Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	33MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	357-BBGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/kmc68360vr33l

pins are the only data pins used. Refer to Section 4 Bus Operation for information on the data bus and its relationship to bus operation.

2.1.3.2 DATA BUS (D15–D0). These pins can function as 16 additional data pins used in long-word and 3-byte transfers. They are three-stated and not used if the QUICC is configured into 16-bit bus mode.

2.1.4 Parity

These three-state bidirectional signals provide parity generation/checking for the data path between the QUICC or external masters and other devices. There are four parity lines—one for every eight data bits. The parity lines consists of two groups. Refer to Section 6 System Integration Module (SIM60) for more information on parity generation/checking.

2.1.4.1 PARITY (PRTY0). This pin is the parity value for data bits 31–24.

2.1.4.2 PARITY (PRTY1). This pin is the parity value for data bits 23–16.

2.1.4.3 PARITY (PRTY2). This pin is the parity value for data bits 15–8.

2.1.4.4 PARITY (PRTY3). This pin has two functions. During total system reset, it is the $\overline{16BM}$ pin to determine whether 16-bit data bus mode is to be enabled. After system reset, it functions as the parity line 3.

PRTY3—This pin is the parity value for data bits 0–7.

$\overline{16BM}$—This pin selects the 16-bit data bus mode. To choose a 32-bit data bus during total system reset, this pin can be left floating (it has an internal pullup resistor) or can be driven/pulled high. To choose a 16-bit data bus during total system reset, this pin should be driven/pulled low.

2.1.5 Memory Controller

The following signals are used to control an external memory device.

2.1.5.1 CHIP SELECT/ROW ADDRESS SELECT ($\overline{CS6}$–$\overline{CS0}$/$\overline{RAS6}$–$\overline{RAS0}$). The chip-select output signals enable peripherals or memory arrays at programmed addresses. CS0 is the global chip select for the boot ROM containing the user's reset vector and initialization program. Refer to Section 6 System Integration Module (SIM60) for more information on chip selects.

NOTE

In addition, $\overline{RAS1}$ can be simultaneously output on the $\overline{RAS1DD}$ pin to increase the $\overline{RAS1}$ line drive capability, and $\overline{RAS2}$ can be simultaneously output on the $\overline{RAS2DD}$ pin to increase the $\overline{RAS2}$ line drive capability.

2.1.5.2 CHIP SELECT/ROW ADDRESS SELECT/INTERRUPT ACKNOWLEDGE ($\overline{CS7}$/$\overline{RAS7}$/$\overline{IACK7}$). This pin can be programmed as a $\overline{CS7}$/$\overline{RAS7}$ pin or as the $\overline{IACK7}$ line. See Section 6 System Integration Module (SIM60) for more information on this selection.

Table 3-3. QUICC SIM Registers Memory Map

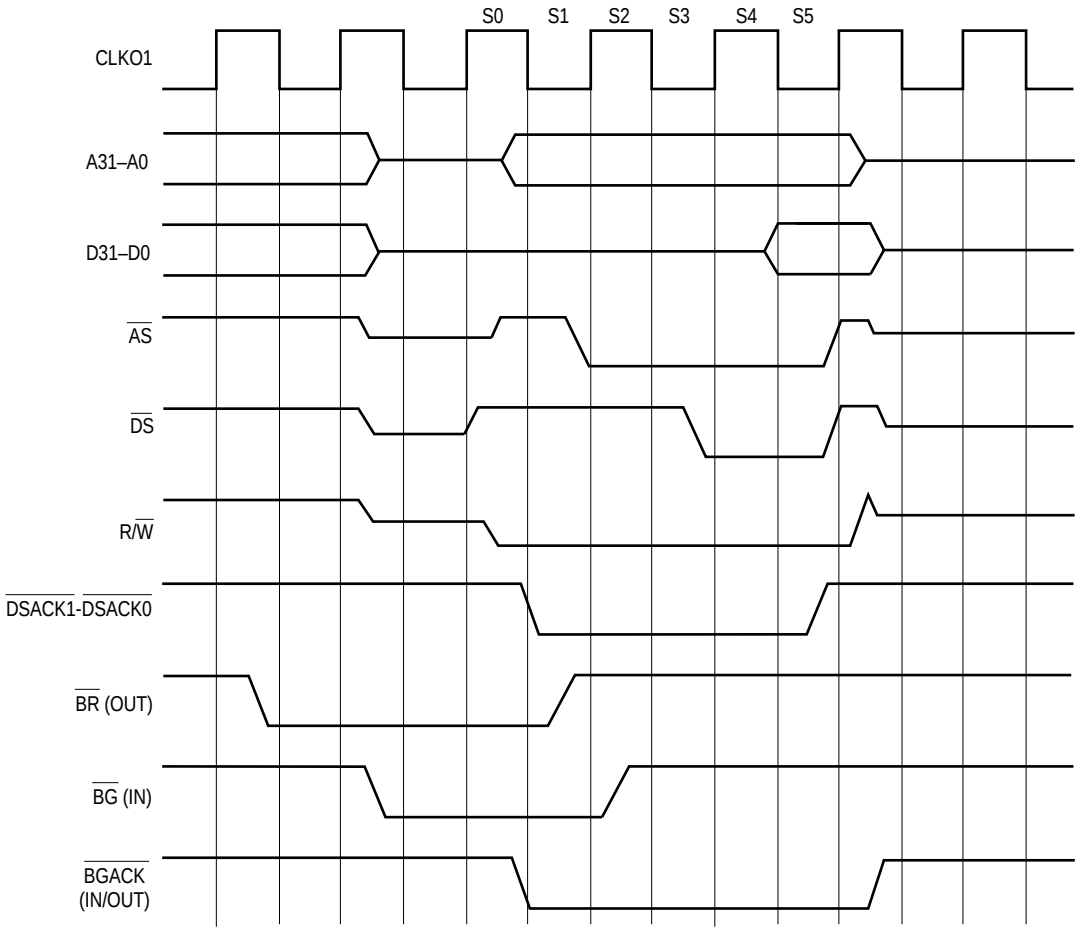
Address	Name	Width	Description	Reset Value		Block
REGB + 0000	MCR	32	Module Configuration Register	0000 7cff	H	SIM
REGB + 0004		32	Reserved			
REGB + 0008	AVR	8	Autovector Register	00	H	
REGB + 0009	RSR	8	Reset Status Register		H/S	
REGB + 000a		16	Reserved			
REGB + 000c	CLKOCR	8	CLKO Control Register	f(MODCK1)	H	
REGB + 000d			Reserved			
REGB + 0010	PLLCR	16	PLL Control Register	f(MODCK1–0)	H	
REGB + 0012		16	Reserved			
REGB + 0014	CDVCR	16	Clock Divider Control Register	0000	H	
REGB + 0016	PEPAR	16	Port E Pin Assignment Register	0000	H	
REGB + 0018 to REGB + 0021			Reserved			
REGB + 0022	SYPCR	8	System Protection Control	f(MODCK1–0)	H	
REGB + 0023	SWIV	8	Software Interrupt Vector	0F	H	
REGB + 0024		16	Reserved			
REGB + 0026	PICR	16	Periodic Interrupt Control Register	000F	H	
REGB + 0028		16	Reserved			
REGB + 002a	PITR	16	Periodic Interrupt Timing Register	0000/0300	H	
REGB + 002c		24	Reserved			
REGB + 002f	SWSR	8	Software Service Register	00	H	
REGB + 0030	BKAR	32	Breakpoint Address Register	XXXX	—	
REGB + 0034	BKCR	32	Breakpoint Control Register	0000 0000	H	
REGB + 0038 to REGB + 003f			Reserved			
REGB + 0040	GMR	32	Global Memory Register	0000 1200	H	MEMC
REGB + 0044	MSTAT	16	Memory Controller Status Register	0000	H	
REGB + 0046 to REGB + 004f			Reserved			
REGB + 0050	BR0	32	Base Register 0	0000 0051	H	
REGB + 0054	OR0	32	Option Register 0	F000 0000	H	
REGB + 0058 to REGB + 005f			Reserved			
REGB + 0060	BR1	32	Base Register 1	0000 0050	H	
REGB + 0064	OR1	32	Option Register 1	F000 000x	H	
REGB + 0068 to REGB +006f			Reserved			
REGB + 0070	BR2	32	Base Register 2	0000 0050	H	
REGB + 0074	OR2	32	Option Register 2	F000 000x	H	

