 162 |
| 9.5 | Functional Description | 162 |
| 9.5.1 | Pin Enables | 162 |
| 9.5.2 | Edge and Level Sensitivity | 162 |
| 9.5.3 | KBI Interrupt Controls | 163 |

Chapter 10 Timer/PWM (S08TPMV2)

| | | |
|--------|---|-----|
| 10.1 | Introduction | 165 |
| 10.2 | Features | 165 |
| 10.2.1 | Features | 167 |
| 10.2.2 | Block Diagram | 167 |
| 10.3 | External Signal Description | 169 |
| 10.3.1 | External TPM Clock Sources | 169 |
| 10.3.2 | TPMxCHn — TPMx Channel n I/O Pins | 169 |
| 10.4 | Register Definition | 169 |
| 10.4.1 | Timer x Status and Control Register (TPMxSC) | 170 |
| 10.4.2 | Timer x Counter Registers (TPMxCNTH:TPMxCNTL) | 171 |

Chapter 4 Memory

4.1 MC9S08AW60 Series Memory Map

Figure 4-1 shows the memory map for the MC9S08AW60 and MC9S08AW48 MCUs. Figure 4-2 shows the memory map for the MC9S08AW32 and MC9S08AW16 MCUs. On-chip memory in the MC9S08AW60 Series of MCUs consists of RAM, FLASH program memory for nonvolatile data storage, plus I/O and control/status registers. The registers are divided into three groups:

- Direct-page registers (\$0000 through \$006F)
- High-page registers (\$1800 through \$185F)
- Nonvolatile registers (\$FFB0 through \$FFBF)

NOTE

The voltage measured on the pulled up IRQ pin may be as low as $V_{DD} - 0.7$ V. The internal gates connected to this pin are pulled all the way to V_{DD} . All other pins with enabled pullup resistors will have an unloaded measurement of V_{DD} .

5.5.2.2 Edge and Level Sensitivity

The IRQMOD control bit reconfigures the detection logic so it detects edge events and pin levels. In this edge detection mode, the IRQF status flag becomes set when an edge is detected (when the IRQ pin changes from the deasserted to the asserted level), but the flag is continuously set (and cannot be cleared) as long as the IRQ pin remains at the asserted level.

5.5.3 Interrupt Vectors, Sources, and Local Masks

Table 5-1 provides a summary of all interrupt sources. Higher-priority sources are located toward the bottom of the table. The high-order byte of the address for the interrupt service routine is located at the first address in the vector address column, and the low-order byte of the address for the interrupt service routine is located at the next higher address.

When an interrupt condition occurs, an associated flag bit becomes set. If the associated local interrupt enable is 1, an interrupt request is sent to the CPU. Within the CPU, if the global interrupt mask (I bit in the CCR) is 0, the CPU will finish the current instruction, stack the PCL, PCH, X, A, and CCR CPU registers, set the I bit, and then fetch the interrupt vector for the highest priority pending interrupt. Processing then continues in the interrupt service routine.

6.6 Pin Behavior in Stop Modes

Depending on the stop mode, I/O functions differently as the result of executing a STOP instruction. An explanation of I/O behavior for the various stop modes follows:

- Stop2 mode is a partial power-down mode, whereby I/O latches are maintained in their state as before the STOP instruction was executed. CPU register status and the state of I/O registers should be saved in RAM before the STOP instruction is executed to place the MCU in stop2 mode. Upon recovery from stop2 mode, before accessing any I/O, the user should examine the state of the PPDF bit in the SPMSC2 register. If the PPDF bit is 0, I/O must be initialized as if a power on reset had occurred. If the PPDF bit is 1, I/O data previously stored in RAM, before the STOP instruction was executed, peripherals may require being initialized and restored to their pre-stop condition. The user must then write a 1 to the PPDACK bit in the SPMSC2 register. Access to I/O is now permitted again in the user’s application program.
- In stop3 mode, all I/O is maintained because internal logic circuitry stays powered up. Upon recovery, normal I/O function is available to the user.

6.7 Parallel I/O and Pin Control Registers

This section provides information about the registers associated with the parallel I/O ports and pin control functions. These parallel I/O registers are located in page zero of the memory map and the pin control registers are located in the high page register section of memory.

Refer to tables in Chapter 4, “Memory,” for the absolute address assignments for all parallel I/O and pin control 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.

6.7.1 Port A I/O Registers (PTAD and PTADD)

Port A parallel I/O function is controlled by the registers listed below.

