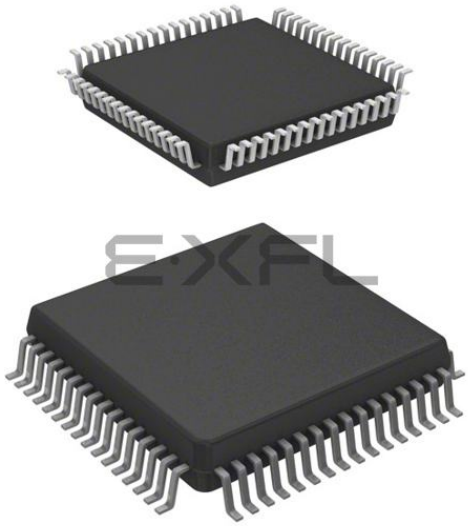

Welcome to E-XFL.COM

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	HC08
Core Size	8-Bit
Speed	8.4MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	52
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 15x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908az60avfuer

List of Chapters

Chapter 4
FLASH-1 Memory

Chapter 5
FLASH-2 Memory

Chapter 6
EEPROM-1 Memory

Table 1-1. External Pins Summary (Continued)

Pin Name	Function	Driver Type	Hysteresis[1]	Reset State
PTF0/TACH2	General-Purpose I/O Timer A Channel 2	Dual State	Yes	Input Hi-Z
PTG2/KBD2–PTG0/KBD0	General-Purpose I/O/ Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
PTH1/KBD4 –PTH0/KBD3	General-Purpose I/O/ Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
V_{DD}	Chip Power Supply	N/A	N/A	N/A
V_{SS}	Chip Ground	N/A	N/A	N/A
V_{DDA}	CGM Analog Power Supply			
V_{SSA}	CGM Analog Ground			
V_{DDAREF}	ADC Power Supply	N/A	N/A	N/A
A_{VSS}/V_{REFL}	ADC Ground/ ADC Reference Low Voltage	N/A	N/A	N/A
V_{REFH}	A/D Reference High Voltage	N/A	N/A	N/A
OSC1	External Clock In	N/A	No	Input Hi-Z
OSC2	External Clock Out	N/A	N/A	Output
CGMXFC	PLL Loop Filter Cap	N/A	N/A	N/A
$\overline{IRQ}$	External Interrupt Request	N/A	N/A	Input Hi-Z
$\overline{RST}$	Reset	N/A	N/A	Output Low
CANRx	CAN Serial Input	N/A	Yes	Input Hi-Z
CANTx	CAN Serial Output	Output	No	Output
BDRxD	BDLC Serial Input	N/A	Yes	Input Hi-Z
BDTxD	BDLC Serial Output	Output	No	Output

1. Hysteresis is not 100% tested but is typically a minimum of 300 mV.

	MC68HC908AZ60A / MC68HC908AS60A	
$FE03	SIM BREAK FLAG CONTROL REGISTER (SBFCR)	$FE03
$FE04	RESERVED	$FE04
$FE05	RESERVED	$FE05
$FE06	RESERVED	$FE06
$FE07	RESERVED	$FE07
$FE08	FLASH-2 CONTROL REGISTER (FL2CR)	$FE08
$FE09	CONFIGURATION WRITE-ONCE REGISER (CONFIG-2)	$FE09
$FE0A	RESERVED	$FE0A
$FE0B	RESERVED	$FE0B
$FE0C	BREAK ADDRESS REGISTER HIGH (BRKH)	$FE0C
$FE0D	BREAK ADDRESS REGISTER LOW (BRKL)	$FE0D
$FE0E	BREAK STATUS AND CONTROL REGISTER (BSCR)	$FE0E
$FE0F	LVI STATUS REGISTER (LVISR)	$FE0F
$FE10	EEPROM-1EEDIVH NONVOLATILE REGISTER(EE1DIVHNVR)	$FE10
$FE11	EEPROM-1EEDIVL NONVOLATILE REGISTER(EE1DIVLNVR)	$FE11
$FE12	RESERVED	$FE12
$FE13	RESERVED	$FE13
$FE14	RESERVED	$FE14
$FE15	RESERVED	$FE15
$FE16	RESERVED	$FE16
$FE17	RESERVED	$FE17
$FE18	RESERVED	$FE18
$FE19	RESERVED	$FE19
$FE1A	EEPROM-1 EE DIVIDER HIGH REGISTER(EE1DIVH)	$FE1A
$FE1B	EEPROM-1 EE DIVIDER LOW REGISTER(EE1DIVL)	$FE1B
$FE1C	EEPROM-1 EEPROM NONVOLATILE REGISTER (EE1NVR)	$FE1C
$FE1D	EEPROM-1 EEPROM CONTROL REGISTER (EE1CR)	$FE1D
$FE1E	RESERVED	$FE1E
$FE1F	EEPROM-1 EEPROM ARRAY CONFIGURATION REGISTER (EE1ACR)	$FE1F
$FE20 ↓ $FF1F	MONITOR ROM 256BYTES	$FE20 ↓ $FF1F
$FF20 ↓ $FF6F	UNIMPLEMENTED 80 BYTES	$FF20 ↓ $FF6F
$FF70	EEPROM-2 EEDIVH NONVOLATILE REGISTER (EE2DIVHNVR)	$FF70
$FF71	EEPROM-2 EEDIVL NONVOLATILE REGISTER (EE2DIVLNVR)	$FF71
$FF72	RESERVED	$FF72
$FF73	RESERVED	$FF73
$FF74	RESERVED	$FF74