The QUICC has another mechanism to assign priorities to the bus masters. A new pin called bus clear in (BCLRI) is defined. BCLRI indicates to the QUICC that a request is being made for the QUICC to release the system bus. The QUICC will then clear all internal bus masters with an arbitration ID smaller than the programmed value of the bus clear in ID (BCLRIID) in the MCR.

Slave (disable CPU32+) mode bus arbitration has fewer arbitration modes than exist in a normal mode, since in slave mode, the SHEN1-SHEN0 bits are forced to be "00":

- In synchronous mode (ASTM bit in the MCR is set), $\overline{BG}$ and $\overline{BGACK}$ have synchronous timing, and the minimal delay between the assertion of $\overline{BG}$ (negation of $\overline{BGACK}$) and the assertion of $\overline{BGACK}$ is one clock.
- In asynchronous mode, the minimum time for $\overline{BGACK}$ assertion after $\overline{BG}$ is asserted ($\overline{BGACK}$ is negated) depends on internal synchronization.
- The QUICC will not request the external bus (assert $\overline{BR}$) when one of its internal masters is making an internal access. The QUICC will request the external bus only when one of its internal masters is beginning an external access. In this case, the arbitration overhead (external bus idle time is minimal).

See Figure 4-40 for the slave mode bus arbitration timing diagram.



NOTES:
1. Synchronous arbitration with SHEN1–SHEN0 = 00.
2. Minimum bus idle time.

Figure 4-40. Slave Mode Bus Arbitration Timing Diagram

BDM operation is enabled when BKPT is asserted (low) at the rising edge of RESET. BDM remains enabled until the next system reset. A high BKPT on the trailing edge of RESET disables BDM. BKPT is relatched on each rising transition of RESET. BKPT is synchronized internally and must be held low for at least two clock(four clocks for RESETS)cycles prior to negation of RESETH.