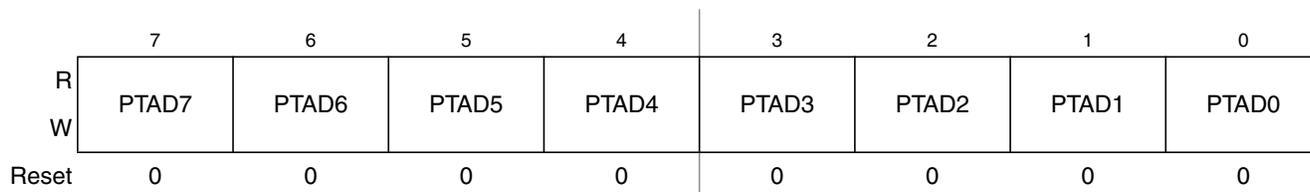


Figure 6-9. Port A Data Register (PTAD)

Table 6-2. PTAD Register Field Descriptions

| Field | Description |
|------------------|--|
| 7:0 PTAD[7:0] | <p>Port A Data Register Bits — For port A pins that are inputs, reads return the logic level on the pin. For port A 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 A pins that are configured as outputs, the logic level is driven out the corresponding MCU pin.</p> <p>Reset forces PTAD 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.</p> |

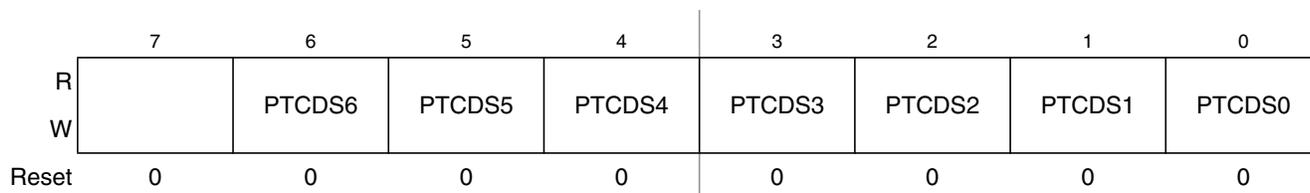


Figure 6-23. Output Drive Strength Selection for Port C (PTCDS)

Table 6-16. PTCDS Register Field Descriptions

| Field | Description |
|-------------------|---|
| 6:0 PTCDS[6:0] | <p>Output Drive Strength Selection for Port C Bits — Each of these control bits selects between low and high output drive for the associated PTC pin.</p> <p>0 Low output drive enabled for port C bit n.</p> <p>1 High output drive enabled for port C bit n.</p> |

6.7.11 Port F I/O Registers (PTFD and PTFDD)

Port F parallel I/O function is controlled by the registers listed below.

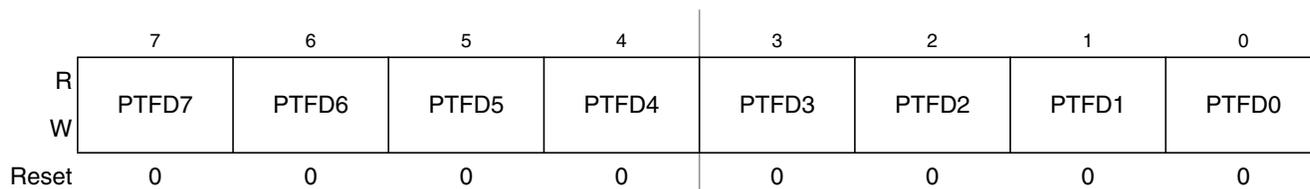


Figure 6-34. Port F Data Register (PTFD)

Table 6-27. PTFD Register Field Descriptions

| Field | Description |
|------------------|---|
| 7:0 PTFD[7:0] | <p>Port F Data Register Bits— For port F pins that are inputs, reads return the logic level on the pin. For port F 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 F pins that are configured as outputs, the logic level is driven out the corresponding MCU pin.</p> <p>Reset forces PTFD 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.</p> |

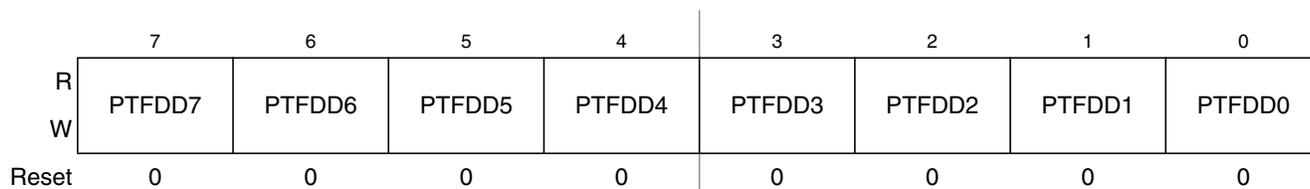


Figure 6-35. Data Direction for Port F (PTFDD)

Table 6-28. PTFDD Register Field Descriptions

| Field | Description |
|-------------------|---|
| 7:0 PTFDD[7:0] | <p>Data Direction for Port F Bits — These read/write bits control the direction of port F pins and what is read for PTFD reads.</p> <p>0 Input (output driver disabled) and reads return the pin value.</p> <p>1 Output driver enabled for port F bit n and PTFD reads return the contents of PTFDn.</p> |