Figure 2-1. Memory Map (Sheet 2 of 3)

2.2 I/O Section

Addresses $0000–$004F, shown in Figure 2-2, contain the I/O Data, Status, and Control Registers.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
$0002	Port C Data Register (PTC)	Read:	0	0	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:	R	R						
$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
$0006	Data Direction Register C (DDRC)	Read:	MCLKEN	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:		R						
$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDR2	DDRD1	DDRD0
		Write:								
$0008	Port E Data Register (PTE)	Read:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
$0009	Port F Data Register (PTF)	Read:	0	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
		Write:	R							
$000A	Port G Data Register (PTG)	Read:	0	0	0	0	0	PTG2	PTG1	PTG0
		Write:	R	R	R	R	R			
$000B	Port H Data Register (PTH)	Read:	0	0	0	0	0	0	PTH1	PTH0
		Write:	R	R	R	R	R	R		
$000C	Data Direction Register E (DDRE)	Read:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
$000D	Data Direction Register F (DDRF)	Read:	0	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
		Write:	R							
$000E	Data Direction Register G (DDRG)	Read:	0	0	0	0	0	DDRG2	DDRG1	DDRG0
		Write:	R	R	R	R	R			
$000F	Data Direction Register H (DDRH)	Read:	0	0	0	0	0	0	DDRH1	DDRH0
		Write:	R	R	R	R	R	R		

= Unimplemented R = Reserved

Figure 2-2. I/O Data, Status and Control Registers (Sheet 1 of 5)

AUTO — Automatic Termination of Program/Erase Cycle

When AUTO is set, EEPGM is cleared automatically after the program/erase cycle is terminated by the internal timer.

(See note D for 6.4.5.2 EEPROM-1 Programming, 6.4.5.3 EEPROM-1 Erasing, and 28.1.13 EEPROM Memory Characteristics)

1 = Automatic clear of EEPGM is enabled
0 = Automatic clear of EEPGM is disabled

EEPGM — EEPROM-1 Program/Erase Enable

This read/write bit enables the internal charge pump and applies the programming/erasing voltage to the EEPROM-1 array if the EELAT bit is set and a write to a valid EEPROM-1 location has occurred. Reset clears the EEPGM bit.

1 = EEPROM-1 programming/erasing power switched on
0 = EEPROM-1 programming/erasing power switched off

NOTE

Writing logic 0s to both the EELAT and EEPGM bits with a single instruction will clear EEPGM only to allow time for the removal of high voltage.

6.5.2 EEPROM-1 Array Configuration Register

The EEPROM-1 array configuration register configures EEPROM-1 security and EEPROM-1 block protection.

This read-only register is loaded with the contents of the EEPROM-1 nonvolatile register (EE1NVR) after a reset.

Address: $FE1F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	UNUSED	UNUSED	UNUSED	EEPRTCT	EEBP3	EEBP2	EEBP1	EEBP0
Write:								
Reset:	Contents of EE1NVR ($FE1C)							

= Unimplemented

Figure 6-3. EEPROM-1 Array Configuration Register (EE1ACR)

Bit 7:5 — Unused Bits

These read/write bits are software programmable but have no functionality.

