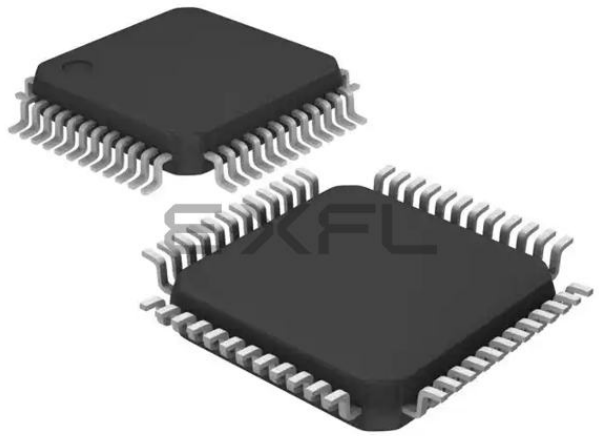


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	37
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gz32cfae

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
April, 2004	N/A	Initial release	N/A
May, 2004	1.0	9.7.3 Keyboard Interrupt Polarity Register — Corrected the bit description of the KBIP7–KBIP0 bits.	119
		14.8.8 ESCI Prescaler Register — Reworked note under PDS2–PDS0 description for clarity.	212
		Table 22-1. MC Order Numbers — Corrected order numbers.	329
		Figure 22-1. Device Numbering System — Reworked diagram to reflect correct order numbers.	329
		Table A-1. MC Order Numbers — Corrected order numbers.	342
		Figure A-3. Device Numbering System — Reworked diagram to reflect correct order numbers.	342
		B.4 Ordering Information — Corrected order numbers.	346
		Figure B-3. Device Numbering System — Reworked diagram to reflect correct order numbers.	346
June, 2005	2.0	Reformatted to Freescale publication standards	Throughout
		Table 14-6. ESCI LIN Control Bits — Corrected Functionality entries	211
		14.9.1 ESCI Arbiter Control Register — Corrected bit ACLK bit description	215
		14.9.3 Bit Time Measurement — Corrected definition for ACLK bit	216
March, 2006	3.0	10.5 Clock Generator Module (CGM) — Updated description to remove erroneous information.	122
July, 2006	4.0	Added section 1.5.15 Unused Pin Termination	31
		Chapter 13 Input/Output (I/O) Ports — Replaced note	169
		Table 14-6. ESCI LIN Control Bits — Updated functionality column.	213
		18.6 TIM1 During Break Interrupts — Updated first paragraph for clarity.	270
		19.6 TIM2 During Break Interrupts — Updated first paragraph for clarity.	290
		20.2.1.2 TIM During Break Interrupts — Updated first paragraph for clarity.	302
		Figure 20-10. Normal Monitor Mode Circuit and Figure 20-11. Forced Monitor Mode — Changed capacitor values	307
		21.5 5.0-Vdc Electrical Characteristics — Updated minimum value for low-voltage inhibit, trip rising voltage (VTRIPR).	317
		21.9.2 CGM Component Information — Updated values for feedback bias resistor	322

Revision History (Continued)

Date	Revision Level	Description	Page Number(s)
October, 2006	5.0	12.2 Features — Corrected timer link connection from TIM2 channel 0 to TIM1 channel 0.	135
		12.9 Timer Link — Corrected timer link connection from TIM2 channel 0 to TIM1 channel 0.	147
		21.5 5.0-Vdc Electrical Characteristics and 21.6 3.3-Vdc Electrical Characteristics — Updated DC injection current specification.	317 319
April, 2007	6.0	Figure 2-2. Control, Status, and Data Registers — Changed TBMCLKSEL to TMBCLKSEL to be compatible with development tool nomenclature	37
		Chapter 5 Configuration Register (CONFIG) — Changed COPCLK to CGMXCLK and TBMCLKSEL to TMBCLKSEL to be compatible with development tool nomenclature	91 92 93
		10.6.2 Stop Mode — Changed COPCLK to CGMXCLK	125
		Figure 14-3. ESCI Module Block Diagram — Changed BUS_CLK to BUS CLOCK and removed reference to 4xBUSCLK	192
		14.4.2 Transmitter — Changed ESCIBDSRC to SCIBDSRC	194
		14.9.1 ESCI Arbiter Control Register and 14.9.3 Bit Time Measurement — Replaced one quarter with one half in the definition for ACLK = 1	217 218
		Figure 17-1. Timebase Block Diagram, 17.5 TBM Interrupt Rate, and Table 17-1. Timebase Divider Selection — Changed TBMCLKSEL to TMBCLKSEL to be compatible with development tool nomenclature	260 261
		21.9 Clock Generation Module (CGM) Characteristics — Updated section to include the following: 21.9.1 CGM Operating Conditions 21.9.2 CGM Component Information 21.9.3 CGM Acquisition/Lock Time Information	 322 322 323

Chapter 18
Timer Interface Module (TIM1)

Chapter 19
Timer Interface Module (TIM2)

Chapter 20
Development Support

Chapter 21
Electrical Specifications

1.5.13 Port F I/O Pins (PTF7/T2CH5–PTF0)

PTF7–PTF4 are special-function, bidirectional I/O port pins that can be individually programmed to be timer interface module (TIM2) pins.