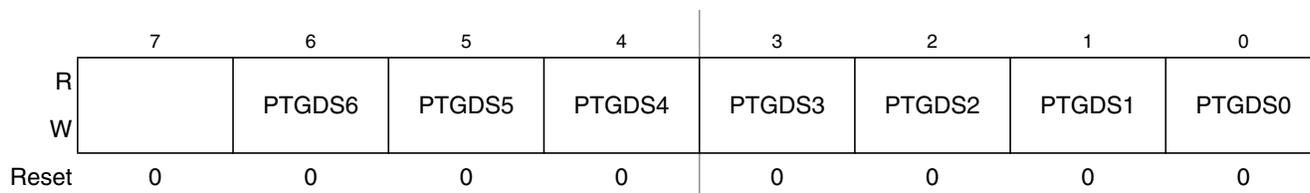


Figure 6-43. Output Drive Strength Selection for Port G (PTGDS)

Table 6-36. PTGDS Register Field Descriptions

| Field | Description |
|-------------------|---|
| 6:0 PTGDS[6:0] | <p>Output Drive Strength Selection for Port G Bits — Each of these control bits selects between low and high output drive for the associated PTG pin.</p> <p>0 Low output drive enabled for port G bit n.</p> <p>1 High output drive enabled for port G bit n.</p> |

7.3.6.7 SP-Relative, 16-Bit Offset (SP2)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.4 Special Operations

The CPU performs a few special operations that are similar to instructions but do not have opcodes like other CPU instructions. In addition, a few instructions such as STOP and WAIT directly affect other MCU circuitry. This section provides additional information about these operations.

7.4.1 Reset Sequence

Reset can be caused by a power-on-reset (POR) event, internal conditions such as the COP (computer operating properly) watchdog, or by assertion of an external active-low reset pin. When a reset event occurs, the CPU immediately stops whatever it is doing (the MCU does not wait for an instruction boundary before responding to a reset event). For a more detailed discussion about how the MCU recognizes resets and determines the source, refer to the Resets, Interrupts, and System Configuration chapter.

The reset event is considered concluded when the sequence to determine whether the reset came from an internal source is done and when the reset pin is no longer asserted. At the conclusion of a reset event, the CPU performs a 6-cycle sequence to fetch the reset vector from 0xFFFFE and 0xFFFF and to fill the instruction queue in preparation for execution of the first program instruction.

7.4.2 Interrupt Sequence

When an interrupt is requested, the CPU completes the current instruction before responding to the interrupt. At this point, the program counter is pointing at the start of the next instruction, which is where the CPU should return after servicing the interrupt. The CPU responds to an interrupt by performing the same sequence of operations as for a software interrupt (SWI) instruction, except the address used for the vector fetch is determined by the highest priority interrupt that is pending when the interrupt sequence started.

The CPU sequence for an interrupt is:

1. Store the contents of PCL, PCH, X, A, and CCR on the stack, in that order.
2. Set the I bit in the CCR.
3. Fetch the high-order half of the interrupt vector.
4. Fetch the low-order half of the interrupt vector.
5. Delay for one free bus cycle.
6. Fetch three bytes of program information starting at the address indicated by the interrupt vector to fill the instruction queue in preparation for execution of the first instruction in the interrupt service routine.

After the CCR contents are pushed onto the stack, the I bit in the CCR is set to prevent other interrupts while in the interrupt service routine. Although it is possible to clear the I bit with an instruction in the

Chapter 8 Internal Clock Generator (S08ICGV4)

The internal clock generation (ICG) module is used to generate the system clocks for the MC9S08AW60 Series MCU. The analog supply lines V_{DDA} and V_{SSA} are internally derived from the MCU's V_{DD} and V_{SS} pins. Electrical parametric data for the ICG may be found in Appendix A, “Electrical Characteristics and Timing Specifications.”