EEPRTCT — EEPROM-1 Protection Bit

The EEPRTCT bit is used to enable the security feature in the EEPROM (see EEPROM-1 Program/Erase Protection).

1 = EEPROM-1 security disabled
0 = EEPROM-1 security enabled

This feature is a write-once feature. Once the protection is enabled it may not be disabled.

7.4.5 EEPROM-2 Programming and Erasing

The unprogrammed or erase state of an EEPROM bit is a logic 1. The factory default for all bytes within the EEPROM-2 array is $FF.

The programming operation changes an EEPROM bit from logic 1 to logic 0 (programming cannot change a bit from logic 0 to a logic 1). In a single programming operation, the minimum EEPROM programming size is one bit; the maximum is eight bits (one byte).

The erase operation changes an EEPROM bit from logic 0 to logic 1. In a single erase operation, the minimum EEPROM erase size is one byte; the maximum is the entire EEPROM-2 array.

The EEPROM can be programmed such that one or multiple bits are programmed (written to a logic 0) at a time. However, the user may never program the same bit location more than once before erasing the entire byte. In other words, the user is not allowed to program a logic 0 to a bit that is already programmed (bit state is already logic 0).

For some applications it might be advantageous to track more than 10K events with a single byte of EEPROM by programming one bit at a time. For that purpose, a special selective bit programming technique is available. An example of this technique is illustrated in Table 7-2.

Table 7-2. Example Selective Bit Programming Description

Description	Program Data in Binary	Result in Binary
Original state of byte (erased)	n/a	1111:1111
First event is recorded by programming bit position 0	1111:1110	1111:1110
Second event is recorded by programming bit position 1	1111:1101	1111:1100
Third event is recorded by programming bit position 2	1111:1011	1111:1000
Fourth event is recorded by programming bit position 3	1111:0111	1111:0000
Events five through eight are recorded in a similar fashion		

NOTE

None of the bit locations are actually programmed more than once although the byte was programmed eight times.

When this technique is utilized, a program/erase cycle is defined as multiple program sequences (up to eight) to a unique location followed by a single erase operation.

7.4.5.1 Program/Erase Using AUTO Bit

An additional feature available for EEPROM-2 program and erase operations is the AUTO mode. When enabled, AUTO mode will activate an internal timer that will automatically terminate the program/erase cycle and clear the EEPGM bit. Please see 7.4.5.2 EEPROM-2 Programming, 7.4.5.3 EEPROM-2 Erasing, and 7.5.1 EEPROM-2 Control Register for more information.

7.4.5.2 EEPROM-2 Programming

The unprogrammed or erase state of an EEPROM bit is a logic 1. Programming changes the state to a logic 0. Only EEPROM bytes in the non-protected blocks and the EE2NVR register can be programmed.

Use the following procedure to program a byte of EEPROM:

1. Clear EERAS1 and EERAS0 and set EELAT in the EE2CR.(A)

NOTE

If using the AUTO mode, also set the AUTO bit during Step 1.

2. Write the desired data to the desired EEPROM address.(B)
3. Set the EEPGM bit.(C) Go to Step 7 if AUTO is set.
4. Wait for time, t_{EEPGM}, to program the byte.
5. Clear EEPGM bit.
6. Wait for time, t_{EEFPV}, for the programming voltage to fall. Go to Step 8.
7. Poll the EEPGM bit until it is cleared by the internal timer.(D)
8. Clear EELAT bits.(E)

NOTE

A. *EERAS1 and EERAS0 must be cleared for programming. Setting the EELAT bit configures the address and data buses to latch data for programming the array. Only data with a valid EEPROM-2 address will be latched. If EELAT is set, other writes to the EE2CR will be allowed after a valid EEPROM-2 write.*

B. *If more than one valid EEPROM write occurs, the last address and data will be latched overriding the previous address and data. Once data is written to the desired address, do not read EEPROM-2 locations other than the written location. (Reading an EEPROM location returns the latched data and causes the read address to be latched).*

C. *The EEPGM bit cannot be set if the EELAT bit is cleared or a non-valid EEPROM address is latched. This is to ensure proper programming sequence. Once EEPGM is set, do not read any EEPROM-2 locations; otherwise, the current program cycle will be unsuccessful. When EEPGM is set, the on-board programming sequence will be activated.*