PTF3–PTF0 are general-purpose, bidirectional I/O port pins that contain higher current sink/source capability.

PTF7–PTF0 are only available on the 64-pin QFP package. See Chapter 18 Timer Interface Module (TIM1), Chapter 19 Timer Interface Module (TIM2), and Chapter 13 Input/Output (I/O) Ports.

1.5.14 Port G I/O Pins (PTG7/AD23–PTBG0/AD16)

PTG7–PTG0 are general-purpose, bidirectional I/O port pins that can also be used for analog-to-digital converter (ADC) inputs. PTG7–PTG0 are only available on the 64-pin QFP package. See Chapter 13 Input/Output (I/O) Ports and Chapter 3 Analog-to-Digital Converter (ADC).

1.5.15 Unused Pin Termination

Input pins and I/O port pins that are not used in the application must be terminated. This prevents excess current caused by floating inputs, and enhances immunity during noise or transient events. Termination methods include:

1. Configuring unused pins as outputs and driving high or low;
2. Configuring unused pins as inputs and enabling internal pull-ups;
3. Configuring unused pins as inputs and using external pull-up or pull-down resistors.

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

Since some general-purpose I/O pins are not available on all packages, these pins must be terminated as well. Either method 1 or 2 above are appropriate.

3.8.2 ADC Data Register High and Data Register Low

3.8.2.1 Left Justified Mode

In left justified mode, the ADRH register holds the eight MSBs of the 10-bit result. The ADRL register holds the two LSBs of the 10-bit result. All other bits read as 0. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.

Address:	$003D							ADRH
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							
Address:	$003E							ADRL
Read:	AD1	AD0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 3-5. ADC Data Register High (ADRH) and Low (ADRL)

3.8.2.2 Right Justified Mode

In right justified mode, the ADRH register holds the two MSBs of the 10-bit result. All other bits read as 0. The ADRL register holds the eight LSBs of the 10-bit result. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.

Address:	$003D							ADRH
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	AD9	AD8
Write:								
Reset:	Unaffected by reset							
Address:	$003E							ADRL
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 3-6. ADC Data Register High (ADRH) and Low (ADRL)