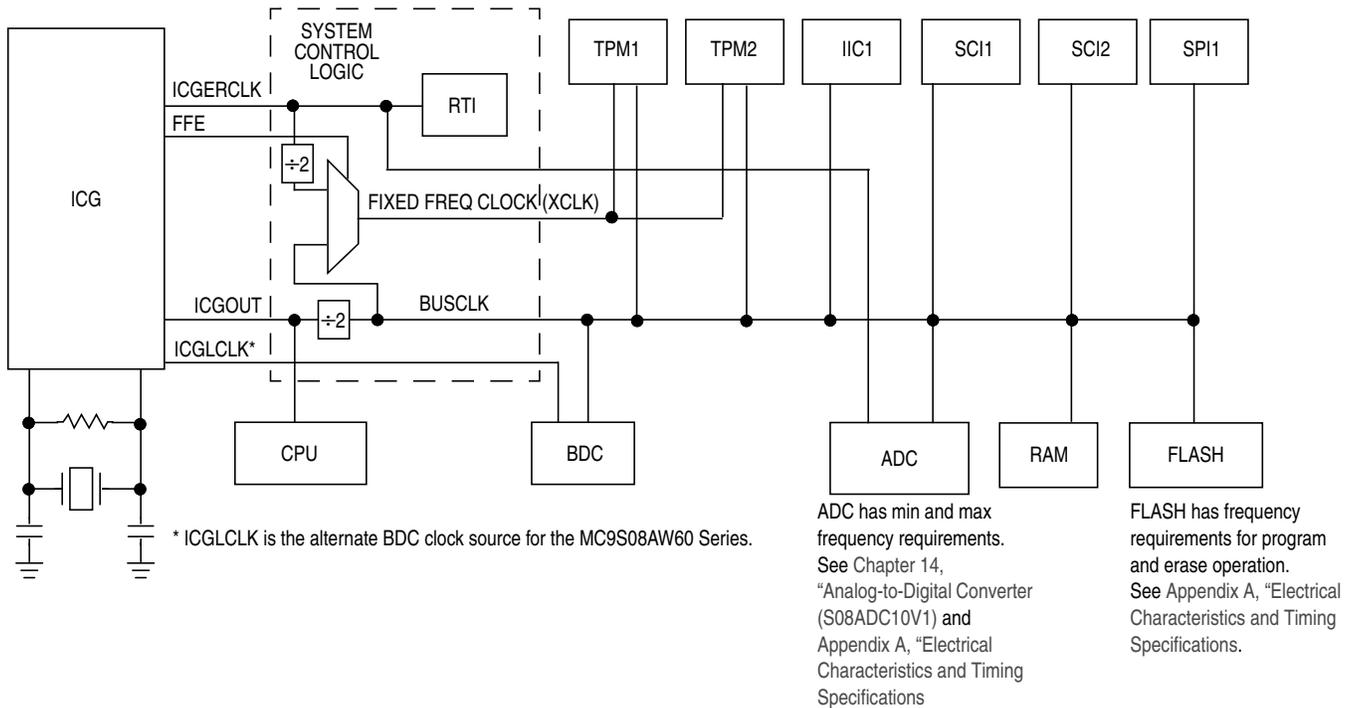


Figure 8-1. System Clock Distribution Diagram

NOTE

Freescle Semiconductor recommends that FLASH location \$FFBE be reserved to store a nonvolatile version of ICGTRM. This will allow debugger and programmer vendors to perform a manual trim operation and store the resultant ICGTRM value for users to access at a later time.

8.3.4 ICG Status Register 2 (ICGS2)

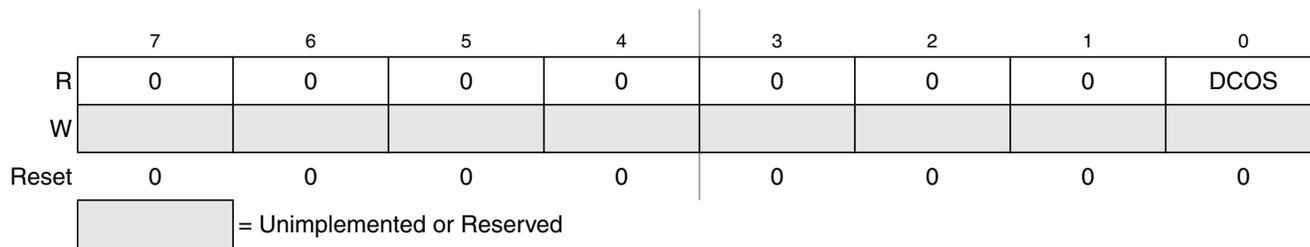


Figure 8-9. ICG Status Register 2 (ICGS2)

Table 8-4. ICGS2 Register Field Descriptions

| Field | Description |
|-----------|---|
| 0 DCOS | <p>DCO Clock Stable — The DCOS bit is set when the DCO clock (ICG2DCLK) is stable, meaning the count error has not changed by more than n_{unlock} for two consecutive samples and the DCO clock is not static. This bit is used when exiting off state if CLKS = X1 to determine when to switch to the requested clock mode. It is also used in self-clocked mode to determine when to start monitoring the DCO clock. This bit is cleared upon entering the off state.</p> <p>0 DCO clock is unstable. 1 DCO clock is stable.</p> |

8.3.5 ICG Filter Registers (ICGFLTU, ICGFLTL)

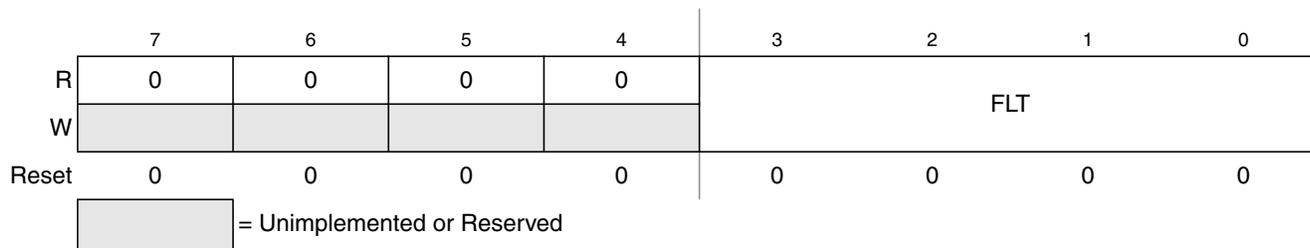


Figure 8-10. ICG Upper Filter Register (ICGFLTU)

Table 8-5. ICGFLTU Register Field Descriptions

| Field | Description |
|------------|---|
| 3:0 FLT | <p>Filter Value — The FLT bits indicate the current filter value, which controls the DCO frequency. The FLT bits are read only except when the CLKS bits are programmed to self-clocked mode (CLKS = 00). In self-clocked mode, any write to ICGFLTU updates the current 12-bit filter value. Writes to the ICGFLTU register will not affect FLT if a previous latch sequence is not complete.</p> |

9.3.1 KBI Block Diagram

Figure 9-2 shows the block diagram for a KBI module.