D. *The delay time for the EEPGM bit to be cleared in AUTO mode is less than t_{EEPGM}. However, on other MCUs, this delay time may be different. For forward compatibility, software should not make any dependency on this delay time.*

E. *Any attempt to clear both EEPGM and EELAT bits with a single instruction will only clear EEPGM. This is to allow time for removal of high voltage from the EEPROM-2 array.*

16.6.2 Stop Mode

With the LVISTOP and LVIPWR bits in the configuration register programmed to a logic 1, the LVI module will be active after a STOP instruction. Because CPU clocks are disabled during stop mode, the LVI trip must bypass the digital filter to generate a reset and bring the MCU out of stop.

With the LVIPWR bit in the configuration register programmed to logic 1 and the LVISTOP bit at a logic 0, the LVI module will be inactive after a STOP instruction.

NOTE

The LVI feature is intended to provide the safe shutdown of the microcontroller and thus protection of related circuitry prior to any application V_{DD} voltage collapsing completely to an unsafe level. It is not intended that users operate the microcontroller at lower than specified operating voltage V_{DD}.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Control Register 3 (SCC3)	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
	Write:								
	Reset:	U	U	0	0	0	0	0	0
SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
	Write:								
	Reset:	1	1	0	0	0	0	0	0
SCI Status Register 2 (SCS2)	Read:	0	0	0	0	0	0	BKF	RPF
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
	Write:	T7	T6	T5	T4	T3	T2	T1	T0
	Reset:	Unaffected by Reset							
SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
	Write:								
	Reset:	0	0	0	0	0	0	0	0

= Unimplemented U = Unaffected R = Reserved

Figure 18-7. SCI I/O Receiver Register Summary

Table 18-4. SCI Receiver I/O Address Summary

Register	SCC1	SCC2	SCC3	SCS1	SCS2	SCDR	SCBR
Address	$0013	$0014	$0015	$0016	$0017	$0018	$0019

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Table 18-5 summarizes the results of the start bit verification samples.

Table 18-5. Start Bit Verification

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 18-6 summarizes the results of the data bit samples.

Table 18-6. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit.