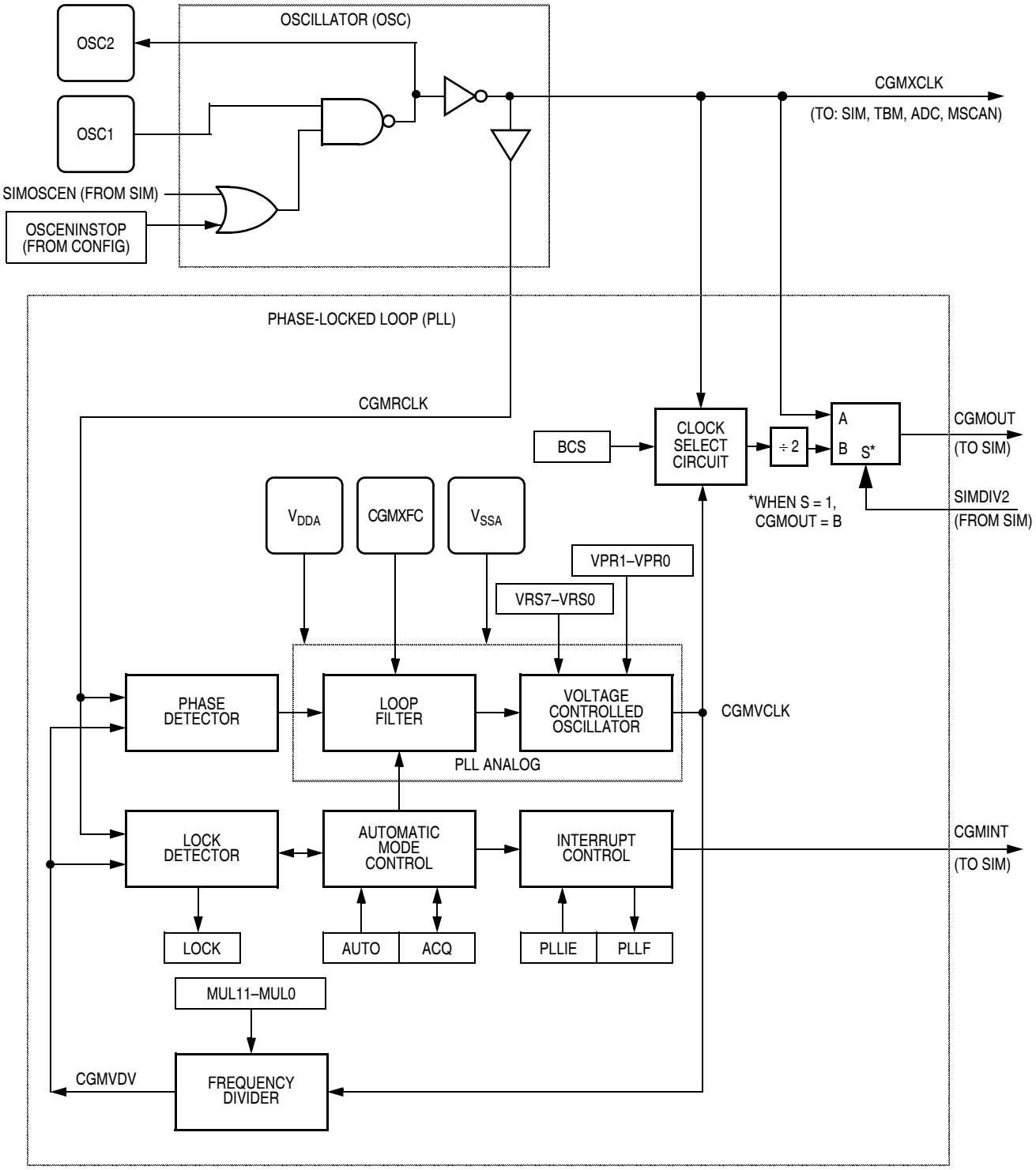


Figure 4-1. CGM Block Diagram

4.4.8 Crystal Output Frequency Signal (CGMXCLK)

CGMXCLK is the crystal oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. Figure 4-2 shows only the logical relation of CGMXCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of CGMXCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of CGMXCLK can be unstable at start up.

4.4.9 CGM Base Clock Output (CGMOUT)

CGMOUT is the clock output of the CGM. This signal goes to the SIM, which generates the MCU clocks. CGMOUT is a 50 percent duty cycle clock running at twice the bus frequency. CGMOUT is software programmable to be either the oscillator output, CGMXCLK, divided by two or the VCO clock, CGMVCLK, divided by two.

4.4.10 CGM CPU Interrupt (CGMINT)

CGMINT is the interrupt signal generated by the PLL lock detector.

4.5 CGM Registers

These registers control and monitor operation of the CGM:

- PLL control register (PCTL)
  (See 4.5.1 PLL Control Register.)
- PLL bandwidth control register (PBWC)
  (See 4.5.2 PLL Bandwidth Control Register.)
- PLL multiplier select register high (PMSH)
  (See 4.5.3 PLL Multiplier Select Register High.)
- PLL multiplier select register low (PMSL)
  (See 4.5.4 PLL Multiplier Select Register Low.)
- PLL VCO range select register (PMRS)
  (See 4.5.5 PLL VCO Range Select Register.)

Figure 4-3 is a summary of the CGM registers.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0036	PLL Control Register (PCTL) See page 83.	Read:	PLLIE	PLLF	PLLON	BCS	R	R	VPR1	VPR0
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0037	PLL Bandwidth Control Register (PBWC) See page 85.	Read:	AUTO	LOCK	$\overline{\text{ACQ}}$	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0038	PLL Multiplier Select High Register (PMSH) See page 86.	Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
			= Unimplemented			R	= Reserved			

Figure 4-3. CGM I/O Register Summary

7.3.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to 1. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	V	1	1	H	I	N	Z	C
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

Figure 7-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE

To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

Chapter 9
Keyboard Interrupt Module (KBI)

9.1 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA7. When a port pin is enabled for keyboard interrupt function, an internal pullup/pulldown device is also enabled on the pin.

9.2 Features

Features include:

- Eight keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level- interrupt sensitivity
- Edge detect programmable for rising or falling edges
- Level detect programmable for high or low levels
- Exit from low-power modes
- Pullup/pulldown device automatically configured based on polarity of edge/level selection

9.3 Functional Description

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin also enables its internal pullup/pulldown device. On falling edge or low level selection a pullup device is configured. On rising edge or high level selection a pulldown device is configured.

- A falling edge is detected when an enabled keyboard input signal is seen as a 1 (the deasserted level) during one bus cycle and then a 0 (the asserted level) during the next cycle.
- A rising edge is detected when the input signal is seen as a 0 during one bus cycle and then a 1 during the next cycle.