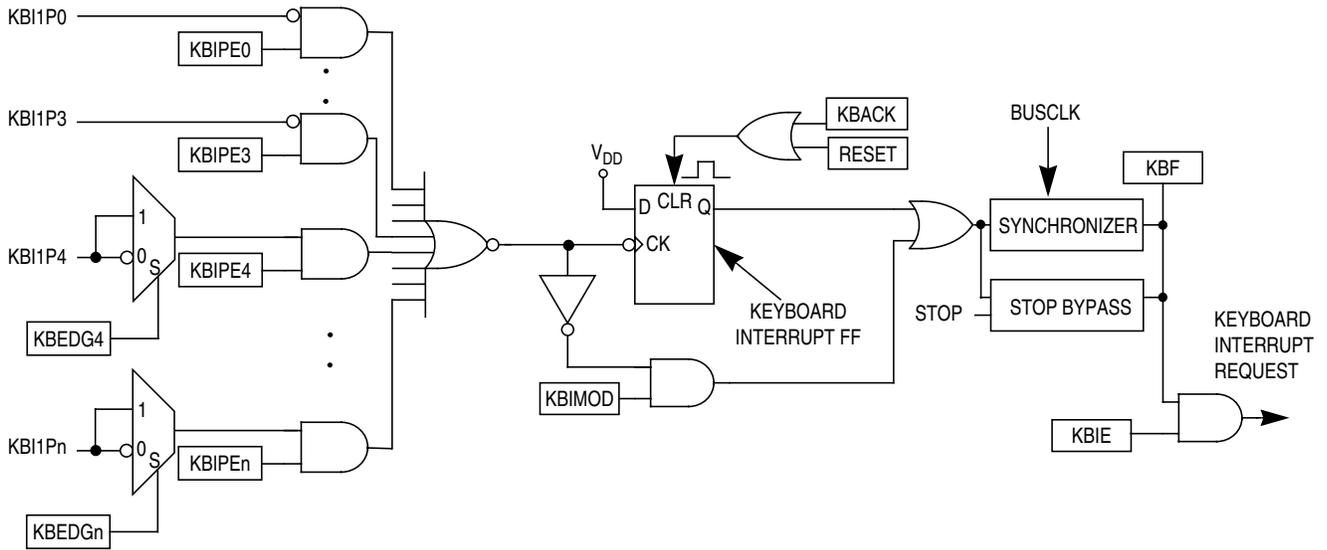


Figure 9-2. KBI Block Diagram

9.4 Register Definition

This section provides information about all registers and control bits associated with the KBI module.

Refer to the direct-page register summary in the Memory chapter of this data sheet for the absolute address assignments for all KBI 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.

Refer to the direct-page register summary in the Memory chapter of this data sheet for the absolute address assignments for all SCI 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.

11.2.1 SCI Baud Rate Registers (SCIxBDH, SCIxBDL)

This pair of registers controls the prescale divisor for SCI baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to SCIxBDH to buffer the high half of the new value and then write to SCIxBDL. The working value in SCIxBDH does not change until SCIxBDL is written.

SCIxBDL is reset to a non-zero value, so after reset the baud rate generator remains disabled until the first time the receiver or transmitter is enabled (RE or TE bits in SCIxC2 are written to 1).

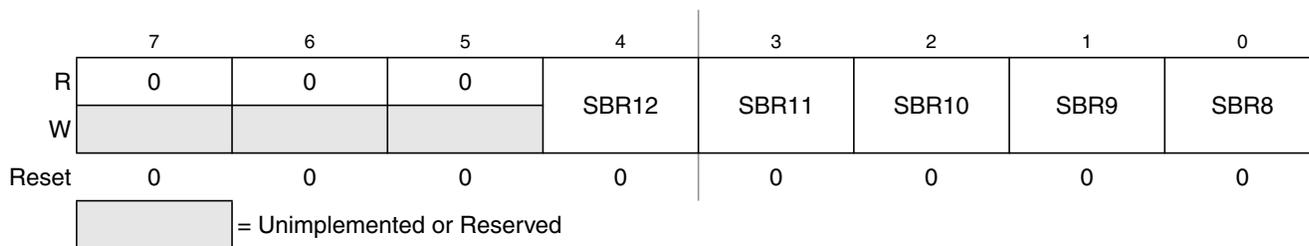


Figure 11-4. SCI Baud Rate Register (SCIxBDH)