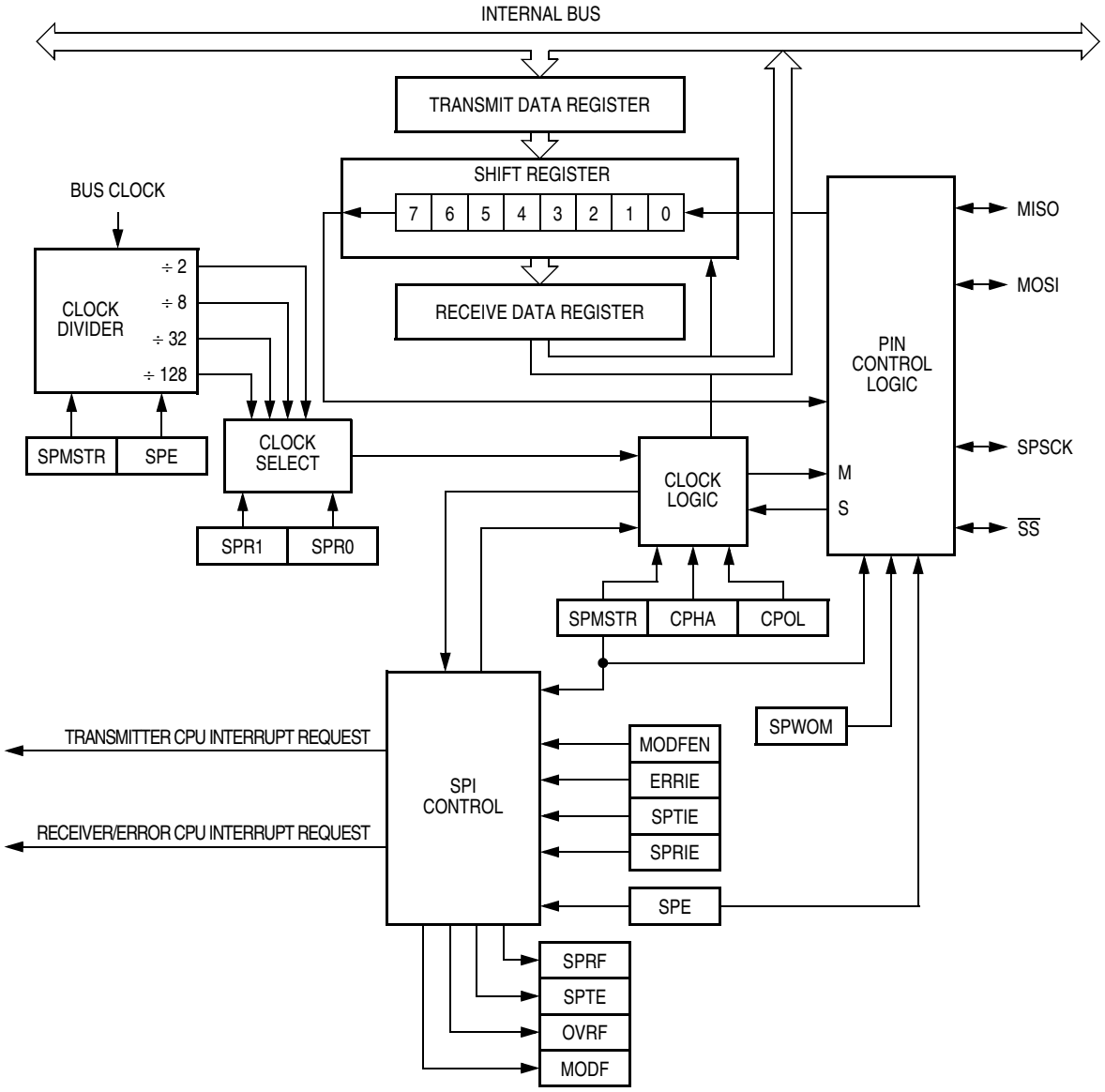

Figure 19-2. SPI Module Block Diagram

22.2 Port A

Port A is an 8-bit general-purpose bidirectional I/O port.

22.2.1 Port A Data Register

The port A data register contains a data latch for each of the eight port A pins.

Address:	$0000							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by Reset							

Figure 22-2. Port A Data Register (PTA)

PTA[7:0] — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

22.2.2 Data Direction Register A

Data direction register A determines whether each port A pin is an input or an output. Writing a logic 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a logic 0 disables the output buffer.

Address:	$0004							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 22-3. Data Direction Register A (DDRA)

DDRA[7:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all port A pins as inputs.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE

Avoid glitches on port A pins by writing to the port A data register before changing data direction register A bits from 0 to 1.

Figure 22-4 shows the port A I/O logic.

22.8 Port G

Port G is a 3-bit special function port that shares all of its pins with the keyboard interrupt module (KBD). Note that Port G is only available on 64-pin package options.

22.8.1 Port G Data Register

The port G data register contains a data latch for each of the three port G pins.

Address: $000A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	PTG2	PTG1	PTG0
Write:	R	R	R	R	R			
Reset:	Unaffected by Reset							
Alternative Function:						KBD2	KBD1	KBD0

R = Reserved

Figure 22-20. Port G Data Register (PTG)

PTG[2:0] — Port G Data Bits

These read/write bits are software programmable. Data direction of each port G pin is under the control of the corresponding bit in data direction register G. Reset has no effect on PTG[2:0].

KBD[2:0] — Keyboard Wakeup pins

The keyboard interrupt enable bits, KBIE[2:0], in the keyboard interrupt control register, enable the port G pins as external interrupt pins (See Chapter 24 Keyboard Module (KBI)). Enabling an external interrupt pin will override the corresponding DDRGx.

22.8.2 Data Direction Register G

Data direction register G determines whether each port G pin is an input or an output. Writing a logic 1 to a DDRG bit enables the output buffer for the corresponding port G pin; a logic 0 disables the output buffer.

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	DDRG2	DDRG1	DDRG0
Write:	R	R	R	R	R			
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 22-21. Data Direction Register G (DDRG)

In Power Down mode, no registers can be accessed.

MSCAN08 bus activity can wake the MCU from CPU Stop/MSCAN08 power-down mode. However, until the oscillator starts up and synchronisation is achieved the MSCAN08 will not respond to incoming data.