A keyboard interrupt is latched when one or more keyboard pins are asserted. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

10.3 Break Module (BRK)

10.3.1 Wait Mode

The break (BRK) module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if the SBSW bit in the break status register is set.

10.3.2 Stop Mode

The break module is inactive in stop mode. The STOP instruction does not affect break module register states.

10.4 Central Processor Unit (CPU)

10.4.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

10.4.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

10.5 Clock Generator Module (CGM)

10.5.1 Wait Mode

The clock generator module (CGM) remains active in wait mode. Before entering wait mode, software can disengage and turn off the PLL by clearing the BCS and PLLON bits in the PLL control register (PCTL). Less power-sensitive applications can disengage the PLL without turning it off. Applications that require the PLL to wake the MCU from wait mode also can deselect the PLL output without turning off the PLL.

10.5.2 Stop Mode

If the OSCENINSTOP bit in the CONFIG2 register is cleared (default), then the STOP instruction disables the CGM (oscillator and phase-locked loop) and holds low all CGM outputs (CGMXCLK, CGMOUT, and CGMINT).

If the OSCENINSTOP bit in the CONFIG2 register is set, then the phase locked loop is shut off, but the oscillator will continue to operate in stop mode.

completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme would de-couple the re-loading of the transmit buffers from the actual message being sent and as such reduces the reactiveness requirements on the CPU. Problems may arise if the sending of a message would be finished just while the CPU re-loads the second buffer. In that case, no buffer would then be ready for transmission and the bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN08 has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN08 implements with the “local priority” concept described in 12.4.2 Receive Structures.

12.4.2 Receive Structures

The received messages are stored in a 2-stage input first in first out (FIFO). The two message buffers are mapped using a "ping pong" arrangement into a single memory area (see Figure 12-3). While the background receive buffer (RxBG) is exclusively associated to the MSCAN08, the foreground receive buffer (RxFG) is addressable by the central processor unit (CPU08). This scheme simplifies the handler software, because only one address area is applicable for the receive process.

Both buffers have a size of 13 bytes to store the CAN control bits, the identifier (standard or extended), and the data content. For details, see 12.12 Programmer’s Model of Message Storage.

The receiver full flag (RXF) in the MSCAN08 receiver flag register (CRFLG), signals the status of the foreground receive buffer. When the buffer contains a correctly received message with matching identifier, this flag is set. See 12.13.5 MSCAN08 Receiver Flag Register (CRFLG)

On reception, each message is checked to see if it passes the filter (for details see 12.5 Identifier Acceptance Filter) and in parallel is written into RxBG. The MSCAN08 copies the content of RxBG into RxFG[1], sets the RXF flag, and generates a receive interrupt to the CPU[2]. The user’s receive handler has to read the received message from RxFG and to reset the RXF flag to acknowledge the interrupt and to release the foreground buffer. A new message which can follow immediately after the IFS field of the CAN frame, is received into RxBG. The overwriting of the background buffer is independent of the identifier filter function.

When the MSCAN08 module is transmitting, the MSCAN08 receives its own messages into the background receive buffer, RxBG. It does NOT overwrite RxFG, generate a receive interrupt or acknowledge its own messages on the CAN bus. The exception to this rule is in loop-back mode (see 12.13.2 MSCAN08 Module Control Register 1), where the MSCAN08 treats its own messages exactly like all other incoming messages. The MSCAN08 receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN08 must be prepared to become the receiver.

An overrun condition occurs when both the foreground and the background receive message buffers are filled with correctly received messages with accepted identifiers and another message is correctly received from the bus with an accepted identifier. The latter message will be discarded and an error interrupt with overrun indication will be generated if enabled. The MSCAN08 is still able to transmit messages with both receive message buffers filled, but all incoming messages are discarded.

1. Only if the RXF flag is not set.
2. The receive interrupt will occur only if not masked. A polling scheme can be applied on RXF also.

14.8.2 ESCI Control Register 2

ESCI control register 2 (SCC2):

- Enables these CPU interrupt requests:
  - SCTE bit to generate transmitter CPU interrupt requests
  - TC bit to generate transmitter CPU interrupt requests
  - SCRF bit to generate receiver CPU interrupt requests
  - IDLE bit to generate receiver CPU interrupt requests
- Enables the transmitter
- Enables the receiver
- Enables ESCI wakeup
- Transmits ESCI break characters

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 14-11. ESCI Control Register 2 (SCC2)

SCTIE — ESCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate ESCI transmitter CPU interrupt requests. Setting the SCTIE bit in SCC2 enables the SCTE bit to generate CPU interrupt requests. Reset clears the SCTIE bit.