Table 11-1. SCIxBDH Register Field Descriptions

| Field | Description |
|------------------|--|
| 4:0 SBR[12:8] | Baud Rate Modulo Divisor — These 13 bits are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16×BR). See also BR bits in Table 11-2. |

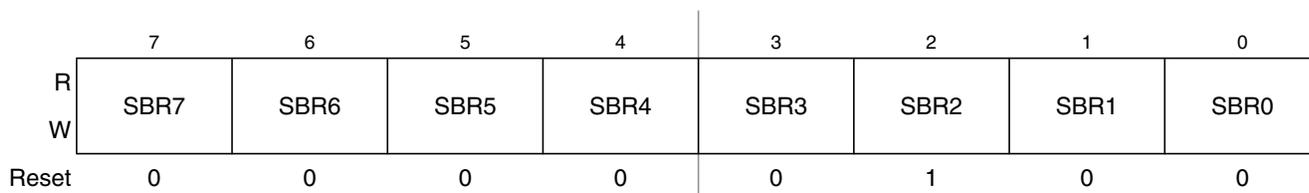


Figure 11-5. SCI Baud Rate Register (SCIxBDL)

Table 11-2. SCIxBDL Register Field Descriptions

| Field | Description |
|------------------|--|
| 4:0 SBR[12:8] | Baud Rate Modulo Divisor — These 13 bits are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16×BR). See also BR bits in Table 11-1. |

Table 11-5. SCIxS1 Register Field Descriptions (continued)

| Field | Description |
|-----------|---|
| 5 RDRF | <p>Receive Data Register Full Flag — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SCIxD). In 8-bit mode, to clear RDRF, read SCIxS1 with RDRF = 1 and then read the SCI data register (SCIxD). In 9-bit mode, to clear RDRF, read SCIxS1 with RDRF = 1 and then read SCIxD and the SCI control 3 register (SCIxC3). SCIxD and SCIxC3 can be read in any order, but the flag is cleared only after both data registers are read.</p> <p>0 Receive data register empty. 1 Receive data register full.</p> |
| 4 IDLE | <p>Idle Line Flag — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line.</p> <p>To clear IDLE, read SCIxS1 with IDLE = 1 and then read the SCI data register (SCIxD). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE will get set only once even if the receive line remains idle for an extended period.</p> <p>0 No idle line detected. 1 Idle line was detected.</p> |
| 3 OR | <p>Receiver Overrun Flag — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SCIxD yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SCIxD. To clear OR, read SCIxS1 with OR = 1 and then read the SCI data register (SCIxD).</p> <p>0 No overrun. 1 Receive overrun (new SCI data lost).</p> |
| 2 NF | <p>Noise Flag — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF will be set at the same time as the flag RDRF gets set for the character. To clear NF, read SCIxS1 and then read the SCI data register (SCIxD).</p> <p>0 No noise detected. 1 Noise detected in the received character in SCIxD.</p> |
| 1 FE | <p>Framing Error Flag — FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bit was expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read SCIxS1 with FE = 1 and then read the SCI data register (SCIxD).</p> <p>0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.</p> |
| 0 PF | <p>Parity Error Flag — PF is set at the same time as RDRF when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, read SCIxS1 and then read the SCI data register (SCIxD).</p> <p>0 No parity error. 1 Parity error.</p> |

to Section 11.3.5.1, “8- and 9-Bit Data Modes.” For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full ($RDRF = 1$), it gets the data from the receive data register by reading SCIxD. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user’s program that handles receive data. Refer to Section 11.3.4, “Interrupts and Status Flags” for more details about flag clearing.

11.3.3.1 Data Sampling Technique

The SCI receiver uses a $16\times$ baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The $16\times$ baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise flag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

11.3.3.2 Receiver Wakeup Operation

Receiver wakeup is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the first

Table 12-1. SPI1C1 Field Descriptions (continued)

| Field | Description |
|------------|--|
| 4 MSTR | Master/Slave Mode Select 0 SPI module configured as a slave SPI device 1 SPI module configured as a master SPI device |
| 3 CPOL | Clock Polarity — This bit effectively places an inverter in series with the clock signal from a master SPI or to a slave SPI device. Refer to Section 12.4.1, “SPI Clock Formats” for more details. 0 Active-high SPI clock (idles low) 1 Active-low SPI clock (idles high) |
| 2 CPHA | Clock Phase — This bit selects one of two clock formats for different kinds of synchronous serial peripheral devices. Refer to Section 12.4.1, “SPI Clock Formats” for more details. 0 First edge on SPSCK occurs at the middle of the first cycle of an 8-cycle data transfer 1 First edge on SPSCK occurs at the start of the first cycle of an 8-cycle data transfer |
| 1 SSOE | Slave Select Output Enable — This bit is used in combination with the mode fault enable (MODFEN) bit in SPCR2 and the master/slave (MSTR) control bit to determine the function of the \overline{SS} pin as shown in Table 12-2. |
| 0 LSBFE | LSB First (Shifter Direction) 0 SPI serial data transfers start with most significant bit 1 SPI serial data transfers start with least significant bit |

Table 12-2. \overline{SS} Pin Function

| MODFEN | SSOE | Master Mode | Slave Mode |
|--------|------|--------------------------------------|--------------------|
| 0 | 0 | General-purpose I/O (not SPI) | Slave select input |
| 0 | 1 | General-purpose I/O (not SPI) | Slave select input |
| 1 | 0 | \overline{SS} input for mode fault | Slave select input |
| 1 | 1 | Automatic \overline{SS} output | Slave select input |

NOTE

Ensure that the SPI should not be disabled (SPE=0) at the same time as a bit change to the CPHA bit. These changes should be performed as separate operations or unexpected behavior may occur.