23.8.4 CPU Wait Mode

The MSCAN08 module remains active during CPU wait mode. The MSCAN08 will stay synchronized to the CAN bus and generates transmit, receive, and error interrupts to the CPU, if enabled. Any such interrupt will bring the MCU out of wait mode.

23.8.5 Programmable Wakeup Function

The MSCAN08 can be programmed to apply a low-pass filter function to the RxCAN input line while in internal sleep mode (see information on control bit WUPM in 23.13.2 MSCAN08 Module Control Register 1). This feature can be used to protect the MSCAN08 from wake-up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic inference within noisy environments.

23.9 Timer Link

The MSCAN08 will generate a timer signal whenever a valid frame has been received. Because the CAN specification defines a frame to be valid if no errors occurred before the EOF field has been transmitted successfully, the timer signal will be generated right after the EOF. A pulse of one bit time is generated. As the MSCAN08 receiver engine also receives the frames being sent by itself, a timer signal also will be generated after a successful transmission.

The previously described timer signal can be routed into the on-chip timer interface module (TIM).This signal is connected to the timer n channel m input[1] under the control of the timer link enable (TLNKEN) bit in the CMCR0.

After timer n has been programmed to capture rising edge events, it can be used under software control to generate 16-bit time stamps which can be stored with the received message.

23.10 Clock System

Figure 23-7 shows the structure of the MSCAN08 clock generation circuitry and its interaction with the clock generation module (CGM). With this flexible clocking scheme the MSCAN08 is able to handle CAN bus rates ranging from 10 kbps up to 1 Mbps.

1. The timer channel being used for the timer link is integration dependent.

25.8.4 TIMA Channel Status and Control Registers

Each of the TIMA channel status and control registers:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare or PWM operation
- Selects high, low or toggling output on output compare
- Selects rising edge, falling edge or any edge as the active input capture trigger
- Selects output toggling on TIMA overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Register Name and Address: TASC0 — $0026

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Register Name and Address: TASC1 — $0029

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0		R					
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Register Name and Address: TASC2 — $002C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	TOV2	CH2MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Register Name and Address: TASC3 — $002F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH3F	CH3IE	0	MS3A	ELS3B	ELS3A	TOV3	CH3MAX
Write:	0		R					
Reset:	0	0	0	0	0	0	0	0

Register Name and Address: TASC4 — $0032

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	TOV4	CH4MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 25-7. TIMA Channel Status and Control Registers (TASC0–TASC5)

26.3.2 Voltage Conversion

When the input voltage to the ADC equals V_{REFH} (see 28.1.6 ADC Characteristics), the ADC converts the signal to $FF (full scale). If the input voltage equals $V_{SSA,}$ the ADC converts it to $00. Input voltages between V_{REFH} and V_{SSA} are a straight-line linear conversion. Conversion accuracy of all other input voltages is not guaranteed. Avoid current injection on unused ADC inputs to prevent potential conversion error.

NOTE

Input voltage should not exceed the analog supply voltages.

26.3.3 Conversion Time

Conversion starts after a write to the ADSCR (ADC status control register, $0038), and requires between 16 and 17 ADC clock cycles to complete. Conversion time in terms of the number of bus cycles is a function of ADICLK select, CGMXCLK frequency, bus frequency, and ADIV prescaler bits. For example, with a CGMXCLK frequency of 4 MHz, bus frequency of 8 MHz, and fixed ADC clock frequency of 1 MHz, one conversion will take between 16 and 17 μs and there will be between 128 bus cycles between each conversion. Sample rate is approximately 60 kHz.

Refer to 28.1.6 ADC Characteristics.

$$\text{Conversion Time} = \frac{\text{16 to 17 ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

$$\text{Number of Bus Cycles} = \text{Conversion Time} \times \text{Bus Frequency}$$

26.3.4 Continuous Conversion

In the continuous conversion mode, the ADC data register will be filled with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit (ADC status control register, $0038) is cleared. The COCO bit is set after the first conversion and will stay set for the next several conversions until the next write of the ADC status and control register or the next read of the ADC data register.

26.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes. See 28.1.6 ADC Characteristics for accuracy information.