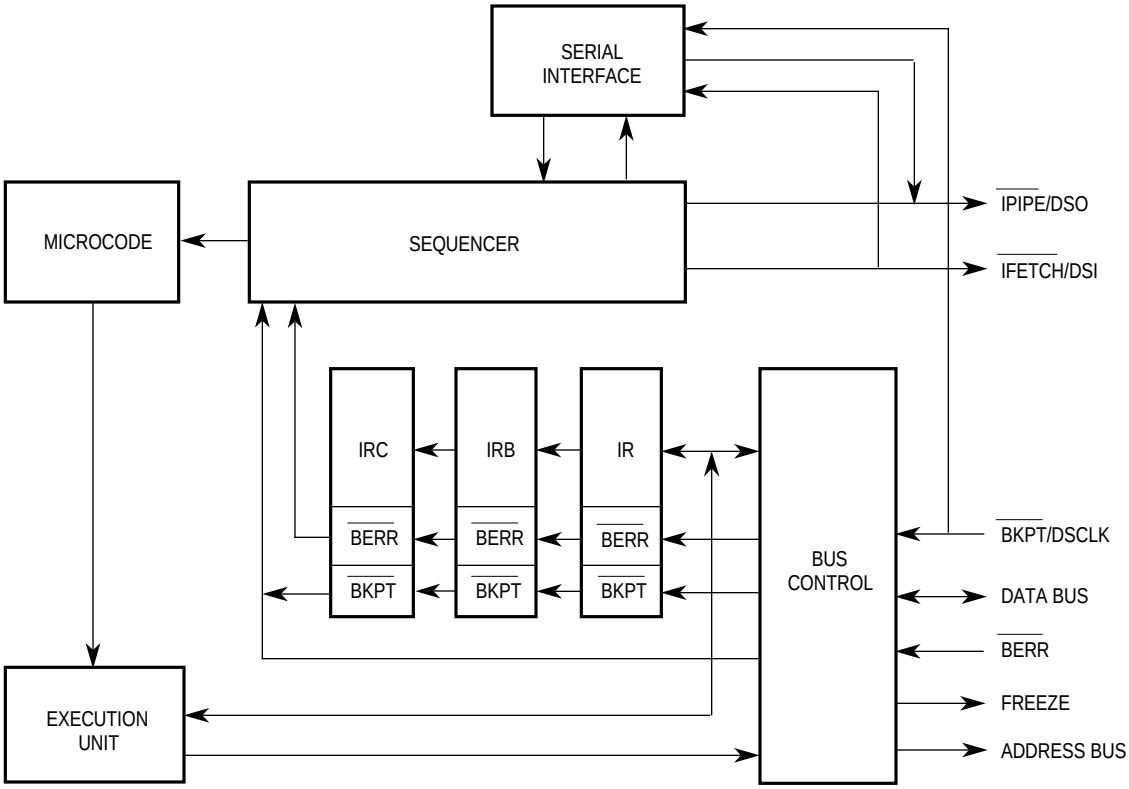


Figure 5-20. BDM Block Diagram

BDM enable logic must be designed with special care. If hold time on BKPT (after the trailing edge of RESET) extends into the first bus cycle following reset, this bus cycle could be tagged with a breakpoint. Refer to Section 4 Bus Operation for timing information.

5.6.2.2 BDM SOURCES. When BDM is enabled, any of several sources can cause the transition from normal mode to BDM. These sources include external BKPT hardware, the BGND instruction, a double bus fault, and internal peripheral breakpoints. If BDM is not enabled when an exception condition occurs, the exception is processed normally. Table 5-19 summarizes the processing of each source for both enabled and disabled cases. Note that the BKPT instruction never causes a transition into BDM.

Table 5-19. BDM Source Summary

Source	BDM Enabled	BDM Disabled
BKPT	Background	Breakpoint Exception
Double Bus Fault	Background	Halted
BGND Instruction	Background	Illegal Instruction
BKPT Instruction	Opcode Substitution/ Illegal Instruction	Opcode Substitution/ Illegal Instruction

NOTES

The breakpoint logic will assert the break signal only when the address and size match the programmable value. For example, if the programmable address is xxx2 with word size, the breakpoint will be asserted only when the access address is xxx2 with word size, not when the address is xxx0 with long-word size.

The MA bits must be 00 for the size comparison to occur.

00 = Long Word
01 = Byte
10 = Word
11 = 3 Byte

Table 6-7. Breakpoint and Size Pin

Programmed BA1-BA0 (BKAR Reg)	IMB/EXT A1-A0	IMB/EXT SIZ1-SIZ0	Assert $\overline{\text{BKPT}}$
00	00	x	Yes
	01	x	No
	10	x	No
	11	x	No
01	00	00	Yes
	00	01	No
	00	1x	Yes
	01	x	Yes
	1x	x	Yes
10	00	00	Yes
	00	01	No
	00	10	No
	00	11	Yes
	01	00	Yes
	01	01	No
	01	1x	Yes
	10	x	Yes
	11	x	No
11	00	00	Yes
	00	01	No
	00	1x	No
	01	00	Yes
	01	01	No
	01	10	No
	01	11	Yes
	10	00	Yes
	10	01	No
	10	1x	Yes
	11	x	Yes

DSSEL—Dynamic RAM Select

This bit determines if the bank is a DRAM or SRAM, which impacts a number of signals: 1) the length of the cycle is different; 2) address muxing is performed if GAMX = 1; and 3) the previous $\overline{RAS}$ is negated if a page bank miss occurs and DSSEL = 1 (for the new bank).

0 = SRAM bank (i.e., SRAM, EPROM, peripherals, etc.)
1 = DRAM bank

SPS1–SPS0—SRAM Port Size (SRAM Bank Only)

This attribute determines whether a given chip select responds with $\overline{DSACKx}$ and, if so, what port size is returned (see Table 6-13).

If the cycle is terminated by using the internal wait-state attributes, the QUICC drives the $\overline{DSACKx}$ lines according to those bits. If the internal wait-state attributes are not used, the cycle should be terminated with external $\overline{DSACKx}$. In this case, the QUICC does not drive the $\overline{DSACKx}$ lines, but rather samples them at every falling edge of the clock.

If an MC68EC040 access is performed using this SRAM bank and SPS= 00, 01, or 10, the SRAM controller operates in the same way, except it asserts $\overline{TA}$ instead of $\overline{DSACKx}$. If SPS= 11, $\overline{TA}$ is sampled at every rising edge of the clock.

Table 6-13. SRAM Port Size

SPS1–SPS0	Result
00	32-Bit Port Size
01	16-Bit Port Size
10	8-Bit Port Size
11	External $\overline{DSACKx}$ Response

NOTES

If DSACK is provided internally, then the $\overline{DSACKx}$ lines are still sampled externally, and can be asserted externally to end the cycle. However, in this case of external $\overline{DSACKx}$ assertion, external $\overline{DSACKx}$ should be asserted and negated prior to when internal DSACK would have been asserted by the QUICC. This is easily accomplished on the boot chip select since the QUICC default value is 14 wait states.

The SRAM controller does not support an external $\overline{TA}$ response for MC68040 burst mode. Also, for non-burst MC68040 cycles, $\overline{TA}$ cannot be externally asserted before $\overline{CS}$ is asserted.