1 = SCTE enabled to generate CPU interrupt
0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate ESCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

1 = TC enabled to generate CPU interrupt requests
0 = TC not enabled to generate CPU interrupt requests

SCRIE — ESCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate ESCI receiver CPU interrupt requests. Setting the SCRIE bit in SCC2 enables the SCRF bit to generate CPU interrupt requests. Reset clears the SCRIE bit.

1 = SCRF enabled to generate CPU interrupt
0 = SCRF not enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables the IDLE bit to generate ESCI receiver CPU interrupt requests. Reset clears the ILIE bit.

1 = IDLE enabled to generate CPU interrupt requests
0 = IDLE not enabled to generate CPU interrupt requests

15.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in Figure 15-3. This clock originates from either an external oscillator or from the on-chip PLL.

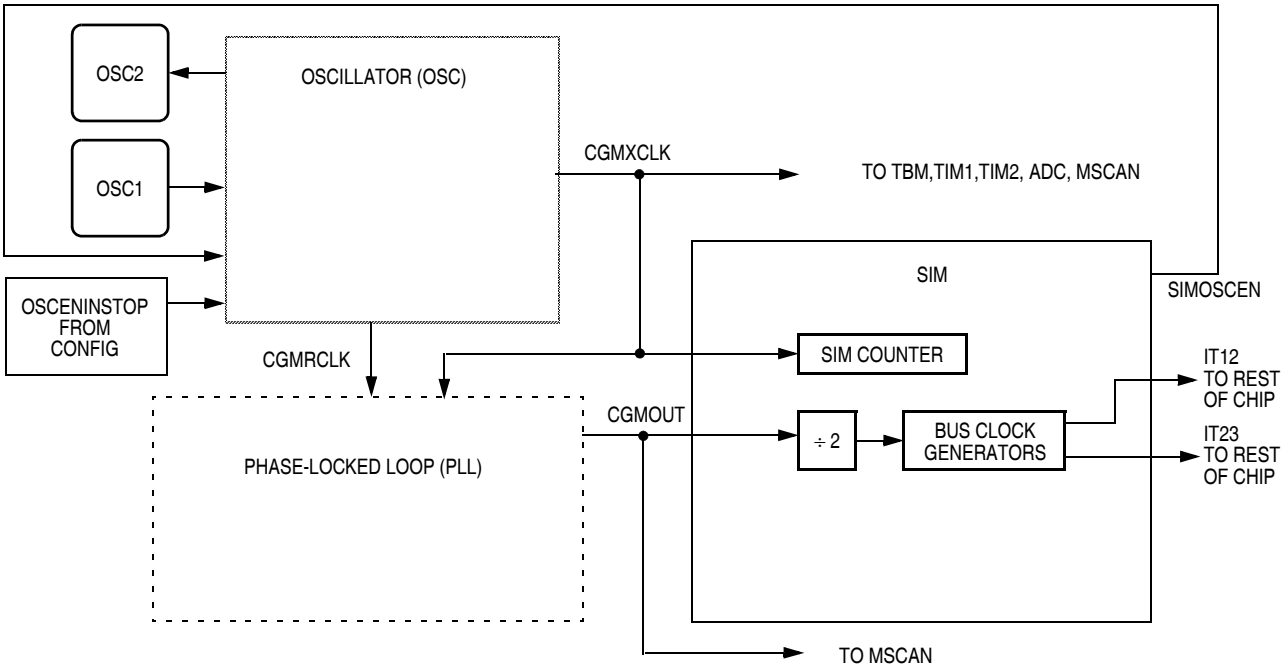


Figure 15-3. System Clock Signals

15.2.1 Bus Timing

In user mode, the internal bus frequency is either the crystal oscillator output (CGMXCLK) divided by four or the PLL output (CGMVCLK) divided by four.

15.2.2 Clock Startup from POR or LVI Reset

When the power-on reset module or the low-voltage inhibit module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after the 4096 CGMXCLK cycle POR timeout has completed. The $\overline{\text{RST}}$ pin is driven low by the SIM during this entire period. The bus clocks start upon completion of the timeout.

15.2.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt or reset, the SIM allows CGMXCLK to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay timeout. This timeout is selectable as 4096 or 32 CGMXCLK cycles. See 15.6.2 Stop Mode.