12.3.2 SPI Control Register 2 (SPI1C2)

This read/write register is used to control optional features of the SPI system. Bits 7, 6, 5, and 2 are not implemented and always read 0.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|---|---|---|--------|---------|---|---------|------|
| R | 0 | 0 | 0 | MODFEN | BIDIROE | 0 | SPISWAI | SPC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Figure 12-6. SPI Control Register 2 (SPI1C2)

Table 12-5. SPI Baud Rate Prescaler Divisor

| SPPR2:SPPR1:SPPR0 | Prescaler Divisor |
|-------------------|-------------------|
| 0:0:0 | 1 |
| 0:0:1 | 2 |
| 0:1:0 | 3 |
| 0:1:1 | 4 |
| 1:0:0 | 5 |
| 1:0:1 | 6 |
| 1:1:0 | 7 |
| 1:1:1 | 8 |

Table 12-6. SPI Baud Rate Divisor

| SPR2:SPR1:SPR0 | Rate Divisor |
|----------------|--------------|
| 0:0:0 | 2 |
| 0:0:1 | 4 |
| 0:1:0 | 8 |
| 0:1:1 | 16 |
| 1:0:0 | 32 |
| 1:0:1 | 64 |
| 1:1:0 | 128 |
| 1:1:1 | 256 |

12.3.4 SPI Status Register (SPI1S)

This register has three read-only status bits. Bits 6, 3, 2, 1, and 0 are not implemented and always read 0. Writes have no meaning or effect.

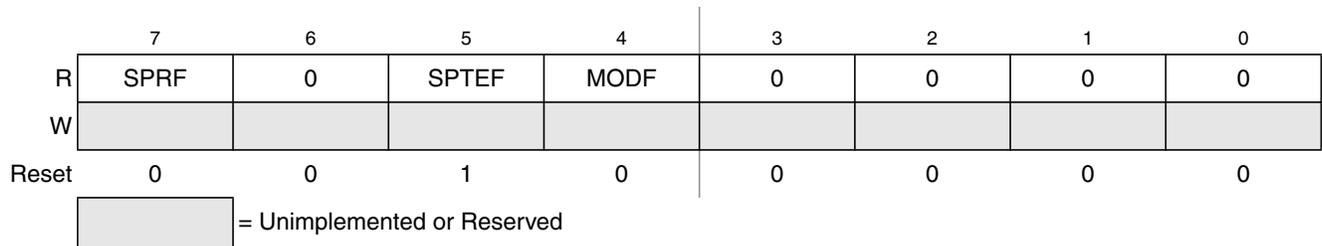


Figure 12-8. SPI Status Register (SPI1S)

Table 14-11. APCTL3 Register Field Descriptions (continued)

| Field | Description |
|-------------|--|
| 1 ADPC17 | ADC Pin Control 17 — ADPC17 is used to control the pin associated with channel AD17. 0 AD17 pin I/O control enabled 1 AD17 pin I/O control disabled |
| 0 ADPC16 | ADC Pin Control 16 — ADPC16 is used to control the pin associated with channel AD16. 0 AD16 pin I/O control enabled 1 AD16 pin I/O control disabled |

14.5 Functional Description

The ADC module is disabled during reset or when the ADCH bits are all high. The module is idle when a conversion has completed and another conversion has not been initiated. When idle, the module is in its lowest power state.

The ADC can perform an analog-to-digital conversion on any of the software selectable channels. The selected channel voltage is converted by a successive approximation algorithm into an 11-bit digital result. In 8-bit mode, the selected channel voltage is converted by a successive approximation algorithm into a 9-bit digital result.

When the conversion is completed, the result is placed in the data registers (ADC1RH and ADC1RL). In 10-bit mode, the result is rounded to 10 bits and placed in ADC1RH and ADC1RL. In 8-bit mode, the result is rounded to 8 bits and placed in ADC1RL. The conversion complete flag (COCO) is then set and an interrupt is generated if the conversion complete interrupt has been enabled (AIEN = 1).

The ADC module has the capability of automatically comparing the result of a conversion with the contents of its compare registers. The compare function is enabled by setting the ACFE bit and operates in conjunction with any of the conversion modes and configurations.

14.5.1 Clock Select and Divide Control

One of four clock sources can be selected as the clock source for the ADC module. This clock source is then divided by a configurable value to generate the input clock to the converter (ADCK). The clock is selected from one of the following sources by means of the ADICLK bits.