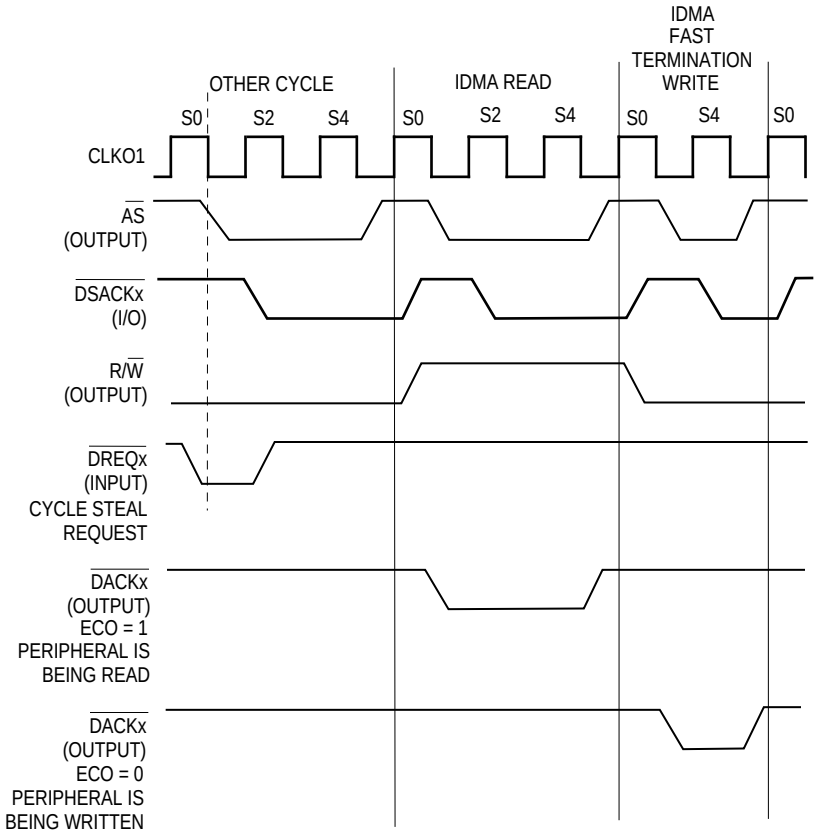
PGME—Page Mode Enabled (DRAM Banks Only)

This bit is used to enable page mode accesses to a DRAM bank. Page mode accesses are performed only for an internal QUICC or an external QUICC/MC68030-type master.

0 = Page mode is disabled.
1 = Page mode is enabled.

Single Address Destination Write. During the single address destination write cycle, the source device is controlled by the IDMA handshake signals ($\overline{\text{DREQx}}$, $\overline{\text{DACKx}}$, and $\overline{\text{DONEx}}$). When the source device requests service from the IDMA channel, the IDMA asserts of $\overline{\text{DACKx}}$ to allow the source device to drive data onto the data bus. The data is written to the device or to memory selected by the address in the DAPR, the destination function codes in the FCR, and the size in the CMR. The data bus is placed in a high-impedance state for this write cycle. For more details about the IDMA handshake signals, see 7.6.3 Interface Signals.

7.6.4.6.3 Fast-Termination Option. While in the operand transfer phase, the IDMA supports an option to achieve a transfer in the shortest possible number of clocks (see Figure 7-16).



NOTE: This example shows a fast termination on the write cycle. The fast termination may occur on the read, write, or both.

Figure 7-16. Fast Termination Example

Using the SIM60 chip-select logic, the fast-termination option can be employed to give a fast bus access of two clock cycles rather than the standard three-cycle access time. The fast-termination option is described in Section 6 System Integration Module (SIM60) and in Section 4 Bus Operation.

If the fast-termination option is used with external request burst mode, an extra IDMA cycle results on every burst transfer. In the burst mode with fast termination selected, a new cycle starts even if $\overline{\text{DREQx}}$ negation and $\overline{\text{DACKx}}$ assertion occur simultaneously.

The user can terminate the transfer by setting the RST bit in the CMR and then issuing the INIT_IDMA command.

The user can terminate the transfer with an "out of buffers" error if the V-bit of one of the BDs is cleared by the user. When the RISC reaches this IDMA BD, it will terminate activity. This technique is useful when the IDMA is required to stop transfers after fully completing a BD transfer.

If the BCR is decremented to zero, the transfer from this BD completes, but the RISC controller reloads the IDMA registers with the values from the next IDMA BD, and the IDMA transfer continues. Thus, the fact that the BCR is decremented to zero does not terminate a transfer in auto buffer mode; it only terminates the current BD transfer.

If DONEx is asserted externally, the transmission from this BD is terminated and the following actions are performed by the RISC controller:

1. Sets the Done Bit in the status register
2. Sets the DA bit in the BD
3. Clears the Valid bit in the BD
4. Resets the start bit in the CMR

Thus the current buffer is closed immediately and all IDMA operation ceases.

7.6.4.8.3 Buffer Chaining Mode Termination. The user can suspend a transfer in auto buffer mode by clearing the STR bit in the CMR. When STR is set once again, the transfer will continue.

The user can terminate the transfer by setting the RST bit in the CMR and then issuing the INIT_IDMA command.

The user can also terminate the transfer by setting the L-bit in the IDMA BD. When processing of this BD has completed, the transmission will terminate with the DONE bit being set in the CSR. This can cause an interrupt if the corresponding bit in the CMAR is set.

If the BCR is decremented to zero, the transfer from this BD completes, but the RISC controller reloads the IDMA registers with the values from the next IDMA BD, and the IDMA transfer continues. Thus, the fact that the BCR is decremented to zero does not terminate a transfer in buffer chaining mode; it only terminates the current BD transfer.

If $\overline{\text{DONEx}}$ is asserted externally, the transmission from this BD is terminated and the following actions are performed by the RISC controller.

1. Sets the Done Bit in the status register
2. Sets the Abort bit in the BD
3. Clears the Ready bit in the BD
4. Resets the start bit in the CMR
5. Sets the Reset bit in the CMR

FEx—Frame Sync Edge for TDM A or B

The L1RSYNCx and L1TSYNCx pulses are sampled with the falling/rising edge of the channel clock according to this bit.