In wait mode, the CPU clocks are inactive. The SIM also produces two sets of clocks for other modules. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

Table 20-7. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high-byte:low-byte order
Opcode	$0C
Command Sequence	

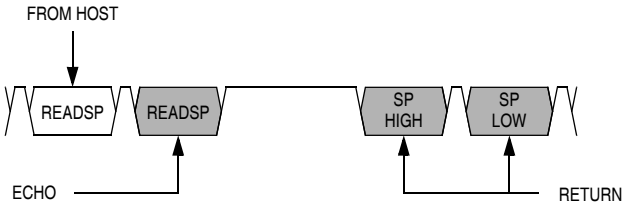
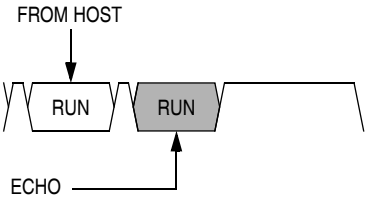


Table 20-8. RUN (Run User Program) Command

Description	Executes PULH and RTI instructions
Operand	None
Data Returned	None
Opcode	$28
Command Sequence	



The MCU executes the SWI and PSHH instructions when it enters monitor mode. The RUN command tells the MCU to execute the PULH and RTI instructions. Before sending the RUN command, the host can modify the stacked CPU registers to prepare to run the host program. The READSP command returns the incremented stack pointer value, SP + 1. The high and low bytes of the program counter are at addresses SP + 5 and SP + 6.

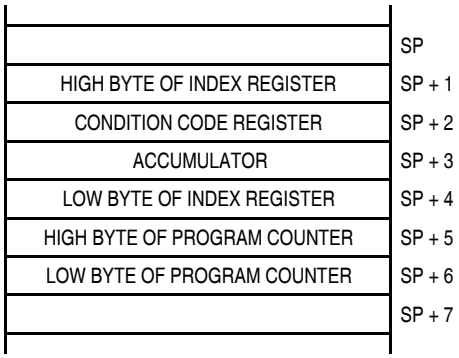
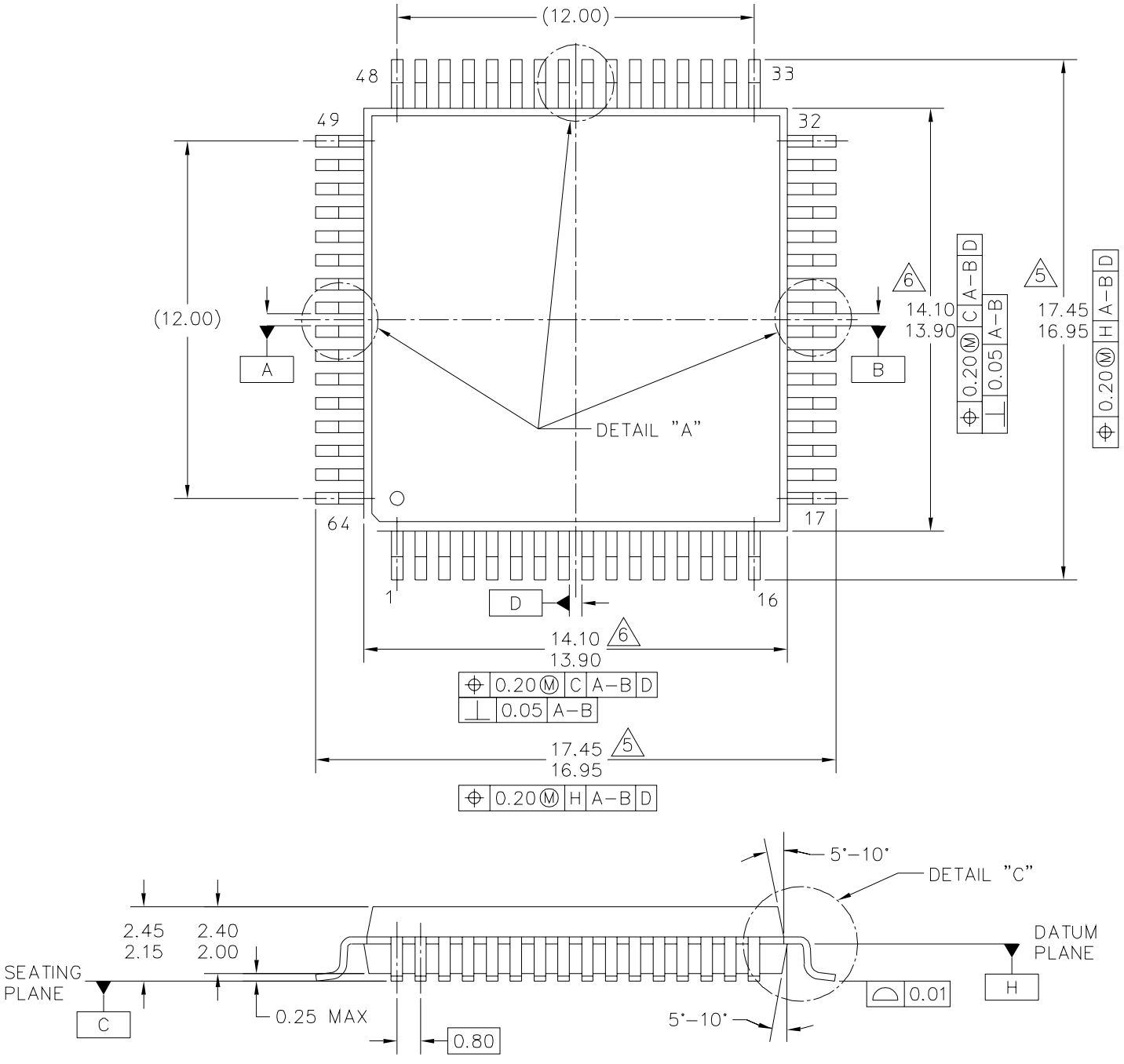
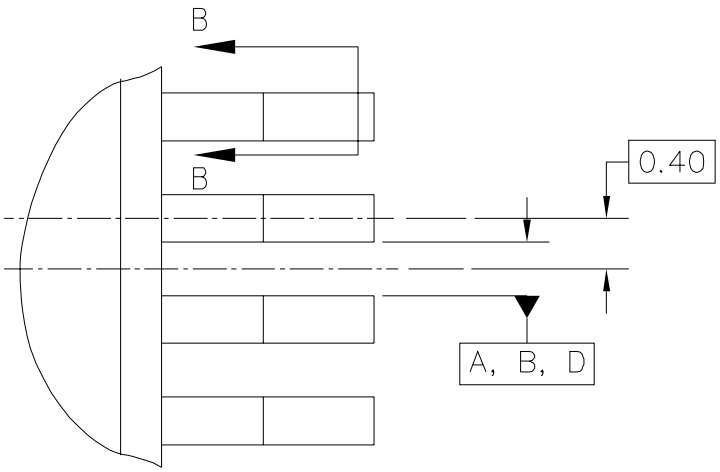