- The bus clock, which is equal to the frequency at which software is executed. This is the default selection following reset.
- The bus clock divided by 2. For higher bus clock rates, this allows a maximum divide by 16 of the bus clock.
- ALTCLK, as defined for this MCU (See module section introduction).
- The asynchronous clock (ADACK) – This clock is generated from a clock source within the ADC module. When selected as the clock source this clock remains active while the MCU is in wait or stop3 mode and allows conversions in these modes for lower noise operation.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC will not perform according to specifications. If the available clocks

A.7 Supply Current Characteristics

Table A-8. Supply Current Characteristics

| Num | C | Parameter | Symbol | V _{DD} (V) | Typ ¹ | Max | Unit | Temp (°C) |
|-----|---|--|------------------------|------------------------|------------------|------------|------|-----------------------------|
| 1 | P | Run supply current ² measured at (CPU clock = 2 MHz, f _{Bus} = 1 MHz) | R _I DD | 5 | 0.750 | 0.950 | mA | -40 to 125°C |
| | | | | 3 | 0.570 | 0.770 | | |
| 2 | P | Run supply current ² measured at (CPU clock = 16 MHz, f _{Bus} = 8 MHz) | R _I DD | 5 | 4.90 | 5.10 | mA | -40 to 125°C |
| | | | | 3 | 3.50 | 3.70 | | |
| 3 | P | Run supply current ³ measured at (CPU clock = 40 MHz, f _{Bus} = 20 MHz) | R _I DD | 5 | 16.8 | 18.5 | mA | -40 to 125°C |
| | | | | 3 | 11.5 | 12.5 | | |
| 4 | P | Stop2 mode supply current | S2 _I DD | 5 | 0.900 | 18.0 60 | μA | -40 to 85°C -40 to 125°C |
| | | | | 3 | 0.720 | 17.0 50 | μA | -40 to 85°C -40 to 125°C |
| 5 | P | Stop3 mode supply current | S3 _I DD | 5 | 0.975 | 20.0 90 | μA | -40 to 85°C -40 to 125°C |
| | | | | 3 | 0.825 | 19.0 85 | μA | -40 to 85°C -40 to 125°C |
| 6 | C | RTI adder to stop2 or stop3 ⁴ | S23 _I DDRTI | 5 | 300 | 500 500 | nA | -40 to 85°C -40 to 125°C |
| | | | | 3 | 300 | 500 500 | nA | -40 to 85°C -40 to 125°C |
| 7 | C | LVD adder to stop3 (LVDE = LVDSE = 1) | S3 _I DDLVD | 5 | 110 | 180 | μA | -40 to 125°C |
| | | | | 3 | 90 | 160 | μA | -40 to 125°C |
| 8 | C | Adder to stop3 for oscillator enabled (OSCSTEN = 1) ⁵ | S3 _I DDOSC | 5,3 | 5 | 8 | μA | -40 to 125°C |

¹ Typical values are based on characterization data at 25°C unless otherwise stated. See Figure A-5 through Figure A-7 for typical curves across voltage/temperature.

² All modules except ADC enabled, but not active. ICG configured for FBE. Does not include any DC loads on port pins.

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

⁴ Most customers are expected to find that auto-wakeup from stop2 or stop3 can be used instead of the higher current wait mode. Wait mode typical is 500 μA at 5 V with f_{Bus} = 1 MHz.

⁵ Values given under the following conditions: low range operation (RANGE = 0) with a 32.768 kHz crystal, low power mode (HGO = 0), clock monitor disabled (LOCD = 1).

Appendix B

Ordering Information and Mechanical Drawings

B.1 Ordering Information

This section contains ordering numbers for MC9S08AW60 Series devices. See below for an example of the device numbering system.

Table B-1. Consumer and Industrial Device Numbering System

| Device Number ¹ | Memory | | Available Packages ² |
|----------------------------|--------|------|---------------------------------|
| | FLASH | RAM | Type |
| MC9S08AW60 | 63,280 | 2048 | 64-pin LQFP |
| MC9S08AW48 | 49,152 | | 64-pin QFP |
| MC9S08AW32 | 32,768 | | 48-pin QFN |
| MC9S08AW16 | 16,384 | 1024 | 44-pin LQFP |

¹ See Table 1-1 for a complete description of modules included on each device.

² See Table B-3 for package information.

Table B-2. Automotive Device Numbering System

| Device Number ¹ | Memory | | Available Packages ² |
|----------------------------|--------|------|---------------------------------|
| | FLASH | RAM | Type |
| S9S08AW60 | 63,280 | 2048 | 64-pin LQFP |
| S9S08AW48 | 49,152 | | 48-pin QFN |
| S9S08AW32 | 32,768 | | 44-pin LQFP |
| S9S08AW16 | 16,384 | 1024 | 48-pin QFN 44-pin LQFP |

¹ See Table 1-1 for a complete description of modules included on each device.

² See Table B-3 for package information.