0 = Falling edge (Use for IDL and GCI.)
1 = Rising edge

GMx—Grant Mode for TDM A or B

0 = GCI/SCIT mode. The GCI/SCIT D channel grant mechanism for transmission is internally supported. The grant is one bit from the receive channel. This bit is marked by programming the channel select bits of the SI RAM with 111 to assert an internal strobe on it. Refer to 7.8.7.2.2 SCIT Programming.
1 = IDL mode. A GRANT mechanism is supported if the corresponding GR1–GR4 bits in the SIMODE register are set. The grant is a sample of the L1GRx pin while L1TSYNCx is asserted. This GRANT mechanism implies the IDL access controls for transmission on the D channel. Refer to 7.8.6.2 IDL Interface Programming.

TFSDx—Transmit Frame Sync Delay for TDM A or B

These two bits determine the number of clock delays between the transmit sync and the first bit of the transmit frame. If the CRTx bit is set (recommended with IDL or GCI), then the transmit sync is not used, and these bits are ignored.

00 = No bit delay (The first bit of the frame is transmitted/received on the same clock as the sync.)
01 = 1 bit delay
10 = 2 bit delay
11 = 3 bit delay

Refer to Figure 7-29 and Figure 7-30 for an example of the use of these bits.

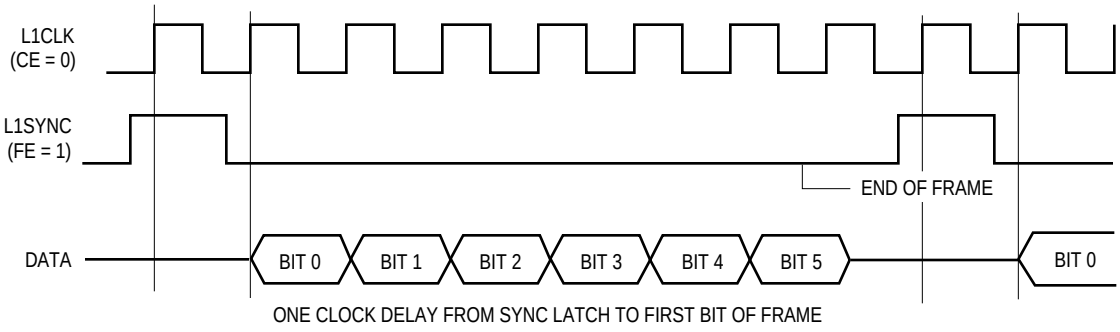


Figure 7-29. One Clock Delay from Sync to Data (RFSD = 01)

7.8.5.4 SI COMMAND REGISTER (SICMR). The 8-bit SICMR allows the user to dynamically program the SI RAM. For more information about dynamic programming, refer to 7.8.4.7 SI RAM Dynamic Changes

The contents of this register are valid only in the RAM division mode (RDM1–RDM0 bits in SIGMR equal 01 or 11). This register is cleared at reset.

7	6	5	4	3	2	1	0
CSRRa	CSRTa	CSRRb	CSRTb	—			

CSRRx—Change Shadow RAM for TDM A or B Receiver

When set, this bit will cause the SI receiver to replace the current route with the shadow RAM. The bit is set by the user and cleared by the SI.

0 = The receiver shadow RAM is not valid. The user can write into the shadow RAM to program a new routing.
1 = The receiver shadow RAM is valid. The SI will exchange between the RAMs and take the new receive routing from the receiver shadow RAM. This bit is cleared as soon as the switch has completed.

CSRTx—Change Shadow RAM for TDM A or B Transmitter

When set, this bit will cause the SI transmitter to replace the current route with the shadow RAM. The bit is set by the user and cleared by the SI.

0 = The transmitter shadow RAM is not valid. The user can write into the shadow RAM to program a new routing.
1 = The transmitter shadow RAM is valid. The SI will exchange between the RAMs and take the new transmitter routing from the receiver shadow RAM. This bit is cleared as soon as the switch has completed.

Bits 3–0—Reserved

These bits should be set to zero by the user.

7.8.5.5 SI STATUS REGISTER (SISTR). The 8-bit SISTR indicates to the user which part of the SI RAM is the current-route RAM. The value of this register is valid only when the corresponding bit in the SIGMR is clear. This register is cleared at reset.

CRORa—Current Route of TDMa Receiver

7	6	5	4	3	2	1	0
CRORa	CROTa	CRORb	CROTb	—			

0 = The current-route receiver RAM is in address:
0–63 when the SI supports one TDM (RDM = 01)
0–31 when the SI supports two TDMs (RDM = 11)
1 = The current route receiver RAM is in address:
64–127 when the SI supports one TDM (RDM = 01)
32–63 when the SI supports two TDMs (RDM = 11)

TPL—Tx Preamble Length

The TPL bits determine the length of the preamble configured by the TPP bits.

000 = No preamble (default)
001 = 8 bits (1 byte)
010 = 16 bits (2 bytes)
011 = 32 bits (4 bytes)
100 = 48 bits (6 bytes) (Select this setting for Ethernet operation.)
101 = 64 bits (8 bytes)
110 = 128 bits (16 bytes)
111 = Reserved

TPP—Tx Preamble Pattern

The TPP bits determine what, if any, bit pattern should precede the start of each transmit frame. The preamble pattern will be sent prior to the first flag/sync of the frame. TPP is ignored if the SCC is programmed to UART mode. The length of the preamble is programmed in TPL. The preamble pattern is typically transmitted to a receiving station that uses a DPLL for clock recovery. The receiving DPLL uses the regular pattern of the preamble to help it lock onto the received signal in a short, predictable time period.

00 = All zeros
01 = Repeating 10’s (Select this setting for Ethernet operation.)
10 = Repeating 01’s
11 = All ones (Select this setting for LocalTalk operation.)

Tend—Transmitter Frame Ending

This bit is intended particularly for the NMSI transmitter encoding of the DPLL. Tend determines whether the TXD line should idle in a high state or in an encoded ones state (which may be either high or low). It may, however, be used with other encodings besides NMSI.

0 = Default operation. The TXD line is encoded only when data is transmitted (including the preamble and opening and closing flags/syncs). When no data is available to transmit, the line is driven high.
1 = The TXD line is always encoded (even when idles are transmitted).