Figure 20-16. Stack Pointer at Monitor Mode Entry

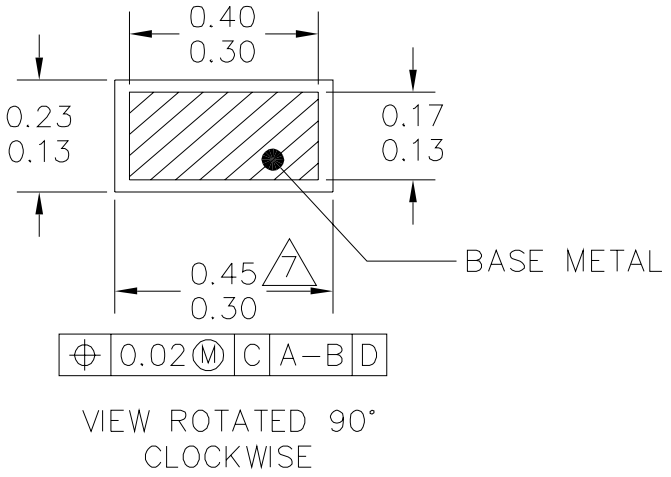


MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42844B	REV: A
64LD QFP (14 X 14)	CASE NUMBER: 840B-02	06 APR 2005
	STANDARD: NON-JEDEC	

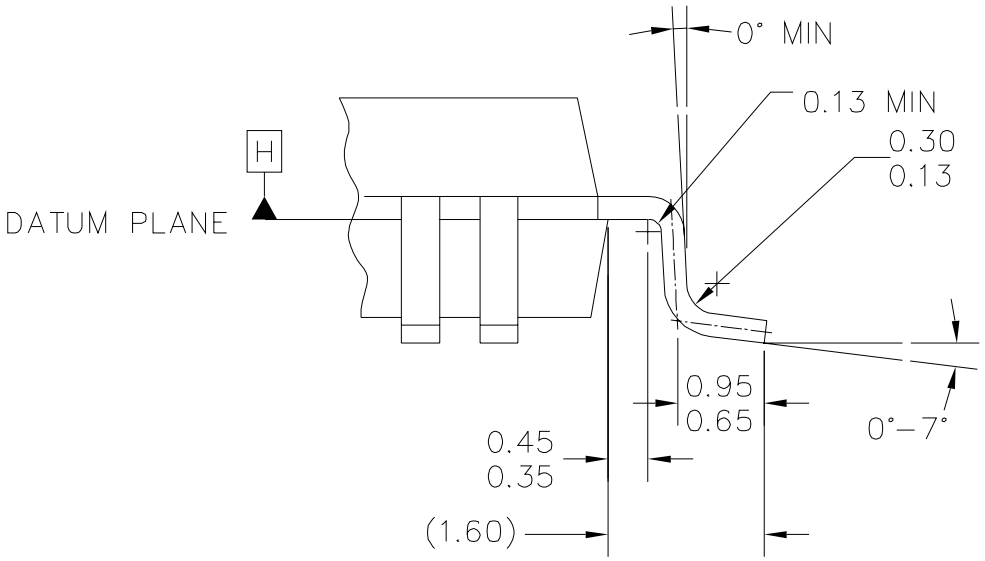


DETAIL "A"