26.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit (ADC status control register, $0038) is at logic 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

28.1.2 Functional Operating Range

Rating	Symbol	Value	Unit
Operating Temperature Range(1)	T_A	–40 to T_A(MAX)	°C
Operating Voltage Range	V_{DD}	5.0 ± 0.5	V

1. T_A(MAX) = 125°C for part suffix MFU/MFN
   T_A(MAX) = 105°C for part suffix VFU/VFN
   T_A(MAX) = 85°C for part suffix CFU/CFN

NOTE

For applications which use the LVI, Freescale guarantees the functionality of the device down to the LVI trip point (V_{LVI}) within the constraints outlined in Chapter 16 Low-Voltage Inhibit (LVI).

28.1.3 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance QFP (64 Pins)	θ_{JA}	70	°C/W
Thermal Resistance PLCC (52 Pins)	θ_{JA}	50	°C/W
I/O Pin Power Dissipation	$P_{I/O}$	User Determined	W
Power Dissipation (see Note 1)	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273\ °C)$	W
Constant (see Note 2)	K	$P_D \times (T_A + 273\ °C) + (P_D^2 \times \theta_{JA})$	W/°C
Average Junction Temperature	T_J	$T_A + P_D \times \theta_{JA}$	°C

1. Power dissipation is a function of temperature.
2. K is a constant unique to the device. K can be determined from a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.

parity — An error-checking scheme that counts the number of logic 1s in each byte transmitted. In a system that uses odd parity, every byte is expected to have an odd number of logic 1s. In an even parity system, every byte should have an even number of logic 1s. In the transmitter, a parity generator appends an extra bit to each byte to make the number of logic 1s odd for odd parity or even for even parity. A parity checker in the receiver counts the number of logic 1s in each byte. The parity checker generates an error signal if it finds a byte with an incorrect number of logic 1s.

PC — See “program counter (PC).”

peripheral — A circuit not under direct CPU control.

phase-locked loop (PLL) — A oscillator circuit in which the frequency of the oscillator is synchronized to a reference signal.

PLL — See "phase-locked loop (PLL)."

pointer — Pointer register. An index register is sometimes called a pointer register because its contents are used in the calculation of the address of an operand, and therefore points to the operand.

polarity — The two opposite logic levels, logic 1 and logic 0, which correspond to two different voltage levels, V_{DD} and V_{SS}.

polling — Periodically reading a status bit to monitor the condition of a peripheral device.

port — A set of wires for communicating with off-chip devices.

prescaler — A circuit that generates an output signal related to the input signal by a fractional scale factor such as 1/2, 1/8, 1/10 etc.

program — A set of computer instructions that cause a computer to perform a desired operation or operations.

program counter (PC) — A 16-bit register in the CPU08. The PC register holds the address of the next instruction or operand that the CPU will use.

pull — An instruction that copies into the accumulator the contents of a stack RAM location. The stack RAM address is in the stack pointer.

pullup — A transistor in the output of a logic gate that connects the output to the logic 1 voltage of the power supply.

pulse-width — The amount of time a signal is on as opposed to being in its off state.

pulse-width modulation (PWM) — Controlled variation (modulation) of the pulse width of a signal with a constant frequency.

push — An instruction that copies the contents of the accumulator to the stack RAM. The stack RAM address is in the stack pointer.

PWM period — The time required for one complete cycle of a PWM waveform.

RAM — Random access memory. All RAM locations can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

RC circuit — A circuit consisting of capacitors and resistors having a defined time constant.

read — To copy the contents of a memory location to the accumulator.

register — A circuit that stores a group of bits.

reserved memory location — A memory location that is used only in special factory test modes. Writing to a reserved location has no effect. Reading a reserved location returns an unpredictable value.

reset — To force a device to a known condition.

ROM — Read-only memory. A type of memory that can be read but cannot be changed (written). The contents of ROM must be specified before manufacturing the MCU.

SCI — See "serial communication interface module (SCI)."

serial — Pertaining to sequential transmission over a single line.

serial communications interface module (SCI) — A module in the M68HC08 Family that supports asynchronous communication.

serial peripheral interface module (SPI) — A module in the M68HC08 Family that supports synchronous communication.

set — To change a bit from logic 0 to logic 1; opposite of clear.

shift register — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

signed — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See "serial peripheral interface module (SPI)."

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU08 containing the address of the next available storage location on the stack.

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

TIM — See "timer interface module (TIM)."

timer interface module (TIM) — A module used to relate events in a system to a point in time.

timer — A module used to relate events in a system to a point in time.