TDCR—Transmit Divide Clock Rate

The TDCR bits determine the divider rate of the transmitter. If the DPLL is not used, the 1× value should be chosen, except in asynchronous UART mode where 8×, 16×, or 32× must be chosen. The user should program TDCR to equal RDCR in most applications.

If the DPLL is used in the application, the selection of TDCR depends on the encoding. NRZI usualy requires 1×; whereas, FM0/FM1, Manchester, and Differential Manchester allow 8×, 16×, or 32×. The 8× option allows highest speed; whereas, the 32× option provides the greatest resolution. TDCR is usually equal to RDCR to allow the same clock frequency source to control both the transmitter and receiver.

00 = 1× clock mode (Only NRZ or NRZI encodings are allowed.)
01 = 8× clock mode
10 = 16× clock mode (normally chosen for UART and AppleTalk)
11 = 32× clock mode

channel counts the number of consecutive idle characters received. If the count reaches the value programmed into MAX_IDL, the buffer is closed, and an RX interrupt is generated. If no receive buffer is open, this event does not generate an interrupt or any status information. The internal idle counter (IDLC) is reset every time a character is received.

NOTE

To disable the idle sequence function entirely, set the MAX_IDL value to zero.

Framing Error*.* A framing error is detected by the UART controller when a character is received with no stop bit. All framing errors are report by the UART controller, regardless of the UART mode. When this error occurs, the channel writes the received character to the buffer, closes the buffer, sets the FR bit in the BD, and generates the RX interrupt (if enabled). The channel also increments the FRMEC. When this error occurs, parity is not checked for this character. In automatic multidrop mode, the receiver enters hunt mode immediately.

If the RZS bit is set in the UART mode register when the UART is in the synchronous mode (SYN is set), then the receiver reports all framing errors, but continues reception with the assumption that the unexpected zero is really the start bit of the next character. If RZS is set, user software may not wish to consider a reported UART framing error as a true UART framing error, unless two or more framing errors occur within a short period of time.

Break Sequence*.* The UART offers very flexible break support for the receiver. When the first break sequence is received (one or more all-zero characters), the UART increments the BRKEC and issues the break start (BRKs) event in the UART event register, which can generate an interrupt (if enabled). The UART then measures the break length, and, when the break sequence is complete, writes the length to the BRKLN register. After the first one is received, the UART also issues the break end (BRKe) event in the UART event register, which can generate an interrupt (if enabled). If the UART was in the process of receiving characters when the break was received, it will also close the receive buffer, set the BR bit in the Rx BD, and write the RX bit in the event register, which can generate an interrupt (if enabled).

If the RZS bit is set in the UART mode register when the UART is in the synchronous mode (SYN is set), then a break sequence will be detected only after two successive break characters are received.

7.10.16.15 UART MODE REGISTER (PSMR). Each PSMR is a 16-bit, memory-mapped, read-write register that controls SCC operation. When the SCC is configured as a UART, this register is called the UART mode register. This register is cleared at reset. Many of the PSMR bits may be modified on the fly (i.e., while the receiver and transmitter are enabled).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLC	SL	CL		UM		FRZ	RZS	SYN	DRT	—	PEN	RPM		TPM	

generates a maskable interrupt, and starts to receive data into the next buffer after one of the following events:

1. Receiving a user-defined control character
2. Detecting an error
3. Detecting a full receive buffer
4. Issuing the ENTER HUNT MODE command
5. Issuing the CLOSE Rx BD command

212	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	**E**	—	**W**	I	L	F	**CM**	—	DE	—	—	NO	—	CR	OV	CD
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RX DATA BUFFER POINTER															
OFFSET + 6																

NOTE: Entries in boldface must be initialized by the user.

E—Empty

0 = The data buffer associated with this Rx BD has been filled with received data, or data reception has been aborted due to an error condition. The CPU32+ core is free to examine or write to any fields of this Rx BD. The CP will not use this BD again while the E-bit remains zero.

1 = The data buffer associated with this Rx BD is empty, or reception is currently in progress. This Rx BD and its associated receive buffer are owned by the CP. Once the E-bit is set, the CPU32+ core should not write any fields of this Rx BD.

Bits 14, 8, 6, 5—Reserved

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Rx BD table.

1 = This is the last BD in the Rx BD table. After this buffer has been used, the CP will receive incoming data into the first BD in the table (the BD pointed to by RBASE). The number of Rx BDs in this table is programmable and is determined only by the W-bit and the overall space constraints of the dual-port RAM.

I—Interrupt

0 = No interrupt is generated after this buffer has been used.

1 = The RX bit in the BISYNC event register will be set when this buffer has been closed by the BISYNC controller. The RX bit can cause an interrupt if it is enabled.

L—Last in Frame

This bit is set by the transparent controller when this buffer is the last in a frame. This implies the negation of CD in envelope mode or the reception of an error, in which

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	**E**	—	**W**	**I**	L	F	—	—	—	—	LG	NO	SH	CR	OV	CL
OFFSET + 2	DATA LENGTH															
OFFSET + 4	**RX DATA BUFFER POINTER**															
OFFSET + 6																

NOTE: Entries in boldface must be initialized by the user.

E—Empty

0 = The data buffer associated with this Rx BD has been filled with received data, or data reception has been aborted due to an error condition. The CPU32+ core is free to examine or write to any fields of this Rx BD. The CP will not use this BD again while the E-bit remains zero.

1 = The data buffer associated with this Rx BD is empty, or reception is currently in progress. This Rx BD and its associated receive buffer are owned by the CP. Once the E bit is set, the CPU32+ core should not write any fields of this Rx BD.

Bits 14, 9–6—Reserved

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Rx BD table.

1 = This is the last BD in the Rx BD table. After this buffer has been used, the CP will receive incoming data into the first BD in the table (the BD pointed to by RBASE). The number of Rx BD s in this table is programmable and is determined only by the W-bit and the overall space constraints of the dual-port RAM.

I—Interrupt

0 = No interrupt is generated after this buffer has been used.