VIEW ROTATED 90° CLOCKWISE

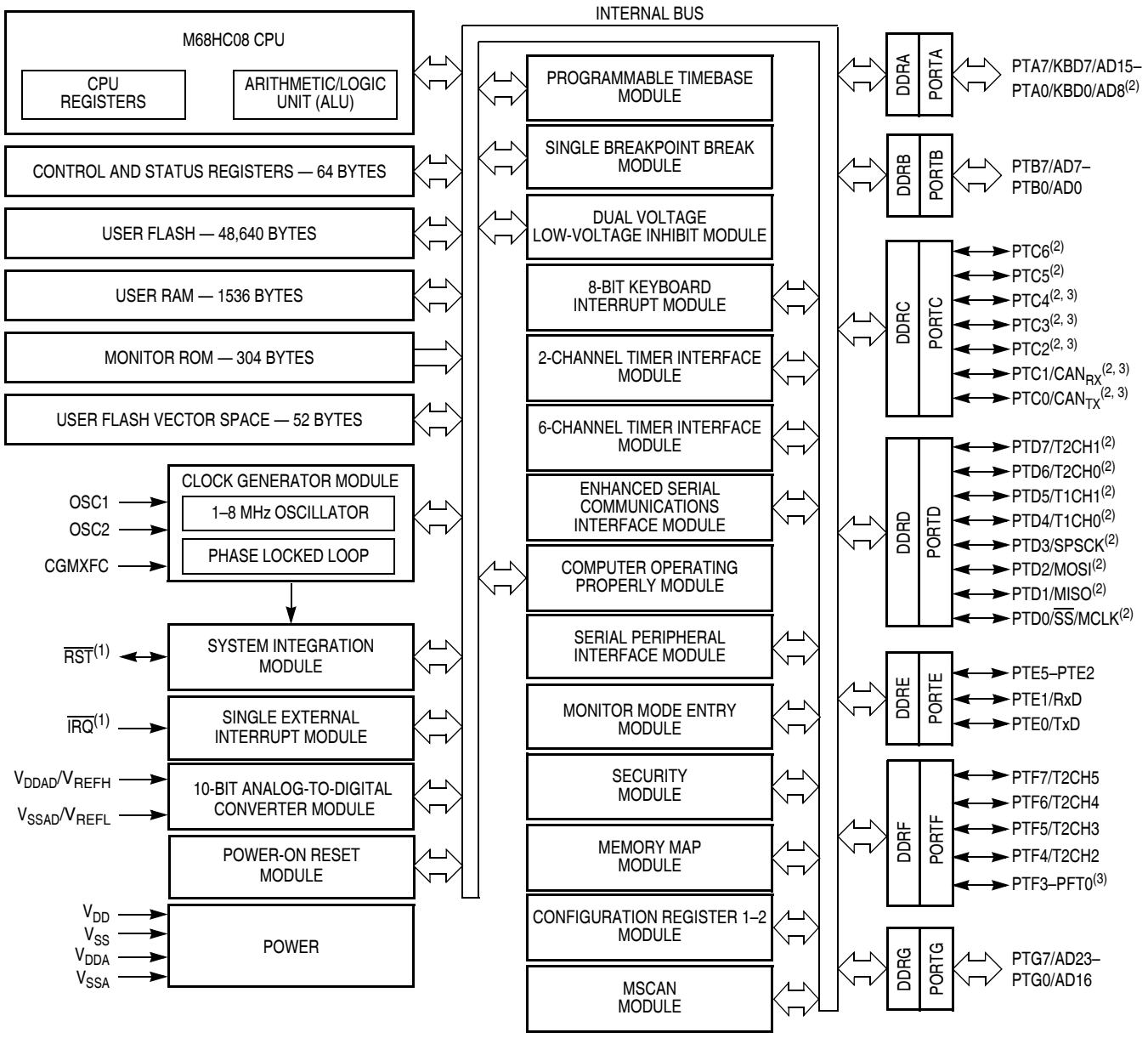
SECTION B–B



DETAIL "C"

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42844B	REV: A
64LD QFP (14 X 14)	CASE NUMBER: 840B-02	06 APR 2005
	STANDARD: NON-JEDEC	



1. Pin contains integrated pullup device.
2. Ports are software configurable with pullup device if input port or pullup/pulldown device for keyboard input.
3. Higher current drive port pins

Figure A-1. MC68HC908GZ48 Block Diagram