1 = The RXB bit or RXF bit in the Ethernet event register will be set when this buffer has been used by the Ethernet controller. These two bits may cause interrupts if they are enabled.

L—Last in Frame

This bit is set by the Ethernet controller when this buffer is the last in a frame. This implies the end of the frame or reception of an error, in which case one or more of the CL, OV, CR, SH, NO, and LG bits are set. The Ethernet controller will write the number of frame octets to the data length field.

0 = The buffer is not the last in a frame.

1 = The buffer is the last in a frame.

F—First in Frame

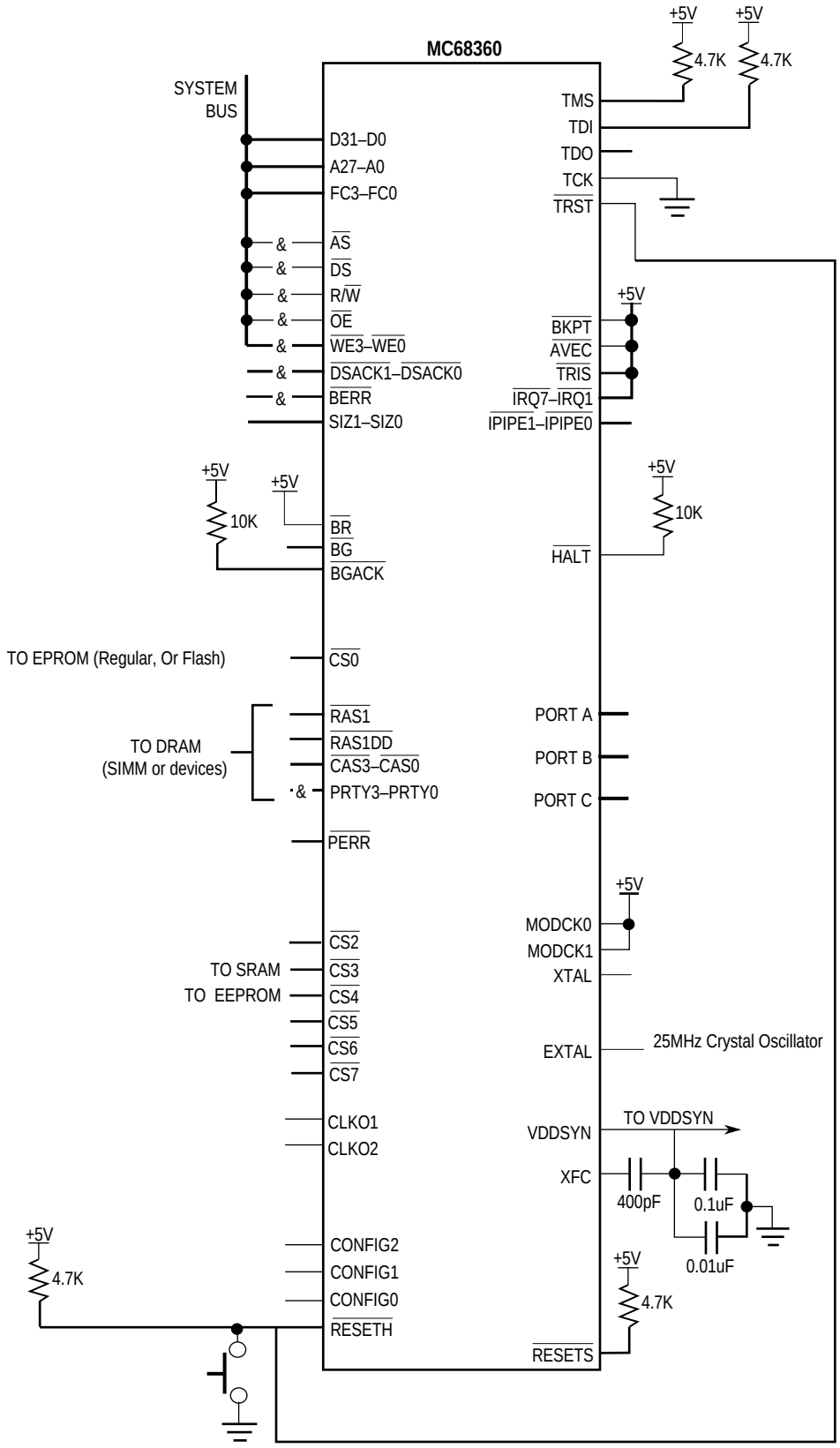
This bit is set by the Ethernet controller when this buffer is the first in a frame.

0 = The buffer is not the first in a frame.

1 = The buffer is the first in a frame.

M—Miss

This bit is set by the Ethernet controller for frames that were accepted in promiscuous mode, but were flagged as a "miss" by the internal address recognition. Thus, while in pro-



LEGEND:
& = Pullups recommended (<= 10K)

Figure 9-1. MC68360 Minimum System Configuration

EEPROM may be accessed in succession. The $\overline{CS4}$ pin should be programmed to respond to an 8-Kbyte area in this design.

Only one byte should be written at a time. After a write is made, software is responsible for waiting the appropriate time (e.g., 10 ms) or for performing data polling to see if the newly written data byte is correct.

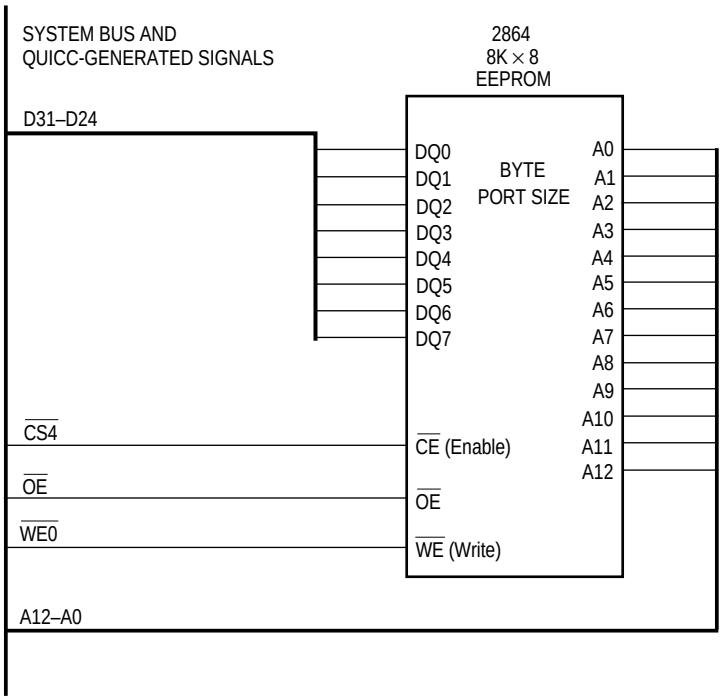


Figure 9-5. Glueless Interface to EEPROM

9.1.2.6 DRAM SIMM. Figure 9-6 shows the glueless interface to an MCM36100S DRAM single in-line memory module (SIMM). The $\overline{RAS1}$ line should be programmed to respond to a 4-Mbyte address space.

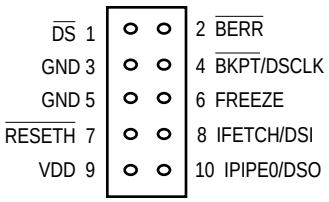
This particular SIMM also includes parity support, which is supported with the PRTY3–PRTY0 signals.

This design also uses the $\overline{RAS1}$ double-drive capability, whereby the $\overline{RAS1DD}$ signal is output by the QUICC to increase the effective drive capability of the $\overline{RAS1}$ signal.

After power-on reset, the software must wait the required time before accessing the DRAM. The required eight read cycles must be performed either in software or by waiting for the refresh controller to perform these accesses.

ing needs (such as in-field debugging). In the second and third cases, the target board needs to provide the BDM pins exposed so that an external monitor/debugger can connect to the target board. Figure 9-34 simply shows a standard connector for use with BDM equipment.

The standard connector originally an 8-pin connector, has been expanded to 10 pins as shown in Figure 9-34. This connector is sometimes referred to as the Berg connector. It has the standard 0.1-inch spacing between pins. The original 8 pins on the connector are the lower 8 pins (pins 3–10). If a debug monitor provides an 8-pin BDM connector, it should be plugged onto the original 8 pins. The additional 2 pins were added to allow hardware breakpoints to be implemented using the BDM connector. Since the QUICC contains an on-chip breakpoint address register with masking, this function can be implemented on-chip. To do this, the user can load the breakpoint address register in the SIM60 using the debugger commands (or under normal program control). Once execution of the program resumes, an address breakpoint then causes the QUICC to reenter BDM.



NOTES:
1. On other M68300 family devices RESETH is simply RESET, and IPIPE0 is simply IPIPE.
2. The original 8-pin connector consists of pins 3 through 10.

Figure 9-34. BDM Connector

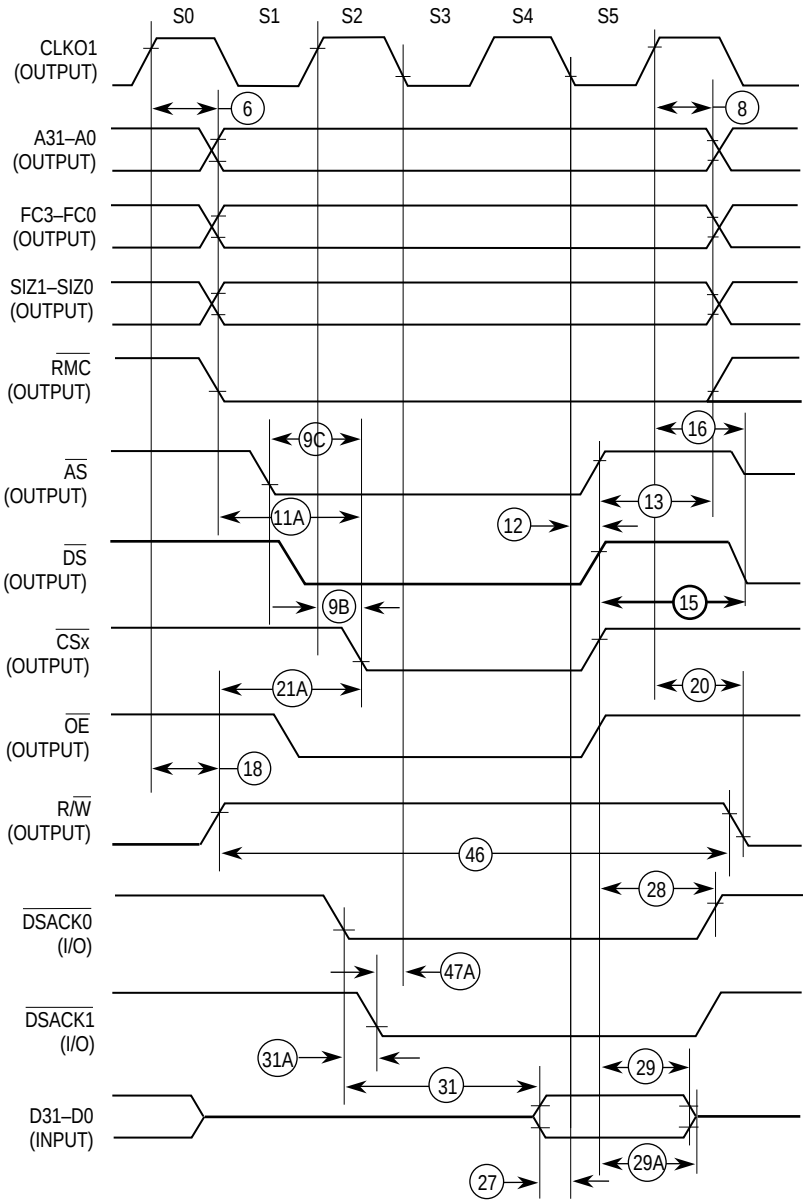


Figure 10-6. SRAM: Read Cycle (TRLX = 1)